NEW FRONTIERS IN NEURAL PROBABILISTIC SCORING: FROM ATTENTION TO OUTPUT GENERATION IN VISION AND LANGUAGE

Inauguraldissertation
zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften
der Universität Mannheim

vorgelegt von

Yuxuan Zhou

aus Chengdu

Mannheim, 2025

Dekan:	Prof. Dr. Claus Hertling, Universität Mannheim
Referent:	Prof. Dr.-Ing. Margret Keuper, Universität Mannheim
Korreferent:	Prof. Dr. Mario Fritz, Universität des Saarlandes
Korreferent:	Prof. Dr. Marcus Rohrbach, Technische Universität Darmstadt

Tag der mündlichen Prüfung: 19.September.2025

Abstract

Recent advancements in deep learning have highlighted the importance of probabilistic scoring within attention mechanisms and model predictions, significantly impacting tasks in computer vision and natural language processing. Neural probabilistic scoring refers to the process of computing normalized relevance scores based on hidden features of a neural network - often via softmax - that sum to one and reflect the relative importance of different tokens or features, without necessarily representing true probability distributions. Traditional reliance on softmax-based attention and output distributions can constrain model capacity and reliability. Its unimodal nature restricts capturing sparse, multi-modal patterns and reduces robustness to signal noise. Additionally, permutation invariance in scoring disrupts spatial and structural information, hindering performance on tasks with complex geometry or topology. This thesis addresses these limitations by introducing novel methodologies that refine probabilistic scoring in both the attention and output layers, aiming to enhance the performance and scalability of machine learning models across vision and language tasks.

In the first block, the work reimagines attention mechanisms. Central to this is MultiMax, a novel softmax alternative that achieves an improved balance between sparsity and multi-modality in the output distribution, enabling the attention mechanism to simultaneously focus on multiple relevant contexts while maintaining resilience to irrelevant entries. In the vision domain, Sp-ViT introduces learnable 2D spatial priors into Vision Transformers, enhancing the model's ability to capture spatial relationships and improving performance in image classification tasks. For structured data, the work proposes Hypergraph Transformer to tackle skeleton-based action recognition, with hypergraph attention and a positional encoding based on graph distances as its core components. The work further extends the positional encoding with topological encoding, which successfully incorporates more comprehensive structural information through topological descriptors beyond graph representation.

The second block focuses on output probabilistic scoring to improve model reliability for both discriminative and generative models. During training, MaxSup regularizes classifiers' output by mitigating the overconfidence in erroneous predictions and representation collapse in label smoothing, leading to more reliable predictions and more powerful feature representations. At inference, sampling-based decoding strategies modulate output distributions to improve LLMs' output, balancing diversity and coherence in open-ended text generation. Together, MaxSup and LLM Sampling provide a unified framework for output probabilistic scoring, ensuring reliability and quality in both classification and generative tasks.

Zusammenfassung

Jüngste Fortschritte im Deep Learning haben die Bedeutung probabilistischer Bewertungen innerhalb von Aufmerksamkeitsmechanismen und Modellvorhersagen hervorgehoben, die Aufgaben in der Computer Vision und der Verarbeitung natürlicher Sprache maßgeblich beeinflussen. Unter neuronaler probabilistischer Bewertung versteht man den Prozess der Berechnung normalisierter Relevanzwerte – oft über Softmax – auf Basis versteckter Merkmale eines neuronalen Netzwerks, die sich zu eins summieren und die relative Bedeutung verschiedener Token oder Merkmale widerspiegeln, ohne dabei notwendigerweise echte Wahrscheinlichkeitsverteilungen darzustellen. Die traditionelle Abhängigkeit von Softmax-basierten Aufmerksamkeits- und Ausgabeverteilungen kann die Modellkapazität und Zuverlässigkeit einschränken. Ihre unimodale Natur behindert die Erfassung sparsamer, multimodaler Muster und verringert die Robustheit gegenüber Störsignalen. Darüber hinaus führt Permutationsinvarianz in der Bewertung zur Vernachlässigung räumlicher und struktureller Informationen, was die Leistung bei Aufgaben mit komplexer Geometrie oder Topologie beeinträchtigt. Diese Arbeit begegnet diesen Einschränkungen durch die Einführung neuer Methoden, die die probabilistische Bewertung sowohl in der Aufmerksamkeits- als auch in der Ausgabeschicht verfeinern, mit dem Ziel, die Leistung und Skalierbarkeit von Modellen des maschinellen Lernens in Aufgaben der Bild- und Sprachverarbeitung zu verbessern.

Im ersten Teil wird der Aufmerksamkeitsmechanismus neu gedacht. Im Zentrum steht MultiMax, eine neuartige Alternative zu Softmax, die ein verbessertes Gleichgewicht zwischen Sparsität und Multimodalität in der Ausgabeverteilung erreicht und es dem Aufmerksamkeitsmechanismus ermöglicht, gleichzeitig auf mehrere relevante Kontexte zu fokussieren, während er gegenüber irrelevanten Einträgen robust bleibt. Im Bereich der Bildverarbeitung führt Sp-ViT lernbare zweidimensionale räumliche Prioren in Vision Transformers ein, wodurch das Modell räumliche Beziehungen besser erfassen kann und die Leistung bei Bildklassifizierungsaufgaben verbessert wird. Für strukturierte Daten schlägt die Arbeit den Hypergraph Transformer zur Erkennung skelettbasierter Aktionen vor, dessen Kernkomponenten eine Hypergraphen-Aufmerksamkeit sowie eine Positionskodierung basierend auf Graph-Distanzen sind. Darüber hinaus wird die Positionskodierung durch eine topologische Kodierung erweitert, welche umfassendere strukturelle Informationen über topologische Deskriptoren jenseits der klassischen Graphdarstellung integriert.

Der zweite Teil konzentriert sich auf die probabilistische Bewertung der Ausgaben, um die Zuverlässigkeit von Modellen sowohl im diskriminativen als auch im generativen Bereich zu verbessern. Während des Trainings reguliert MaxSup die Ausgabe von Klassifikatoren, indem es übermäßiges Vertrauen in fehlerhafte Vorhersagen und den Zusammenbruch der Repräsentation durch Label Smoothing reduziert. Dies führt zu zuverlässigeren Vorhersagen und ausdrucksstärkeren Merkmalsrepräsentationen. Beim Inferenzprozess modulieren Sampling-basierte Dekodierungsstrategien die Ausgabeverteilungen, um die Generierung offener Texte durch große Sprachmodelle (LLMs) zu verbessern – mit einem

ausgewogenen Verhältnis von Vielfalt und Kohärenz. Zusammen bilden MaxSup und LLM Sampling einen einheitlichen Rahmen für die probabilistische Bewertung von Ausgaben, der Zuverlässigkeit und Qualität in Klassifikations- und Generierungsaufgaben gleichermaßen gewährleistet.

Declaration on the Use of Language Models

During the preparation of this dissertation, the language model ChatGPT (developed by OpenAI) was used exclusively to assist with language polishing and improving readability. All scientific content, including the development of ideas, methods, results, and conclusions, was independently created and verified by the author. The tool was employed solely for stylistic and linguistic enhancement and did not influence the scientific substance of the work.

Acknowledgement

I would like to sincerely thank my supervisors, Prof. Margret Keuper and Prof. Mario Fritz, for their patience, encouragement, and invaluable mentorship throughout my PhD journey. I especially appreciate Prof. Keuper's trust and encouragement during the early years when I was struggling to publish papers. I am equally grateful to Prof. Fritz for agreeing to co-supervise me and for trusting me at an early stage of my research.

I also thank both Prof. Keuper and Prof. Fritz for granting me the freedom to explore my research interests while providing valuable guidance to shape and refine my research direction. Their mentorship greatly improved my ability to express complex ideas clearly and helped transform my early concepts into well-developed research. I deeply appreciate their willingness to help, sometimes late into the night, generously sacrificing personal time to offer thorough and thoughtful feedback. Their support has been essential to my academic and personal growth.

I express my gratitude to all colleagues at both the University of Mannheim and CISPA Helmholtz Center for Information Security for their invaluable discussions, constructive feedback, and continuous support throughout this journey. Special thanks to Shashank Agnihotri, Dingfan Chen, Hossein Hajipour, Xiaowen Jiang, Tobias Lorenz, Katharina Prasse, Ivaxi Sheth, Hui-Po Wang, Xun Wang, Yuan Xin and Zhixiong Zhuang for their kind help and support in daily work. Your practical assistance and friendly presence made my time during the PhD smoother and more enjoyable.

I would also like to express my sincere thanks to my collaborators, especially Prof. Zhi-Qi Cheng, for the opportunity to learn from him and for the supportive, inspiring environment he has fostered. His generous collaboration and insightful guidance have made a lasting impact on my work. Additionally, I thank Xudong Yan and Heng Li for their valuable collaboration.

I am thankful to the members of my thesis committee for their time, constructive comments, and encouragement.

Finally, I would like to thank my friends and family for their patience, love, and constant encouragement. Your belief in me gave me the strength to overcome many challenges along the way.

Contents

Chapter I.

Introduction

Deep learning [133] has revolutionized both computer vision [63, 126, 96, 24] and natural language processing [2, 224, 57]. Central to many of these advances are neural probabilistic scoring techniques, which compute normalized relevance or predictive scores based on hidden features of a neural network — often implemented via the Softmax function — to represent the importance of input elements or the likelihood of output classes. These mechanisms govern both attention layers [224] —where normalized scores determine contextual relevance—and output layers—where predictive distributions guide decisions and generation. However, these distributional approaches exhibit inherent limitations, particularly in adapting to structured data and aligning with real-world risk profiles.

While Softmax-based attention has provided a practical and widely adopted foundation, its limitations are increasingly apparent. As a smooth approximation to Argmax function, Softmax often struggles to balance sparsity and multi-modality in its generated scores [35, 174]. Moreover, attention mechanism naturally disrupts valuable spatial or structural information [201] due to its permutation equivalent formulation. In computer vision, for instance, Vision Transformers (ViTs) [63, 221] typically rely on hand-crafted one-dimensional positional encodings [201, 50], which constrain their ability to model the rich two-dimensional spatial relationships essential for understanding geometric structures. Similarly, in structured data tasks such as skeleton-based action recognition [249, 176], standard attention mechanisms and graph convolutional networks (GCNs) face challenges in capturing higher-order joint interactions and preserving the inherent topology of skeletal representations.

To address these challenges in attention mechanisms, this thesis presents several novel approaches. The Sp-ViT model [282] introduces learnable 2D spatial priors for Vision Transformers, enabling the model to capture richer geometric relationships and improve performance in vision tasks that require precise spatial understanding. Meanwhile, the MultiMax method [279] reimagines attention learning by extending the pareto frontier of the trade-off between sparsity and multi-modality, overcoming the limitations of traditional Softmax-based attention mechanisms. Additionally, the Hypergraph Transformer [278] and BlockGCN [283] tackle the challenge of skeleton-based action recognition by modeling higher-order inter-joint relationships and offering topology-aware graph convolution methods, respectively. These contributions address key limitations in structured data, ensuring better preservation of spatial and topological information in complex tasks like action recognition.

Despite advancements in attention mechanisms, optimizing output probabilistic scoring remains a challenge. For classifier outputs, regularization [216, 129, 287] is necessary to prevent overconfidence and improve the reliability of predictions. While label smoothing has been commonly adopted for this purpose via soft targets, it is shown to cause representation collapse [161], reduced transfer performance [127], and over-confidence in the erroneous prediction [284]. In generative tasks, such as open-ended text generation [70, 156], the situation is further complicated—directly sampling from the Softmax distribution may yield low-quality outputs, while greedy decoding and especially beam search often result in repetitive or overly predictable outputs [70, 108]. Striking the right balance between diversity and quality in these generative tasks is crucial; an improper sampling strategy can lead to incoherent or insufficiently varied text. These challenges underscore the need for more refined techniques to regulate output distributions, ensuring that model outputs are accurate and reliable, as well as diverse for generative models.

To address these challenges in output probabilistic scoring, this thesis introduces two novel approaches. MaxSup [281] is a new regularization technique that mitigates overconfidence in erroneous prediction and representation collapse in label smoothing. By penalizing the top-1 logit instead of the ground-truth logit, MaxSup uniformly applies regularization to both correct and incorrect predictions, leading to improved calibration and enhanced feature robustness. The second contribution [280] offers a systematic framework for selecting optimal sampling methods and parameters in large language models. By considering the trade-off between diversity and risk at each decoding step, this work facilitates adaptive decoding strategies to improve the quality and diversity of the generated text. Together, these contributions provide robust solutions for output probabilistic scoring, ensuring that both classifier outputs and generative model predictions are more reliable and aligned with real-world expectations.

This thesis is organized into two parts, each addressing key limitations in neural probabilistic scoring. Part One focuses on enhancing attention mechanisms by improving their expressiveness and structural alignment across different data modalities. It begins with Chapter III, which introduces MultiMax, a novel attention formulation that balances sparsity and multi-modality more effectively than Softmax. Chapter IV presents Sp-ViT, a Vision Transformer architecture augmented with learnable 2D spatial priors for improved geometric modeling. Chapters V and VI further explore structural modeling in skeleton-based action recognition: Hypergraph Transformer introduces higher-order attention over joints and structural encoding based on graph distances, while BlockGCN extends structural encoding beyond connectivity by leveraging topological descriptors that capture higher-order skeletal structures. Part Two shifts focus to output probabilistic scoring, addressing limitations in classification reliability and generative diversity. Chapter VII proposes MaxSup, a regularization method that mitigates over-confidence in erroneous predictions by eliminating the error amplification term in label smoothing, and is also shown to alleviate representation collapse. Chapter VIII presents a systematic framework for evaluating decoding methods and recommending their associated hyperparameters for large language models, balancing diversity and risk through adaptive sampling strategies.

1. Contribution Overview

In this thesis, we propose several contributions aimed at advancing probabilistic scoring techniques in both the attention and output layers across a variety of data modalities. An overview of each chapter's key contributions is provided in Table I.1, while the following sections present these contributions in greater detail.

Table I.1.: Thesis structure and key contributions under the unified probabilistic scoring framework.

Probabilistic Scoring in Neural Networks	
Part One: Enhancing Attention Scoring	**Part Two: Improving Output Scoring**
Chapter III: MultiMax [**279**] Balances sparsity and multi-modality in attention scores beyond Softmax limitations for image classification and language modeling.	**Chapter VII: MaxSup** [**281**] Regularizes top-1 logits to mitigate overconfidence and representation collapse in label smoothing for image classification.
Chapter IV: Sp-ViT [**282**] Enhances Vision Transformer with learnable 2D spatial priors for richer geometric modeling, while retaining the global receptive field.	**Chapter VIII: Decoding Framework** [**280**] Systematically evaluates the sampling strategies of Large Language Models and recommends their hyperparameters to balance diversity and risk for open-ended text generation.
Chapters V & VI: Hypergraph Transformer [**278**] **& BlockGCN** [**283**] [278] models higher-order joint relationships with hypergraph attention and incorporates structural encoding based on graph distances for skeleton-based action recognition. [283] extends the structural encoding with topological analysis beyond connectivity.	

1.1. Improving the Trade-Off between Sparsity and Multi-Modality in Attention

Background: The Softmax function is a fundamental component in modern machine learning, particularly within attention mechanisms [224], where it transforms input vectors into probabilistic scores. As a differentiable approximation of the Argmax operation, the entropy of the Softmax distribution is controlled by a scale factor, called temperature. We reveal that the expressivity of Softmax is severely limited by the trade-off between the sharpness and flatness of Softmax scores. For attention layers, a small temperature will cause relevant positions except the peak to be overlooked, whereas a large temperature will cause the distraction of attention on irrelevant keys. Sparse Softmax alternatives like Sparsemax [174] have been proposed to promote sparsity by assigning zero weights to less relevant inputs. However, these approaches are shown to further sacrifice the model's capacity to capture multiple relevant contexts simultaneously [35].

Contributions: To address this fundamental trade-off, we propose MultiMax [279] in Chapter III, which adopts a piecewise differentiable function that adaptively modulates attention distributions based on the input value range. Its piecewise differentiable nature ensures stable gradient-based optimization, facilitating seamless integration into existing models. MultiMax proves to extend the pareto frontier of the balance between sparsity and multi-modality compared to traditional SoftMax and its variants, leading to improved expressivity. Furthermore, it serves as a drop-in replacement for SoftMax, requiring no additional loss functions or significant architectural changes. Equipped with MultiMax, attention layers are shown to suppress irrelevant entries effectively while maintaining the ability to attend to multiple significant inputs concurrently. Empirical evaluations demonstrate that MultiMax enhances performance across various domains, including image classification, language modeling, and machine translation.

As the first author of [279], Yuxuan Zhou proposed the idea, derived all the proofs, implemented the code, conducted all the experiments, and served as the main writer of the paper. This paper is published at ICML 2024 and the code can be found at Github Repository.

1.2. Incorporating Geometric Prior into Attention for Image Modeling

Background: Vision Transformers (ViTs) [147, 63, 221] have recently achieved remarkable success in image classification tasks and established state-of-the-art results on the ImageNet benchmark. Compared to Convolutional Neural Networks (CNNs) [133, 96], they enjoy the merit of larger model capacity thanks to the global receptive field at each layer, but also suffer from slower convergence and potential overfitting, especially in low-data regimes, due to the lack of inherent spatial inductive biases. This deficiency arises because standard ViTs treat image patches as sequences, disregarding the two-dimensional spatial relationships crucial for understanding visual content. Positional embeddings are directly inherited from language transformers [224, 201, 98], which are in one-dimensional form and not tailored for capturing the complex structural patterns in images.

Contributions: To address these challenges, we introduce SP-ViT [282] in Chapter IV, with Spatial Prior-enhanced Self-Attention (SP-SA) as the core component, a novel mechanism that incorporates learnable 2D Spatial Priors (SPs) into the attention computation. Unlike fixed windows in CNNs, these spatial priors are learned during training, allowing the model to focus on relevant spatial relationships automatically, without imposing a preference for any hard-coded region in advance. This approach enhances the model's ability to capture local and global spatial dependencies, leading to improved performance in image classification tasks. Our proposed SPs are beneficial for general vision tasks. SPs are are compatible with various input sizes, as they are derived from relative coordinates between each pair of patches instead of their absolute positions.

As the first author of [282], Yuxuan Zhou proposed the idea, implemented the code except for the visualization using Transformer Explainability, conducted all the experiments, and served as the main writer of the paper. This paper is published at BMVC 2022 and the code can be found at Github Repository.

1.3. Incorporating Structural Prior into Attention for Skeleton-Based Action Recognition

Background: Skeleton-based action recognition [249, 208, 176] requires modeling the human body's structural relationships to understand complex movements. Recent methods often treat joints and their natural connections as nodes and edges of a graph, and employ a GCN [125] on such a predefined graph to learn joint interactions. Since then, GCNs have become the de facto standard of choice for skeleton-based action recognition. In GCNs, the adjacency matrix defining joint connections is fixed after training, which can lead to suboptimal representations, as the learned adjacency matrix may not accurately reflect the unique joint co-occurrences in different actions. Therefore, State-of-the-art GCNs [42, 38] heavily rely on attention mechanisms to relax the restriction of the fixed topology. Nevertheless, the performance gains are accompanied by increased complexity and computational overhead. Recent studies [176, 207] have attempted to adopt Transformers for this task, but their performance still lags far behind that of GCNs. We reveal that the permutation equivalent attention operation is agnostic to the bone connectivity between human body joints, and simple positional embeddings are incapable of capturing the complex structural information of skeleton data. Furthermore, both GCNs and Graph Transformers have a common limitation of assuming pairwise joint relationship, which overlooks the higher-order dependencies, which are beneficial for complex action understanding.

Contributions: To address these limitations, we propose the Hypergraph Transformer [278] in Chapter V, which is built on Hypergraph Self-Attention (HyperSA), a novel self-attention mechanism that models higher-order kinematic dependencies by incorporating hyperedges connecting multiple joints. This approach captures intricate joint interactions beyond pairwise connections. Additionally, the model incorporates a relative positional encoding based on graph distances to retain connectivity information during training, allowing the model to adaptively incorporate the unique structural information of skeletal graph. The resulting Hyperformer model outperforms existing methods on benchmarks like NTU RGB+D and Northwestern-UCLA datasets, demonstrating superior accuracy and efficiency in action recognition tasks. This advancement provides a more comprehensive understanding of human actions by capturing complex joint interactions and preserving skeletal connectivity.

As the first author of [278], Yuxuan Zhou proposed the core idea, implemented the codebase, conducted all experiments, and served as the primary author of the manuscript. The paper has received approximately 100 citations to date and has been followed up by a diverse range of works across multiple domains [34, 184, 160]. The accompanying implementation is available at the GitHub repository, which has also garnered around 100 stars.

In addition to the graph distance encoding, we further propose a novel *topological encoding* method in Chapter VI, which integrates topological descriptors [68, 271] into the latent representation. This design enables the model to capture higher-order skeletal topology beyond connectivity, providing a more holistic understanding of self-organizing dynamics. Equipped with our proposed topological encoding, the GCN is shown to eliminate the need for additional attention mechanisms and hypergraph modeling, while still capturing complex skeletal relationships. Moreover, we introduce BlockGC, a novel Graph Convolution layers with block-diagonal weight matrix. Our complete model, termed *BlockGCN* [283], outperforms existing approaches across all categories, achieving state-of-the-art results with significantly fewer parameters and lower computational cost.

As the first author of [283], Yuxuan Zhou proposed the central idea, implemented the majority of the codebase (excluding the topological encoding components), conducted all experiments apart from those related to topological encoding, and served as the lead writer of the manuscript. The paper is published at CVPR 2024 and has received approximately 70 citations to date and has inspired follow-up research across diverse domains [148, 143]. The implementation is publicly available at the GitHub repository, which has also attracted around 100 stars.

1.4. Overcoming the Error-Enhancement Defect in Label Smoothing for Image Classifiers

Background: Despite advancements in attention mechanisms, optimizing output probabilistic scoring remains a challenge. For classifier outputs, regularization [216, 129, 287] is necessary to prevent overconfidence and improve the reliability of predictions. Label Smoothing (LS), a widely adopted technique, aims to address this by assigning soft targets to ground-truth labels, thereby preventing the model from becoming overly confident. However, recent findings challenge this conventional wisdom. LS has been shown to cause a collapse in feature representation [161], degrade transfer learning performance [127], and, paradoxically, reinforce incorrect predictions with high confidence [284].

Contributions: To address these issues, we propose MaxSup [281] in Chapter VII, a novel regularization strategy that penalizes the top-1 predicted logit instead of the ground-truth logit, regardless of correctness. This key design eliminates the reliance on label knowledge during regularization and applies consistent penalty to both correct and incorrect predictions. A central contribution of this work is the discovery that the conventional LS formulation introduces an "error-enhancing" term, which inadvertently penalizes the ground-truth logit even when the model's prediction is incorrect. MaxSup avoids this issue entirely by shifting the regularization focus to the model's own top-1 prediction, thereby preserving more discriminative feature representations and significantly improving prediction calibration. Empirical evaluations across diverse benchmarks demonstrate that MaxSup not only outperforms LS in classification performance, but also enhances feature diversity and transferability.

As the first author of [281], Yuxuan Zhou proposed the core idea, derived all theoretical results except for the gradient analysis component, and implemented the majority of the codebase—excluding parts related to online label smoothing [263] and feature visualization [161]. He conducted all experiments except for the convolutional neural network–based classification comparisons and feature representation analyses, and served as the primary author of the manuscript. The paper was accepted as an Oral presentation at NeurIPS 2025. The accompanying code is available at the GitHub repository.

1.5. Balancing Diversity and Risk in Sampling-Based Decoding for Large Language Models

Background: In generative tasks, such as open-ended text generation [70, 156], additional challenges arise. Sampling directly from the Softmax distribution can produce incoherent outputs, while deterministic decoding strategies like greedy decoding or beam search often lead to repetitive or overly conservative results [108]. Achieving the right balance between diversity and coherence in generated text requires more refined sampling strategies that account for both model confidence and the specific risk profile of the task.

Contributions: This work introduces a systematic framework [280] in Chapter VIII for the comprehensive comparison of existing sampling methods for large language models [108, 285, 104, 14], and provides practical user guidelines for parameter selection. To this end, parameter selection is guided by the expected trade-off between diversity and coherence in generated text, as estimated using our proposed context-preserving prefix tree. Since each decoding method requires a specific parameter setting to attain a desired level of coherence, we compare methods at equivalent points along their respective trade-off curves. The evaluation results are no longer independent of parameter tuning, thus enabling a robust and fair comparison. Furthermore, the provided guidance on parameter selection helps users balance exploration and exploitation during inference, resulting in responses that are both diverse and contextually appropriate.

As the first author of [280], Yuxuan Zhou proposed the core idea, implemented the complete codebase, conducted all experiments, and served as the primary author of the manuscript. The paper was accepted to the ACL 2025 Main Proceedings and has inspired follow-up research [7], as well as contributed to the development of a new sampling-based decoding approach [164]. The accompanying implementation is publicly available at the GitHub repository.

2. Publications

The following first-authored papers contribute to this thesis:

- [278] **Zhou, Y.**, Cheng, Z.-Q., Li, C., Fang, Y., Geng, Y., Xie, X., & Keuper, M. (2022). Hypergraph Transformer for Skeleton-Based Action Recognition. arXiv preprint arXiv:2211.09590.
- [279] **Zhou, Y.**, Fritz, M., & Keuper, M. (2024). MultiMax: Sparse and Multi-Modal Attention Learning. 41st International Conference on Machine Learning (ICML 2024).
- [280] **Zhou, Y.**, Keuper, M., & Fritz, M. (2024). Balancing Diversity and Risk in LLM Sampling: How to Select Your Method and Parameter for Open-Ended Text Generation. Proceedings of the Association for Computational Linguistics (ACL 2025).
- [281] **Zhou, Y.**, Li, H., Cheng, Z.-Q., Yan, X., Fritz, M., & Keuper, M. (2025). MaxSup: Overcoming Representation Collapse in Label Smoothing. 39th Annual Conference on Neural Information Processing Systems (NeurIPS 2025 Oral).
- [282] **Zhou, Y.**, Xiang, W., Li, C., Wang, B., Wei, X., Zhang, L., Keuper, M. and Hua, X. (2022). SP-ViT: Learning 2D Spatial Priors for Vision Transformers. 33rd British Machine Vision Conference (BMVC 2022).
- [283] **Zhou, Y.**, Yan, X., Cheng, Z.-Q., Yan, Y., Dai, Q., & Hua, X.-S. (2024). BlockGCN: Redefine Topology Awareness for Skeleton-Based Action Recognition. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2024).

Chapter II.

Background

Before formally introducing our work, we review the key technical foundations relevant to this thesis. A central theme of this work is the concept of *neural probabilistic scoring*—the mechanism by which neural networks assign likelihoods or confidence over input relevance or output predictions. This concept plays a critical role in a wide range of deep learning systems. In particular, the probabilistic scoring performed in attention mechanisms and output layers has proven to be both a cornerstone of recent successes—such as image and action recognition [63, 221, 8], as well as large language models [223, 2, 118]. The attention mechanism computes context-dependent relevance between tokens, patches, or nodes via a normalized score distribution, while the output layer uses a softmax distribution to model predictive uncertainty over discrete outputs. These components are not only central to the success of large language models and vision transformers, but also represent key bottlenecks in expressiveness, structural modeling, and output reliability.

Transformers serve as a natural foundation for this investigation due to their centrality in modern deep learning and their remarkable generality: they provide a unified architecture that can be seamlessly applied across diverse data modalities, including sequences, grids, and graphs. This modality-agnostic design enables attention-based models to adapt to natural language [156, 151], images [54], and structured motion data such as human skeletons[144], using a shared set of principles for learning relationships and making predictions.

This thesis investigates and proposes improvements to both of these mechanisms. Specifically, we study the limitations of the softmax attention formulation and its impact on contextual modeling in both discriminative [96] and generative models [223]. We also address challenges in the output layer—ranging from label smoothing [216] in image classification to sampling-based decoding [108, 70, 14, 154, 104] in open-ended text generation [156, 70]—where current probabilistic approaches often struggle to balance efficacy and reliability. Applications of our work span language, image, and skeleton data modalities, where the trade-off between sparsity and multi-modality in the softmax distribution, as well as lack of structural awareness or sampling robustness motivate the need for more expressive probabilistic scoring.

We begin by reviewing the Transformer architecture and its applications across different data modalities, followed by a discussion on label smoothing and its impact on output reliability, as well as the limitations of sampling-based decoding strategies in language models.

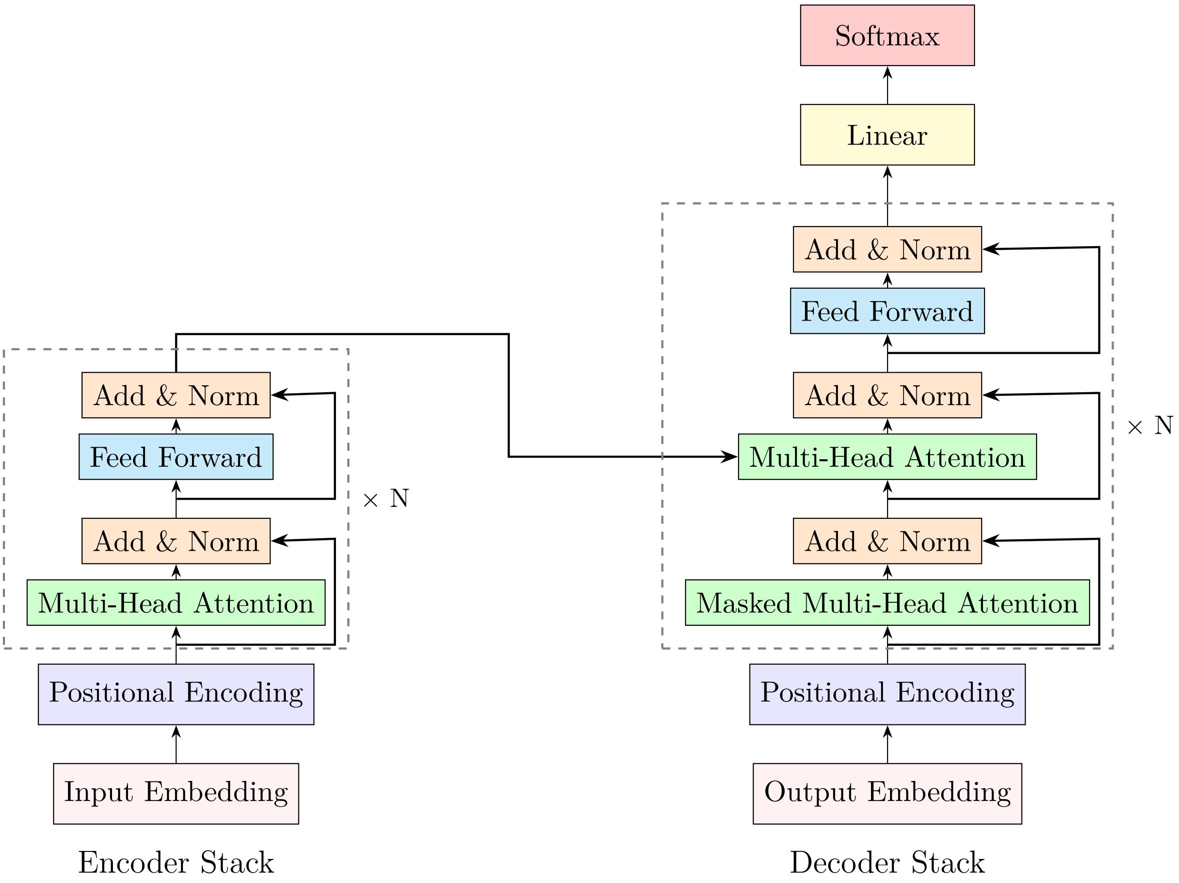


Figure II.1.: Transformer architecture with repeated encoder and decoder layers.

1. Transformer Fundamentals

The Transformer model was first introduced by Vaswani et al. [224] for natural language processing tasks such as machine translation [18], and has since revolutionized deep learning, with remarkable achievements in a wider range of areas such as computer vision and graph-based modeling. A key factor behind its widespread success is its modality-agnostic architecture: the same core building blocks—attention mechanisms and feed-forward layers—can be applied with minimal modification to sequential, spatial, and structured data alike, offering a unified framework across diverse learning domains.

Compared to CNNs, which heavily exploit local correlations between neighboring pixels through inductive biases, transformers are designed with minimal hard-coded prior knowledge. While they typically converge more slowly, transformers can achieve higher representational capacity when trained on sufficiently large datasets. **The scalability of transformer underpins many of the recent advances in deep learning** [126, 2, 179].

Attention The core component of the transformer model is the scaled dot-product attention:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V, \tag{II.1}$$

where Q, K, and V are the query, key, and value matrices respectively, and d_k is the key dimension.

Multi-Head Attention To enhance the model's ability to capture information from different representation subspaces, the Transformer employs *multi-head attention*, which runs several self-attention mechanisms (or "heads") in parallel:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O, \tag{II.2}$$

where each head is computed as:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V), \tag{II.3}$$

with learnable projection matrices W_i^Q, W_i^K, W_i^V, and W^O.

Masked Multi-Head Attention In autoregressive generation tasks such as language modeling, a variant called masked multi-head attention is used to prevent the model from attending to future tokens. This is implemented via a masking matrix M:

$$\text{MaskedAttention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}} + M\right)V, \tag{II.4}$$

where M assigns $-\infty$ to positions corresponding to future tokens, ensuring causal decoding.

Positional Encoding Since the attention mechanism is inherently permutation equivalent, the Transformer must inject positional information to preserve sequence order. This is done using positional encodings added to the input embeddings:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right), \tag{II.5}$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right), \tag{II.6}$$

where pos is the token position and i is the dimension index. These sinusoidal encodings allow generalization to longer sequences. Alternatively, learnable positional embeddings are often used in practice. Follow-up works also found that relative positional encodings [201, 147] outperform absolute positional encodings, because of their larger capacity to capture spatial relations.

As shown in Figure II.1, each Transformer layer consists of this multi-head attention mechanism followed by a position-wise feed-forward network, with both sublayers wrapped in residual connections and layer normalization [11]. The overall Transformer model is constructed by stacking multiple such layers, enabling the network to capture increasingly abstract and hierarchical representations across the input sequence.

1.1. Transformer Architectures

Encoder-Only Models These architectures encode the entire input into contextual representations in parallel, making them suitable for classification and representation learning tasks (e.g., BERT [56], ViT[63], and MAE[95]). They are commonly used in vision and skeleton-based action recognition.

Decoder-Only Models These models generate output tokens autoregressively by attending only to previously generated tokens. They are widely used in language modeling and text generation tasks (e.g., GPT series). Masked attention ensures that predictions are conditioned solely on prior context.

Encoder-Decoder Models This is the original Transformer architecture, comprising an encoder that processes the input and a decoder that generates output while attending to the encoder's representation. Although encoder-decoder models such as T5 [182] and BART [135] continue to show strong performance in tasks like summarization and translation, and have been widely studied, they have become less dominant in large-scale pretrained systems, where decoder-only architectures [2, 223] are favored for their simplicity and scalability in autoregressive generation.

1.2. Applications

Language Modeling Decoder-only models generate text by predicting the next token given previous context, as shown in Figure II.2. These are the basis for large language models (LLMs).

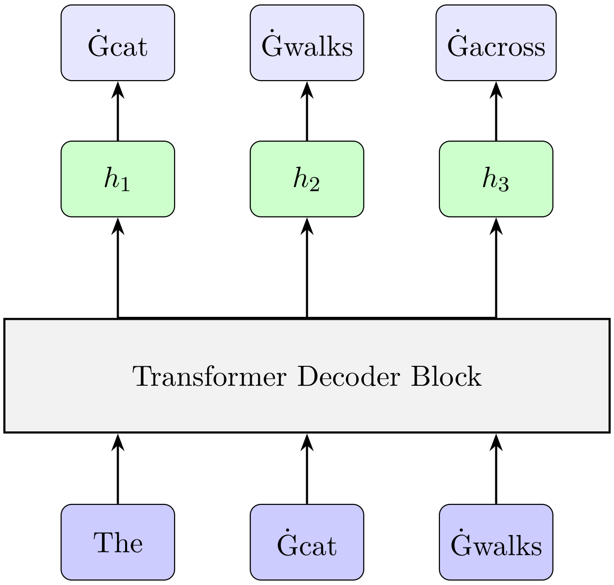


Figure II.2.: Decoder-only Transformer for autoregressive language modeling with subword tokenization. The prefix "The cat walks" is given as input and they are first reorganized as tokens, e.g., the whitespace after "The" and the word "cat" are regarded as a single token "Ġcat". The model predicts the next token at each position.

Image Classification In Vision Transformers (ViTs), images are transformed into a format compatible with transformers by dividing them into smaller patches, as shown in Figure II.3. The procedure is as follows:

- Dividing the Image into Patches: To align with the input format of transformers, the image is divided into smaller, non-overlapping patches, typically of size 16x16 pixels. Each patch is treated as a token in the sequence, analogous to how words or subwords are tokenized in natural language processing.
- Flattening the Patches: After dividing the image into patches, each patch is flattened into a 1D vector. This vector serves as the input token for the transformer model,

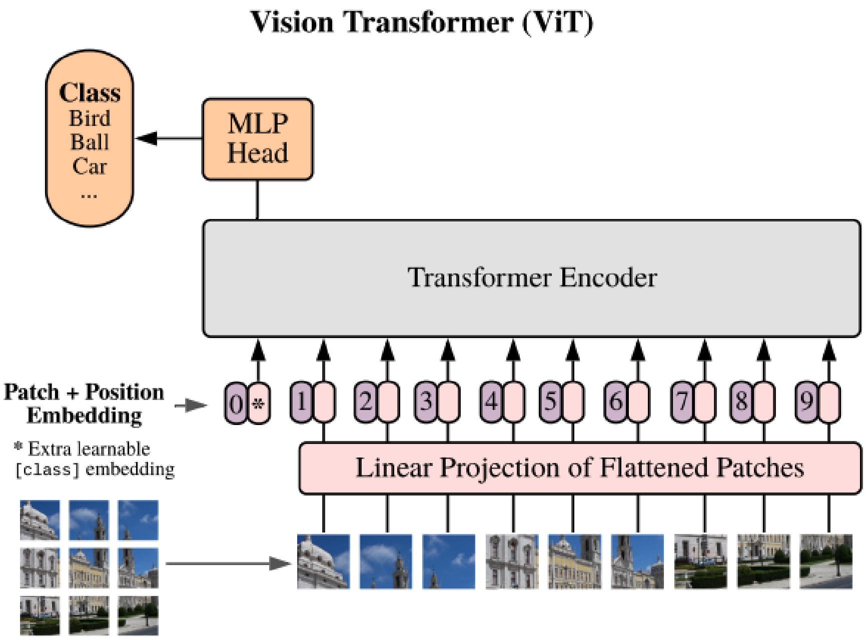

Vision Transformer (ViT)
Class
Bird
Ball
Car
...
MLP Head
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
0
*
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches

Figure II.3.: Illustration of Vision Transformer (Figure from [63]).

creating a sequence of tokens that represent the entire image. These tokens are then passed through the model in a similar manner to tokenized words in text.

While this tokenization procedure effectively transforms the image into a sequence that the transformer model can process, the spatial relationship between the patches is lost during the flattening process. Although positional encodings are added to attempt to capture spatial information, this method does not fully preserve the complex spatial relationships between patches, which may limit the model's ability to understand fine-grained spatial patterns.

Skeleton-Based Action Recognition This involves classifying human actions based on sequences of joint positions, representing the human skeleton. As shown in Figure II.4, the input is represented as a spatio-temporal graph $G = (V, E)$, where nodes represent joints and edges represent spatial or temporal links. Given a sequence $\mathbf{X} = \{x_1, \ldots, x_T\}$ with $x_t \in \mathbb{R}^{J \times d}$ (for J joints and feature dimension d), the spatial and temporal relationships are often modeled in an alternating fashion, as the temporal dimension often contains redundancy and entails heavy computational costs. Despite efforts to adapt Transformers for skeleton-based action recognition [176, 207], these models still fall short of the state-of-the-art performance achieved by GCNs or hybrid approaches. A major limitation is the scarcity of labeled data in this domain, which constrains the scalability and generalization of Transformer models. In this context, effectively incorporating the complex skeletal topology into Transformer architectures remains a critical challenge.

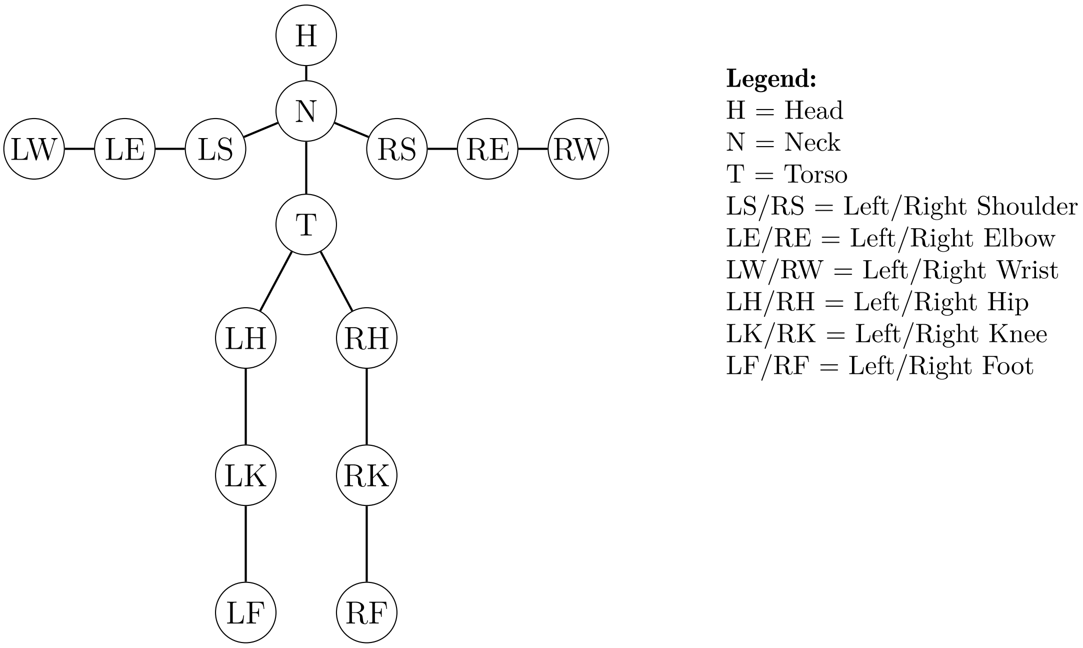


Figure II.4.: Illustration of Human skeleton data, where each joint location is recorded by a 3D coordinate.

2. Sampling-Based Decoding Methods

Sampling-based decoding techniques [108, 70] offer alternatives to traditional deterministic decoding methods like greedy search and beam search [75]. These techniques allow for more diversity and creativity in generating text by introducing randomness into the token selection process. In contrast to greedy decoding, which picks the most probable token at each step, and beam search, which keeps track of multiple hypotheses and evaluates a fixed number of the most probable sequences, sampling methods explore a broader set of possible continuations. This results in less repetitive and potentially more interesting output. However, sampling introduces its own set of challenges, such as balancing diversity and coherence in the generated text.

Limitations of Greedy and Beam Search Greedy decoding selects the most probable token at each time step, resulting in deterministic outputs. Beam search, on the other hand, maintains multiple hypotheses to approximate the most probable overall sequence. However, despite its broader search space, beam search often converges on similar high-probability paths and tends to produce degenerate text — including repetitive loops and incoherent outputs [234, 155].

Truncation Sampling Techniques Truncation sampling methods are designed to focus on a limited set of candidate tokens by cutting off those with lower probabilities. By truncating the distribution, the model avoids selecting tokens that are highly unlikely, thereby improving both the efficiency and quality of the generated text. Truncation restricts the number of choices at each step, balancing the need for diversity in the generated output with the need for coherence and relevance.

Most of the truncation sampling methods fall under a general truncation scheme:

$$P_{\text{Trunc}}(x_t \mid \boldsymbol{x}_{<t}) = \begin{cases} \frac{P_\theta(x \mid \boldsymbol{x}_{<t})}{Z_{\boldsymbol{x}_{<t}}} & \text{if } x \in \mathcal{A}_{\boldsymbol{x}_{<t}}, \\ 0 & \text{otherwise,} \end{cases} \tag{II.7}$$

where $\mathcal{A}_{\boldsymbol{x}_{<t}}$ is the allowed set of tokens at time step t, defined according to a truncation strategy (e.g., Top-k, Top-p). $P_\theta(x \mid \boldsymbol{x}_{<t})$ is the base model probability, and $Z_{\boldsymbol{x}_{<t}} = \sum_{x \in \mathcal{A}_{\boldsymbol{x}_{<t}}} P_\theta(x \mid \boldsymbol{x}_{<t})$ is the normalizing constant to ensure the probabilities sum to 1 over the allowable set.

Two common strategies that instantiate this general framework are Top-k and Top-p sampling:

Top-k Sampling In Top-k sampling, the allowable set $\mathcal{A}_{\boldsymbol{x}_{<t}}$ consists of the k most probable tokens under the model's distribution at each time step:

$$\mathcal{A}_{\boldsymbol{x}<t}^{\text{Top-}k} = \text{Top-}k(P_\theta(x \mid \boldsymbol{x}_{<t})). \tag{II.8}$$

Top-p (Nucleus) Sampling In Top-p sampling, the allowable set includes the smallest number of tokens whose cumulative probability exceeds a threshold p:

$$\mathcal{A}_{\boldsymbol{x}<t}^{\text{Top-}p} = \left\{ x \in \mathcal{V} \;\middle|\; \sum_{x' \in \text{ranked-prefix}(x)} P_\theta(x' \mid \boldsymbol{x}_{<t}) \leq p \right\}, \tag{II.9}$$

where ranked-prefix(x) denotes the set of tokens with probability greater than or equal to that of x, i.e., all tokens ranked higher than or equal to x in descending order of $P_\theta(\cdot \mid \boldsymbol{x}_{<t})$.

More advanced adaptive sampling methods, such as typical sampling [154] and Mirostat [14], aim to improve the trade-off between coherence and diversity by dynamically adjusting the allowed set. Typical sampling filters tokens based on how close their information content (negative log-probability) is to the expected entropy of the distribution, encouraging outputs that are statistically "typical" rather than overly generic or rare. Mirostat, on the other hand, maintains a target level of surprise (measured in bits) by continuously adjusting internal parameters to regulate the entropy of generated text. While these methods are shown to produce more coherent and contextually appropriate outputs, it can be difficult in practice to choose the most suitable method and, more importantly, to determine the appropriate hyperparameters (e.g., entropy targets or typicality thresholds) that align with specific generation goals. As a result, these challenges have hindered the widespread adoption of advanced sampling techniques in practical applications.

Part I.

Part One: Probabilistic Scoring in Attention

Attention mechanisms have become fundamental components in modern machine learning architectures, enabling models to selectively focus on relevant parts of the input and thereby capture complex dependencies efficiently. Their success across diverse domains—ranging from natural language processing to computer vision—hinges on their ability to dynamically weight inputs based on context. However, the most widely used form, Softmax attention, exhibits inherent limitations that constrain its effectiveness.

Softmax attention struggles with the trade-off between sparsity and multi-modality, often failing to capture multiple relevant inputs simultaneously. As a smooth approximation of argmax, it inherently promotes unimodal distributions, which tend to highlight a single dominant input while suppressing others, even when multiple are semantically important — a limitation particularly problematic in domains like vision and language where attending to multiple contextual regions or tokens is necessary. Attempts to adjust the temperature to control focus reveal a fundamental trade-off: lowering the temperature sharpens the distribution, enhancing focus but suppressing secondary modes; increasing it allows for more distributed attention but at the cost of amplifying noise and reducing selectivity. Moreover, its permutation-invariant nature disregards crucial spatial or structural relationships in data, limiting its effectiveness in domains like vision and structured action recognition, where understanding relative position, continuity, and hierarchical organization is essential for modeling complex patterns and behaviors.

To address these limitations, we examine alternative attention formulations that aim to overcome the unimodal bias, improve robustness to noise, and incorporate structural inductive biases better suited for tasks requiring multi-context awareness.

Chapter III.

Improving the Trade-Off between Sparsity and Multi-Modality in Attention

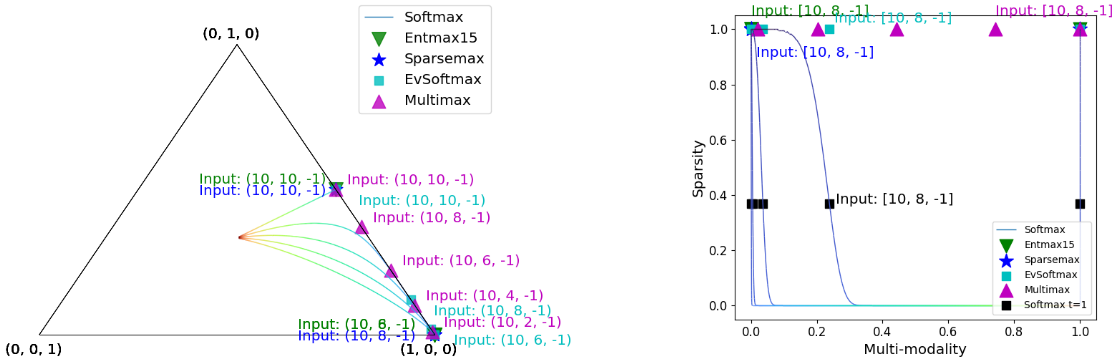
Attention mechanisms are central to modern deep learning in NLP and computer vision, selectively weighting inputs to balance focus and flexibility. However, Softmax-based attention faces a fundamental trade-off between sparsity—sharply attending to few inputs—and multi-modality—capturing multiple relevant signals—limiting expressivity and robustness.

The content of this chapter corresponds to our established work [279], which introduces MultiMax, a novel probabilistic scoring function that extends the Pareto frontier between sparsity and multi-modality. MultiMax adaptively balances these competing objectives, enabling attention to capture multiple important contexts without losing sharpness. It serves as a drop-in Softmax replacement that improves expressivity and robustness, constituting a key contribution of this thesis.

1. Introduction

The SoftMax has remained in wide use in modern machine learning methods and finds its application in a variety of algorithms such as multi-class classification [133, 87, 17], attention mechanisms [224, 225, 12, 82] and reinforcement learning [215, 188, 237]. It can be regarded as a differentiable approximation of the Argmax operation and projects the input onto the probability simplex, which allocates most of the probability mass to large entries. From the perspective of optimization, the SoftMax function allows for a reasonable trade-off between exploitation and exploration [236], i.e., important positions are emphasized while every position has a chance of being explored. This trade-off can be controlled by a scale factor, which is often referred to as temperature.

However, the expressivity of SoftMax is severely limited by the following dilemma: a high temperature leads to over-smoothing and reduces the efficiency of the optimization, whereas a small temperature collapses multi-modality and makes training unstable. In attention layers for example, a small temperature will cause relevant positions except the peak to be overlooked, whereas a large temperature will "waste" a non-negligible portion of attention on irrelevant keys. Therefore, temperature is often set to one by default in attention layer. As shown later, such a compromise also results in the recently observed over-smoothing issue in both vision [85, 229] and language [206] transformers. Moreover, transformer-based Large Language Models are shown to be prone to the interference of irrelevant context [205, 117], which is also highly related to the portion of attention on irrelevant tokens [235].



(a) SoftMax output depends on the temperature, which we show by the color coding from dark blue (low temperature) to red (high temperature). Sparse SoftMax variants collapse multi-modality, while MultiMax successfully produces approximately **sparse** and **multi-modal** distributions.

(b) SoftMax and its sparse extensions are limited by the trade-off between sparsity and multi-modality, which is improved by our MultiMax.

Figure III.1.: We evaluate SoftMax, SparseMax, EntMax, EvSoftMax and MultiMax (using the parameters of a hidden layer MultiMax trained on ImageNet directly) functions on a series of example input points $\boldsymbol{v} \in \mathbb{R}^3$ and project the resulting distribution on a simplex Δ^2. Informally, the interior of the simplex stands for trimodal distributions, the edges constitute the set of bimodal distributions, and the vertices are unimodal distributions. Notably, the above figures highlight the advantage of MultiMax's multi-modality. EntMax, Sparsemax and SoftMax with small temperature (blue colored line) yield a (quasi) unimodal distribution, which ignore the second largest entry. In contrary, SoftMax with higher temperatures (green and orange colored line) fails to ignore the negative entry.

To overcome the issue, previous works have proposed sparse SoftMax alternatives, which allow to completely ignore small entries below a threshold. These sparse SoftMax variants have been studied in diverse contexts, e.g., generative modeling [35], output activations of multi-class classifiers, and/or attention mechanisms [174, 152, 92].

However, such methods often suffer from poor gradient signal, which leads to instability during training. Moreover, the number of non-sparse dimensions is often treated as empirically selected hyperparameter.

In contrast to sparsity, multi-modality has been less discussed in the previous studies. Since attention is not supposed to be exclusive in most cases, the vanilla SoftMax, as an approximation of Argmax, does not easily comply with multi-modality. The sparse alternatives [152, 174, 131] to SoftMax have even a larger tendency to not preserve the multi-modality of distributions [115].

In this chapter, we propose MultiMax as an alternative to SoftMax. MultiMax allows for learning when to emphasize sparsity and when to emphasize multi-modality, offering a flexible trade-off between both. At the same time, it remains piecewise differentiable such as to allow for stable gradient-based optimization.

Specifically, MultiMax extends the traditional SoftMax by a preceding parameterized function that enables to learn distinct temperature values for particular input value ranges separately. Used within a self-attention mechanism, this facilitates for example to learn particularly low temperatures that induce sparsity for low input value ranges, i.e. unrelated tokens can be ignored, while learning high temperatures for higher input value ranges, i.e. several related tokens can share the attention in a multi-modal way. The improved multi-modality and sparsity brought by MultiMax is demonstrated in Fig. III.1. MultiMax is able to serve as a drop-in replacement of SoftMax in any applications and adapt to an appropriate form via training.

After a theoretic analysis, we show empirically that MultiMax can improve the attention mechanism and is an effective classifier output activation as well. MultiMax consistently improves over SoftMax baselines in a wide range of tasks, with an increase of 0.6% classification accuracy on ImageNet, an improve of 0.7 in perplexity for language modeling on WikiText-103, and a gain of 0.3 in BLEU score for English to German translation on WISLT-2014.

The contributions of this chapter are as follows:

- We generate insights in the trade-off between sparsity and multi-modality in SoftMax.
- We propose MultiMax – an alternative to SoftMax with better and learnable tradeoffs between both, multi-modality and sparsity.
- We show advantageous properties of MultiMax theoretically and demonstrate performance improvements on diverse tasks ranging from image classification over language modeling to machine translation.

2. Related Work

We organize the related work by first discussing related SoftMax alternatives afterwards more broadly approaches that have aimed to improve attention mechanism as well as prevent oversmoothing.

SoftMax alternatives. In previous work, huge efforts have been made to pursue sparsity. Sparsemax [152] and its generalization EntMax-α [174] are sparse SoftMax variants through thresholding the output probability. Although the hyperparameter α is supposed to control the degree of sparsity, the functions lack full support for $\alpha > 1$. Another variant, in principle similar to EntMax-1.5, with control of the sparsity is Sparsehourglass [131]. As output activation of a classifier, these approaches require alternative losses to enable gradient-based optimization. Yet, this can cause slow convergence and training instability as well as an additional approximation error. Ev-SoftMax [35] additionally reveals that these sparse SoftMax variants could harm multi-modality. It achieves sparsification by zeroing out input entries smaller than average and provides a training-time modification strategy to enable gradient-based training. This is indeed similar to the broadly adopted top-k selection of SoftMax output, e.g., in attention layers of vision [230, 270] and language [92] transformers. In contrast, our MultiMax achieves sparsity and improved multi-modality at the same time without extra hyperparameters. It has also full support and thus is a drop-in replacement of SoftMax in any context.

Anti-oversmoothing approaches. Over-smoothing refers to the issue that the representations of different tokens tend to become more similar as layer depth increases. This problem is observed in both vision [229, 85] and language transformers [206]. Patch Diversification [229] combines three regularization losses to explicitly encourage diversity in patch representations. AttnScale [229] decomposes a self-attention block into low-pass and high-pass components, and rescales the high-pass component of the self-attention matrix. While these remedies have been proposed, the reason behind lacks in-depth discussion. Notably, [206] has attempted an analysis by relating self-attention matrix to adjacent matrix of a graph. Their claim of post-normalization being the root cause has led to further discussion, as they stick to post-normalization in the end and pre-normalization empirically performs no better than post-normalization [100]. We find that the over-smoothing problem is indeed is comparable to over-smoothing problem in GCNs [33, 167], and strongly related to the inevitable amount of attention assigned to irrelevant tokens. The identity of each token degrades rapidly due to the repetitive attention operations. As shown in the studies of GCNs, sparsification [187, 94, 272] is a direct and effective solution.

Attention mechanism A vast amount of efforts have been invested in proposing new or improving the existing attention mechanisms [224, 225, 12, 82]. [124] successfully incorporated richer structural distributions into attention networks via graph encodings. [165] introduced a new framework for sparse and structured attention with a smoothed max operator, which can be regarded as a generalization of softmax and sparsemax. [55] considered variational attention networks as alternatives to soft and hard attention for better learning latent variable alignment models. [153] suggested to adopt sparse attention to selectively focus on relevant sentences in the document context for improved neural machine translation. [262] explored the feasibility of specifying rule-based patterns to sparsify encoder outputs for improved decoding efficiency. While these approaches mainly focus on improving sparsity, our MultiMax improves both multi-modality and sparsity at the same time. Moreover, MultiMax is a universal alternative to the SoftMax function, which is not limited to the application in the attention mechanism.

3. Background, Metrics, and Analysis

In this section, we state the challenge of sparsity-multi-modality trade offs in reweighting functions such as softmax. Based on metrics to measure these quantities, we provide a theoretical analysis that shows the tension between those two goals in previous formulations.

3.1. Background

SoftMax is the most widely adopted **reweighting function** in machine learning and is formulated as follows:

Definition 3.1. Let $\Delta^{K-1} = \{\boldsymbol{p} \in \mathbb{R}^K_{\geq 0} | \mathbb{1}^T \boldsymbol{p} = 1\}$ be the $K-1$ dimensional simplex. SoftMax maps a vector $\boldsymbol{x} \in \mathbb{R}^K$ with $K \in \mathbb{Z}_+$ to a proper distribution in Δ^{K-1}:

$$\phi_{SoftMax}(\boldsymbol{x})_i = \frac{e^{tx_i}}{\sum_{k=1}^K e^{tx_k}}, \tag{III.1}$$

where $\frac{1}{t}$ controls the entropy of the generated distribution and is often referred to as "temperature". The exponential term makes the distribution concentrated on the largest entries, which reflects the selective nature of for example the attention mechanism or multi-class classification.

3.2. Sparsity and Multi-Modality Trade-off

Although sparsity seems to be easily acquired by decreasing the temperature, we find that the gain of increased sparsity comes at a cost in practice. We exemplify such an issue by

Table III.1.: Classification accuracy on ImageNet1K using Deit-small baseline with Global Avarege Pooling (GAP) and classification token (CLS) respectively.

Model	Head	Temperature $\frac{1}{t}$					
		0.1	0.5	1	2	10	trainable
Deit-small	CLS	5.1	79.9	79.9	**80.0**	79.5	79.7
	GAP	4.7	80.3	**80.4**	80.0	79.9	80.2

comparing the classification performance of a transformer on ImageNet1K with different SoftMax temperatures in Table III.1. As shown in the table, tuning temperature is tedious and brings no obvious advantage. Moreover, a small temperature typically provides poor learning signal and can hamper training stability, as suggested by the low accuracy for temperature 0.1. For a better understanding of the inefficacy of temperature tuning, we follow-up with a brief theoretical study to show that the temperature tuning of SoftMax function is indeed limited by an inherent trade-off between sparsity and multi-modality.

To enable a precise analysis on the trade-off between multi-modality and sparsity, we need to define appropriate quantitative metrics for these two properties of reweighting functions.

Quantifying Multi-Modality and Sparsity of Reweighting Functions

For multi-modality and sparsity, the probabilities close to peak and zero are with no doubt the most relevant, respectively. And such relevance equivalently transfers to the largest and smallest input entries, since the studied reweighting (activation) functions should be monotonically non-decreasing [76, 77]. For simplification, we omit the trivial case when two entries are equal, since they remain equal after any valid function.

To quantitatively compare the **multi-modality** of the distributions generated by different reweighting functions ϕ w.r.t. a given input $\boldsymbol{x}$, we propose the following metric $\mathcal{M}(\boldsymbol{x})$:

Definition 3.2. Without loss of generality, let x_{max} be the largest entry and $x_{max} > x_n > \epsilon$, where ϵ could be any reasonable threshold for a entry to be considered relevant and N is the counts of such entries. The **Multi-Modality Metric** is given by:

$$\mathcal{M}(\boldsymbol{x}) = 1 - \frac{1}{N} \sum_{\epsilon < x_n < x_{max}}^{N} (\phi(\boldsymbol{x})_{max} - \phi(\boldsymbol{x})_n), \tag{III.2}$$

Intuitively, this metric captures the average difference between the reweighted relevant entries $\phi(\boldsymbol{x})_n \; \forall x_n > \epsilon$ and the maximum $\phi(\boldsymbol{x})_{max}$. The average distance would be close to 0, if all output entries are about the same (maximum multi-modality). In order to make it a large=better metric, we subtract it from 1.

Analogously, we build a **Sparsity Metric** for the reweighting functions upon the common $-L_\epsilon^1$ sparsity metric for vectors [113], which calculates the negative sum of entries smaller than ϵ. Although sparse or non-sparse is a binary status, a smooth metric is desired to additionally consider values close to zero (i.e. approximately sparse). Moreover, we would like to take the non-linear nature of such sparsity into account, i.e., above a reasonably small threshold, a large portion of the range from 0 to 1 is supposed to be non-sparse. In this case, a non-linear scaling (especially an approximation of a step function) helps to better reflect the actual degree of sparsity. Thus, we define the sparsity metric as follows:

Definition 3.3.

$$\mathcal{S}(\boldsymbol{x}) = \frac{1}{L}\sum_{x_l<\epsilon}^{L} \exp\left(\frac{s-\phi(\boldsymbol{x})_l}{s} - 1\right), \tag{III.3}$$

where $s \in [0,1]$ can be any reference value for a non-linear scaling of the sparsity score and L is the counts of entries smaller than ϵ. For example, the probability of the smallest entry x_{min} after SoftMax ($\underset{t=1}{\text{SoftMax}}(\boldsymbol{x})_{min}$) can be chosen as a reasonable reference value. Together with the exponential term, $\mathcal{S}(\boldsymbol{x})$ results in a smooth approximation of a step function, with the output range normalized to $[0,1]$, where larger values indicate stronger degrees of sparsity. Having defined the two metrics, we are able to prove there exists a trade-off between them.

Proofing the Trade-off

Lemma 3.4. *$\mathcal{S}(\S)$ is monotonically decreasing w.r.t. $\phi(\boldsymbol{x})_l$. (See Appendix 2 for the proof.)*

This can be easily proved by checking the partial derivative. Similar proof can be done for the following:

Proposition 3.5. *For a given input $\boldsymbol{x}$, the following statements hold w.r.t. temperature t.*

(i) *Multi-modality of SoftMax is monotonically increasing.*

(ii) *Sparsity of SoftMax is monotonically decreasing for $\epsilon \leq \frac{\|\boldsymbol{x}\|_1}{K}$.*

(See Appendix 2 for the proof.)

It is clear that we could increase either multi-modality or sparsity by simply varying temperature, but at the cost of decreasing the other. As a remedy, we suggest a piece-wise modulation scheme, which modulates small and large entries via two corresponding temperatures independently.

4. MultiMax

Based on our insights in the trade-off between sparsity and multi-modality in SoftMax, we propose MultiMax that reconciles those two objectives in a learnable formulation. We start by defining MultiMax that introduces two temperature terms that control for sparsity and multi-modality respectively. We analyze improved properties that are achieved by this formulation and finally extend the concept to higher order polynomials and beyond attention mechanisms.

The following sections will provide a theorectic analysis of MultiMax, starting with its first-order form.

4.1. First-order MultiMax

Definition 4.1. Let b and d be two control parameters. We apply two corresponding temperatures t_b and t_d only to the entries smaller than b and larger than d, respectively.

We construct a piece-wise linear function σ to modulate the SoftMax input $\boldsymbol{x}$, which defines the proposed MultiMax:

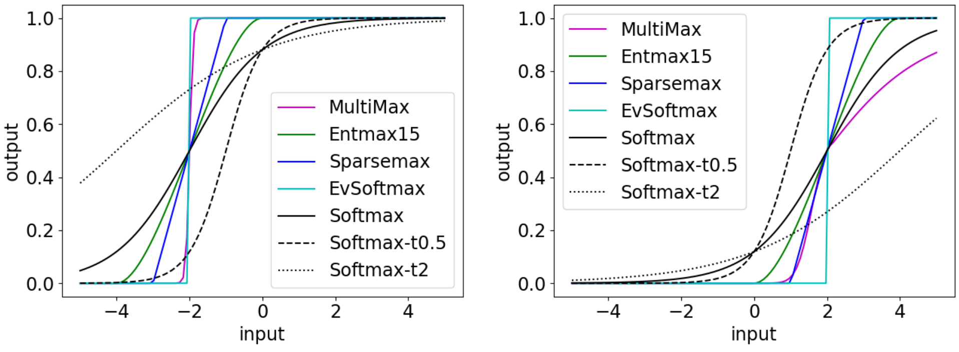
$$\begin{aligned} \phi_{_{MultiMax}}(\boldsymbol{x})_i &= \frac{\exp\left(\sigma(x_i)\right)}{\sum_{k=1}^{K}\exp\left(\sigma(x_k)\right)}, \quad \text{where} \\ \sigma(x) &= x + \underbrace{(1-t_b)Max(b-x,0)}_{\text{term(1)}} + \underbrace{(t_d-1)Max(x-d,0)}_{\text{term(2)}}, \end{aligned} \tag{III.4}$$

We call the above function the first-order MultiMax function and we will generalize it to a higher-order version towards the end of this section. For now, the first-order MultiMax has an intuitive interpretation:

$$\sigma(x) = \begin{cases} t_b x + (1-t_b)b & x < b \\ x & b \leq x \leq d \\ t_d x + (1-t_d)d & x > d \end{cases}, \tag{III.5}$$

where the bias terms $(1-t_b)b$ and $(1-t_d)d$ guarantees continuity of the modulator, e.g., $\lim_{x\to b^-}\sigma(x) = \lim_{x\to b^+}\sigma(x) = b$. To guarantee differentiability, subgradients can be defined for the turning points, e.g., $d\sigma(x)/dx = 1$ at $x = b$, please refer to [19] for more details. For $t_b > 1$ and $0 < t_d < 1$, we could prove that MultiMax achieve a better balance between multi-modality and sparsity than SoftMax. Intuitively, a large t_b pushes small entries closer to zero, while a small t_d reduces the gap between large entries. Therefore, the output distribution is modulated to exhibit higher sparsity as well as multi-modality.

To disclose the mechanism behind, we first study the impact of modulating only the small entries on the output distribution. Then we show that additionally modulating the large entries increases multi-modality further.



(a) Input point [-2, x].

(b) Input point [2, x].

Figure III.2.: Illustration of different reweighting functions in the two-dimensional case. It can be seen clearly that MultiMax weigh the entries at small and large value ranges in a different manner, thus it does not suffer from the trade-off between sparse and multi-modal.

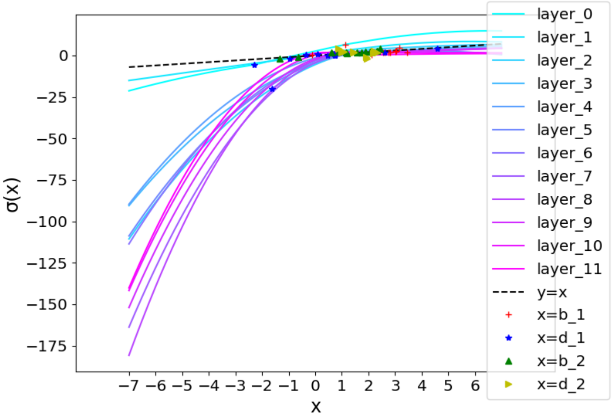


Figure III.3.: The learned modulator functions σ (Eq. (III.6)) at each layer, comparing to identity mapping of the SoftMax input $\boldsymbol{x}$ (dashed black line). All layers except for the first two converge to a form that is consistent to our analysis, i.e., low temperature (steep slope) for small entries and high temperature (flat slope) for large entries.

4.2. Improved Pareto Efficiency

Improving sparsity With the above defined metrics, we show that adding term (1) alone (denoted by MultiMax-l), i.e., modulating smaller entries, already leads to a better *Pareto Optimality* [22] regarding sparsity and multi-modality than SoftMax.

Proposition 4.2. *The following properties hold for $t_b > 1$.*

(i) *MultiMax-l generates sparser distribution than SoftMax with temperature 1.*

(ii) *MultiMax-l achieves better multi-modality than SoftMax with temperature 1.*

(See Appendix 2 for the proof.)

From the above analysis, we could see that MultiMax-l has higher *Pareto Efficiency* than SoftMax: MultiMax-l with $t_b > 1$ has both better sparsity and multi-modality than Softmax with temperature 1 (Proposition 3.5), and Softmax can not improve both properties at the same time by changing temperature (Proposition 4.2).

Enhancing multi-modality further As shown in Proposition 4.3, including the modulation of larger entries further enhances multi-modality while retaining better sparsity than SoftMax.

Proposition 4.3. *The following properties hold for $t_d < 1$ and $t_b > 1$:*

(i) *MultiMax can achieve better sparsity than SoftMax with temperature 1.*

(ii) *MultiMax can achieve better multi-modality than MultiMax-l.*

(See Appendix 2 for the proof.)

4.3. Generalization

Generalization to other activations

Piece-wise linear activation functions are widely adopted in modern machine learning algorithms, e.g., ReLU [4], Leaky ReLU [150] and PReLU [97]. Although MultiMax focuses on a different purpose, it can seen from Eq. (III.4) that the modulator/rectifier function σ of MultiMax is a generalization of these activation functions. For example, if $b = d = 0$, $t_d = 1$ and $t_b = 0$, then σ is reduced to ReLU. For the rest, it can be shown easily in a similar way.

Generalization to higher-order polynomials

So far, it has been shown that higher *Pareto Efficiency* can be realized with a piece-wise linear modulation function, which belongs to the family of first-order polynomials. To obtain smoother transitions at turning points and larger capacity, second-order terms are included in our final formulation of MultiMax:

$$\sigma(x) = x + \sum_{n=1}^{N} \underbrace{(1 - t_{b_n})Max(b_n - x, 0)^n}_{term(1)} + \underbrace{(t_{d_n} - 1)Max(x - d_n, 0)^n}_{term(2)}, \tag{III.6}$$

where n ranges from 1 to 2. We don't include higher orders beyond the second, because it proves to be sufficient in practice. We show in the ablation Section 5.3 that the extra nonlinearities brought by the second-order terms benefit the learning of the modulation scheme, in analogy to the previous study on activation functions [102, 45, 69].

As shown in Fig. III.1b, the output of SoftMax with varied temperatures forms a trajectory and converges to sparsemax as temperature approaches 0. EntMax-α stays close to the trajectory with $\alpha = 1.5$, and is indeed equivalent to softmax or SparseMax when $\alpha = 1$ or 2. MultiMax achieves, in the example, an otherwise non-reachable trade-off, with values close to the simplex that vary in two out of three possible modes. For a less complex illustration, we also provide the comparison with other reweighting functions with 2D inputs in Fig. III.2, in which case SoftMax is equivalent to Sigmoid. While other approaches handle small and large entries equally, MultiMax provides an input-adaptive reweigthing scheme.

We show in Fig. III.3 the learned modulator function of deit-small on ImageNet and compare it to the original input $\boldsymbol{x}$ (dashed black line) when used in attention layers. The learned functions at most layers (except the first two) conforms to our analysis: steeper slope for small entries (below the dashed black line on the left side means temperature smaller than 1) and flatter slope for large entries (below the dashed black line on the right side means temperature larger than 1). This conforms to our theoretical analysis that small entries should be suppressed with smaller temperature and large entries should be pushed closer with large temperature. Moreover, it is noteworthy that the need for sparsity increases as the layer goes deeper, according to the learned curves.

Generalization beyond Attention

As shown in the above analysis, the proposed MultiMax not only generalizes SoftMax, but also achieves a better Pareto optimality w.r.t. sparsity and multi-modality with appropriate parameterization. Due to its fully parameterized formulation, it is learnable and adaptable to any scenario where a reweighting function is required. Since the need for the degree of multi-modality and sparsity may vary among different applications, we do not explicitly constrain any of the parameters and optimize them jointly with the model.

Table III.2.: Comparing to Deit [221] baseline and anti-over-smoothing methods on ImageNet-1k by replacing SoftMax with MultiMax in the attention and/or output layers. * denotes that results are not strictly comparable: these methods rely on a different training setup. For example, additional training epochs are adopted by both works, talking-head [202] and a higher drop-path [110] rate are applied together with Patch Diversification.

Model	Method	Parameters	Epochs	Modulation		Acc. (%)
				Output	Attention	
Deit-tiny	SoftMax	5M	300	N/A	N/A	72.8
	MultiMax		300	✓	✓	**73.4**
Deit-small	Softmax	22M	300	N/A	N/A	80.4
	Top-k [230]		300	✓	N/A	80.6
	Ev-SoftMax [35]		300	-	✓	80.0
	MultiMax		300	✓	-	80.7
			300	-	✓	80.7
			300	✓	✓	**81.0**
Deit-base	SoftMax	86M	300	N/A	N/A	82.1
	MultiMax		300	✓	✓	**82.6**
Deit-small	Patch Diversification [86]		400	N/A	N/A	81.2*
	AttnScale [229]		500	✓	N/A	80.9*
	MultiMax		400	✓	✓	81.2
			500	✓	✓	**81.3**

4.4. Computational Efficiency

The extra computation of MultiMax is negligible for modern machine learning algorithms: As shown in Eq. (III.4), the total amount of additional parameters for a 12 layer Transformer with 2nd-order MultiMax is just $8 \times 12 = 96$, because each order only contains 4 parameters, including t_b, t_d, b and d. Moreover, the modulation function $\sigma(x)$ merely consists of cheap element-wise operations, i.e., multiplication with t_b and t_d, subtraction with b and d, two Max operations, addition of the two terms at each order as well as a residual addition. Thus a second-order MultiMax requires $7 \times 2 + 1 = 15$ extra Floating Point Operations (FLOPs) for a univariant input. For Deit-small model with input length of 256, hidden dimension of 384 and 12 layers, replacing MultiMax with SoftMax in all attention layers leads to 0.0168G extra FLOPs, i.e. only 0.37% of the original model's 4.6G FLOPs.

In practice, customized layers often run much slower than the highly optimized built-in Pytorch layers. The performance gap between theory and practice is mainly because the PyTorch framework is eagerly evaluated and thus brings additional memory access time and kernel launch time, please refer to this page [1] for more details. Thus a native Pytorch implementation of MultiMax increases the training time of Deit-small on ImageNet by about 40% (0.19 s/iteration vs 0.26 s/iteration), while the increase in inference time is negligible (less than 2%). However, we are able to achieve a reduction from 40% (native Pytorch implementation) to only about 10% increase of training time (0.21 s/iteration) by implementing the *Max* operator with 0 as built-in ReLU function and applying torch.jit.script decorator to fuse the remaining elementwise operations of our MultiMax,

[1] https://residentmario.github.io/pytorch-training-performance-guide/jit.html

following the documentation [2]. Notably, a fully optimized implementation of MultiMax in C++ or CUDA as done with Pytorch built-in layers might further reduce the gap.

5. Experiments

In this section, we replace SoftMax with MultiMax in different baselines and apply them to the corresponding tasks, including image classification on ImageNet1K, langauge modeling on Wiki-Text-103 corpus and machine translation on IWSLT-2014 corpus. Experimental results demonstrate consistent improvement with MultiMax, without any extra changes, e.g. hyperparameters or architecture. Moreover, we provide additional insights and demonstrate that advantagesous properties, including reduced over-smoothing (Section 5.2) and improved sparsity & multi-modality (Section 5.2), are achieved.

5.1. Benchmarking

ImageNet1K Classification

For classification, we train the widely adopted Deit [221] from scratch on ImageNet1K as baseline. Following the same training setup, we train Deit by only replacing the SoftMax function with our MultiMax, in the attention layers and/or output layer for a fair comparison. For training, we closely follow the training settings provided in [221] and train all the models for 300 epochs. Following the more recent works [44, 147], we also adopt Global Average Pooling (GAP) instead of using Class Token (CLT) as classification head. While class token causes discrepancy in attention [222] and breaks translation invariance [44], GAP avoids this problem and improves the accuracy.

The results in Table III.2 show a consistent improvement by using MultiMax for both attention and output activation layers. Although those sparse SoftMax variants work well for Machine Translation tasks, most of them have issues with Deit models. Ev-SoftMax decreases the performance when used in attention layers and the training does not converge (accuracy below 10%) when used in the output layer. For the inferior performance of Ev-SoftMax, we hypothesize that less sparsity is required for the attention among image patches than for language tokens, and zeroing out the entries smaller than average might be too aggressive. For the unstable training, their simple training-time modification might not be sufficient. The alternative losses provided by Sparse SoftMax and EntMax-1.5 require integer labels, thus are not compatible with the widely adopted label smoothing technique in vision transformers. Training instability issues are also encountered when using SparseMax in attention layers only. Therefore, we excluded them for the image classification task.

Language Modeling

We test the effectiveness of our MultiMax further on the Language Modeling task on WikiText-103 [156] using a 6-layer Transformer Decoder with 156M parameters. The implementation is based on the official fairseq repository[3] and the training setup is kept as default, i.e., 5e$-$4 learning rate with a maximum of 2048 tokens per GPU for 50k iterations on 4 GPUs. The results of the baseline transformer using SoftMax attention

[2] https://pytorch.org/tutorials/recipes/recipes/tuning_guide.html

[3] https://github.com/facebookresearch/fairseq

and our MultiMax are shown in Table III.3. We again observe a consistent improvement by applying MultiMax in the output activation for this task.

Table III.3.: Evaluation of the performance on WikiText-103 language modeling task by test perplexity.

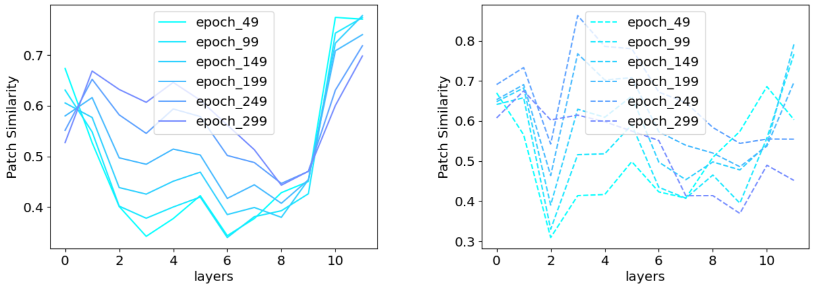
Method	Attention	Output	Perplexity ↓
SoftMax	-	-	29.4
Top-k [92]	✓	N/A	29.1
MultiMax	✓	-	29.0
	✓	✓	**28.7**

Table III.4.: Comparing to other SoftMax variants using two different baseline settings (see Section 5.1 for more details) on IWSLT 2014 English to German Translation task.

SoftMax	SparseMax	EntMax-1.5	EvSoftMax	MultiMax
34.4 ± 0.07	28.7 ± 0.16	34.6±0.09	34.7 ± 0.06	34.7 ± 0.07

Machine Translation

Following previous approaches, we also evaluate our method on the task of machine translation. We train a 38M 12-layer Transformer baseline with encoder-decoder (6 layers each) architecture [224] from scratch on the IWSLT2014 German to English dataset [27], following the training setup provided in the fairseq repository (Footnote 3). Under the same setting, we also train the transformer with our MultiMax in replacement of SoftMax in the attention layers, following the common setup in previous work. The single best checkpoint and a beam size of 5 is adopted. The detokenized SacreBLEU [177] scores (mean and standard deviation) of 3 runs are compared in Table III.4. MultiMax performs on par with EvSoftMax and is slightly better than EntMax-1.5 for this task.



(a) Softmax Deit-small

(b) MultiMax Deit-small

Figure III.4.: Patch similarities for each layer and at different epochs. Darker color denotes the patch similarities at a larger training epoch.

5.2. Empirical Studies and Insights

In this section, we empirically verify the positive impact of MultiMax on the over-smoothing issue, as well as the improvement on multi-modality and sparsity in the attention scores of Deit-small trained on ImageNet1K.

Analysis on Over-smoothing

To validate the efficacy of our MultiMax on preventing over-smoothing, we adopt the *Patch Similarity* [86] or *Mean Average Distance* (MAD) [33] metric to compare transformers using SoftMax and MultiMax on ImageNet1K. The numbers are shown in Fig. III.4. It can be observed that patch similarity increases as the depth grows for SoftMax attention during the entire training, whereas the patch similarity converges to a much lower level for MultiMax attention in deeper layers. We attribute this to the undesirable amount of attention assigned to irrelevant tokens which contributes the over-smoothing issue in Transformers. Moreover, it also showcases the flexibility of MultiMax's parameterized formulation, which can encourage exploration in the early stage and shift the distribution gradually towards higher sparsity as the training progresses. We have also examined the increased discrepancy between single layer attention and accumulated roll-out attention [1], which further indicates the strong connection between non-sparse SoftMax attention and the over-smoothing issue. Please refer to Appendix 4 for more details.

Analysis on Sparsity and Multi-modality

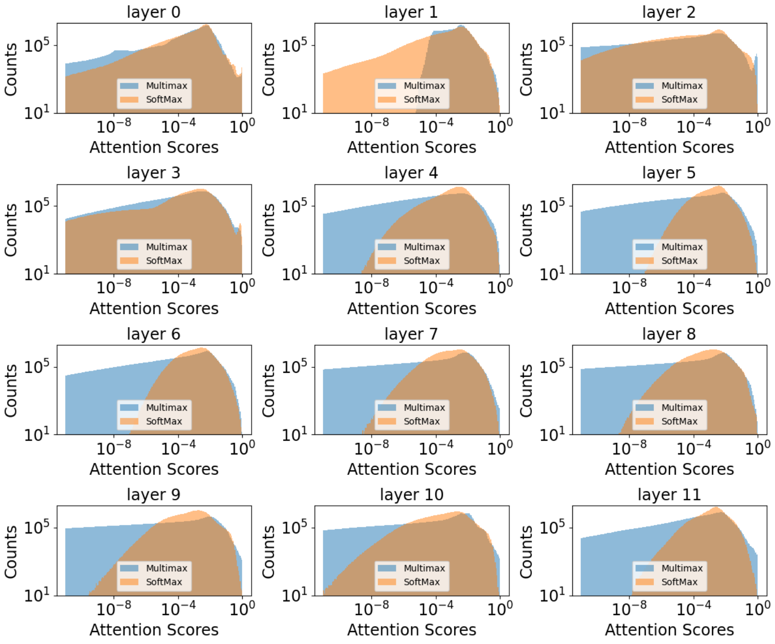


Figure III.5.: Histograms of the attention scores at each layer. MultiMax attention is distributed towards both ends: small scores are pushed closer to zero and more scores lie above 0.1.

In this section, we empirically evaluate the impact of using our MultiMax on the sparsity of attention scores. To achieve this, we evaluate the trained model on 1000 images and collect the attention scores at each layer.

As shown in Fig. III.5 in a log-log histogram, the attention scores of MultiMax are distributed more towards both ends of the score range, i.e., extremely small values near zero and large values between 0.1 and 1. In comparison, the attention scores of SoftMax are concentrated in the region in between, which corresponds to the bumps in the figure. Note that the number of counts are drawn at logarithmic scale, thus a small bump indeed indicates a large amount of counts. Notably, MultiMax attention behaves differently in the first two layers, which actually shows the flexibility of learning: the need for multi-modality or sparsity varies with varying context. Thus it can be a disadvantage to manually define the trade-off in advance. We also visualize the cumulative distribution of these attention scores in Appendix 3, which also indicates a stronger sparsity achieved by MultiMax.

5.3. Ablation

To study the effect of each design component of our MultiMax independently, we conduct experiments using Deit-small as the baseline on ImageNet1K for ablation, as shown in Table III.5. Since the language modeling and image classification tasks are computationally heavy, we report the result of a single run with the seed unchanged for all these experiments, as commonly done for ImageNet models.

Table III.5.: Impact of each MultiMax component.

Config	term (1)	term (2)	second order	Acc
1	-	-	-	80.4
2	✓	-	-	80.6
3	✓	✓		80.7
4	✓	✓	✓	81.0

To further validate the statistical significance of these results, we additionally conduct experiments using Deit-small with GAP on ImageNet1K and the results are recorded in Table III.6. Comparing to the relatively small standard deviation, the improvement of using MultiMax is reliable.

Table III.6.: Multiple runs with random seeds using Deit-small on ImageNet1k. MultiMax shows consistent improvement over SoftMax.

Method	Runs			Mean	Std
	1	2	3		
SoftMax	80.4	80.3	80.3	80.3	0.05
MultiMax	**81.0**	**80.8**	**80.7**	**80.8**	0.12

5.4. Attention Visualization

As Transformer models [224, 147, 278, 282, 226] stack a number of attention layers and aggregates the information repetitively, the attention scores at a single layer do not reflect the true information flow. To evaluate the impact on the classification more directly, we employ the well-established Grad-CAM [195] to qualitatively evaluate the impact on the model's decision making. We additionally provide single layer attention scores in Appendix 3 for reference.

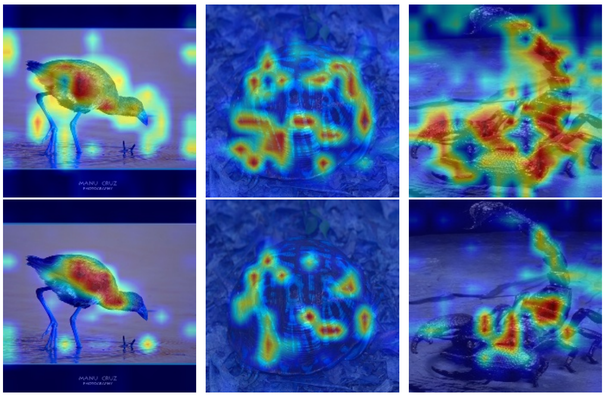


Figure III.6.: Grad-CAM of Deit-small using SoftMax (top row) and MultiMax (bottom row). The MultiMax attention maps are better localized on the objects and are close to zero in most background regions, indicating sparsity at the attention level.

6. Conclusion

In this chapter, we formalize, analyze, and evaluate the sparsity and multi-modality trade-off of SoftMax and proposed MultiMax as a remedy for tension between these two desirable objectives. Through both experimental evaluation and analysis, we validated that MultiMax successfully learns to achieve higher multi-modality and sparsity at the same time. Although we have already demonstrated the benefits of MultiMax in attention layers and output activation of a classifier and a generative model across a wide range of tasks, we believe it has an even broader range of applications, such as in value networks and policy gradient for reinforcement learning as well as the learning of categorical distributions with Gumbel Softmax [116].

Chapter IV.

Incorporating Geometric Prior into Attention for Image Modeling

Computer vision poses challenging tasks due to its rich hierarchical spatial structures and intricate data dependencies, making it an ideal domain to push the boundaries of neural probabilistic scoring within attention mechanisms. Vision Transformers (ViTs) leverage flexible self-attention to capture complex content relationships but often converge more slowly than CNNs, which benefit from fixed spatial inductive biases that enhance convergence and generalization, particularly in low-data regimes. However, these hard-coded priors in CNNs can restrict model capacity and flexibility when large-scale datasets are available.

Building on our prior work [282], this chapter introduces learnable Spatial Priors (SPs) via Spatial Prior–enhanced Self-Attention (SP-SA). This enables ViTs to automatically discover diverse spatial relations during training, combining inductive biases with the adaptability of transformers without restricting attention to local windows.

Our SP-ViT achieves state-of-the-art ImageNet performance without extra data, advancing this thesis on neural probabilistic scoring by integrating geometric priors into flexible attention mechanisms—improving convergence, generalization, and expressivity across data regimes.

1. Introduction

Transformers [224] have recently achieved exciting results in image classification [63, 147, 256, 93, 32, 221, 62, 222, 120, 48], after dominating in natural language processing (NLP) [58, 146, 20]. At the heart of transformer lies the so-called self-attention mechanism, which captures the content relations between all pairs of input tokens and focuses on related pairs selectively. Self-attention is more flexible in comparison to convolution, which is hard-coded to capture local dependencies exclusively. This can possibly equip transformer models with larger capacity and greater potential for computer vision tasks. As reported in recent works, transformers outperform Convolutional Neural Networks (CNNs), when pretrained on large dataset [63], facilitated with knowledge distillation [221] or pseudo labels [120] from pretrained CNNs.

Nevertheless, CNNs generalize better and converge faster than Vision Transformers (ViT). This suggests that certain types of inductive biases employed in convolution can still be beneficial to vision tasks. Not surprisingly, many recent studies [147, 62, 255, 49, 256, 239, 88, 221, 48] propose to incorporate convolutional inductive biases into ViTs in different ways. The effectiveness of convolution relies on the fact that neighboring

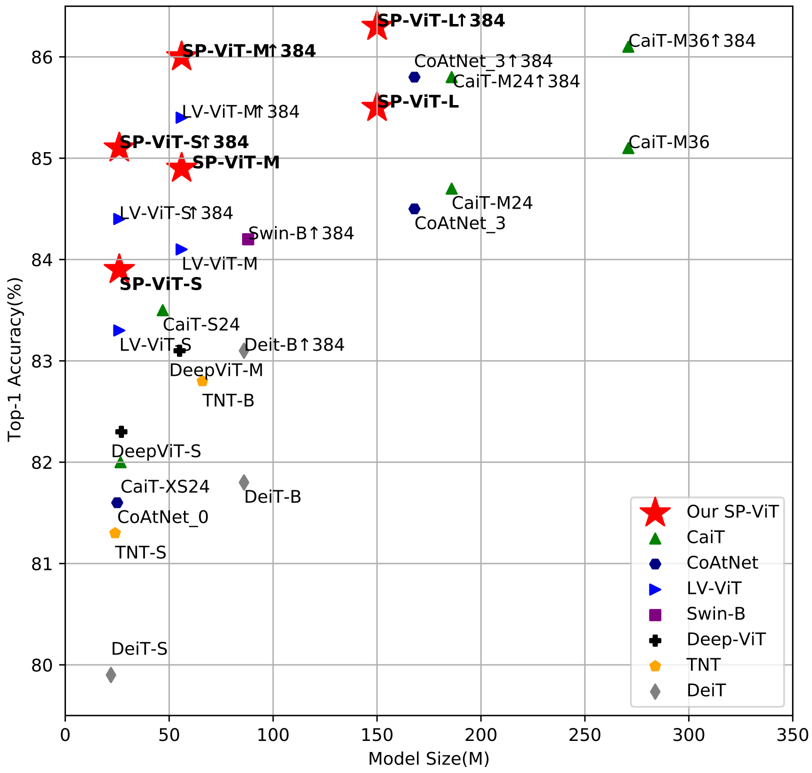
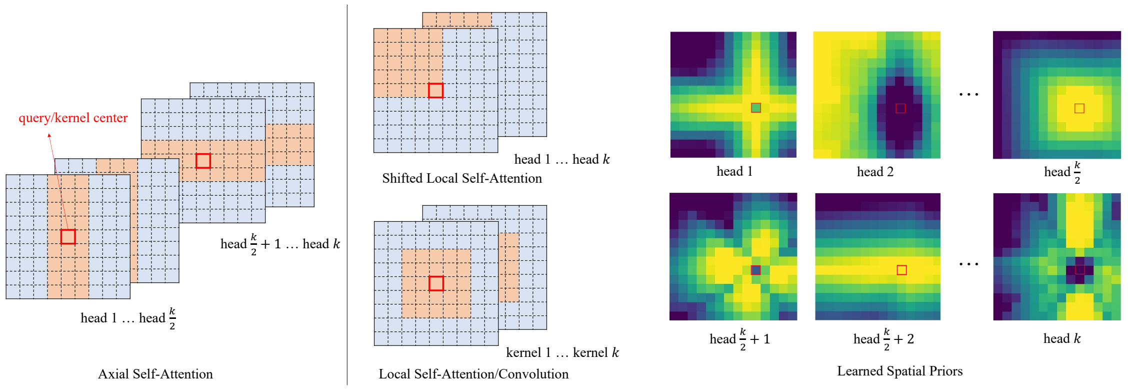


Figure IV.1.: ImageNet-1K top-1 accuracy of our proposed SP-ViT and state-of-the-art ViTs. The models shown are all trained on 224 × 224 resolution, ↑ denotes that models are fine-tuned on a higher resolution. Note that we exclude models pretrained on extra data or larger resolution than 224 × 224 for a fair comparison.

pixels of natural images are highly correlated, but there may exist other highly correlated contents outside the local receptive field of a convolutional filter. Therefore, we propose to make use of a variety of inductive biases simultaneously, just as humans do, e.g., if we see a part of a horizontal object, we naturally look along its direction instead of restricting our sight within a local area.

In this chapter, we introduce a novel family of inductive biases named *Spatial Priors* (SPs) into ViTs via an extension of vanilla self-attention (SA), called *Spatial Prior–enhanced Self-Attention* (SP-SA). SP-SA highlights a certain group of 2D spatial relations at each attention head based on the relative position of key and query patches. Since the construction and validation of appropriate spatial priors are extremely laborious, we introduce the idea of *learnable spatial priors*. More specifically, we only impose the weak prior knowledge to the model that different relative distances should be treated differently. Yet we do not force the model to favor any kind of spatial relation, e.g., neither local nor non-local. Effective spatial priors (SPs) are supposed to be discovered by the model itself in the training stage. For this purpose, SPs are represented by a family of mathematical functions which map the relative coordinates to abstracted scores, called *spatial relation functions*. To search for desirable spatial relation functions, we parameterize these functions by neural networks and optimize them jointly with ViTs.



(a) Hand-crafted convolutional inductive biases

(b) Proprosed SP-SA

Figure IV.2.: (a) Convolutional inductive biases proposed for ViTs: axial self-attention in CSWin-Transformer [62] and shifted local self-attention in Swin-Transformer [147]. (b) Our Spatial Priors (SPs) are learned by our model automatically. The learned SPs assign different scores for different spatial relations. Given a certain SP, attention is forced to be within high-score regions. Our SP-SA handles different types of spatial relations in a complementary manner, e.g., SPs which focus on local and non-local relations are both learned.

Thereby, the model can learn spatial priors similar to the ones induced in convolutions, as well as spatial relationships over larger distances. Examples for learned SPs are shown in Fig. IV.2(b). Diverse complementary patterns are presented in different attention heads, so that different types of spatial relations are handled individually.

As a matter of fact, convolutional inductive biases can be seen as a special kind of spatial priors: they first divide coordinate spatial relations into two categories, i.e., ones focusing on the local neighborhood and ones focusing on non-local regions. Then they learn priors of the local neighborhoods and ignore the non-local relations. For comparison, some of the existing approaches to combine such convolutional biases with ViTs are illustrated in Fig. IV.2(a).

In summary, we make the following contributions:

- We propose a family of inductive biases for ViTs that focus on different types of spatial relations, called Spatial Priors (SP). SPs generalize convolutional inductive biases to both local and non-local correlations. Parameterized with neural networks, SPs are automatically learned during training, w/o preference for any hard-coded region.

- We propose SP-SA, a novel self-attention variant that automatically learns beneficial spatial inductive biases. Built on SP-SA, we construct a ViT variant called SP-ViT. SP-ViTs establish state-of-the-art results on the ImageNet Benchmark w/o extra data.

- Our SPs are compatible with various input sizes, as they are derived from relative coordinates . SP-ViTs also demonstrate improved classification performance over the baseline model when fine-tuned on higher resolution.

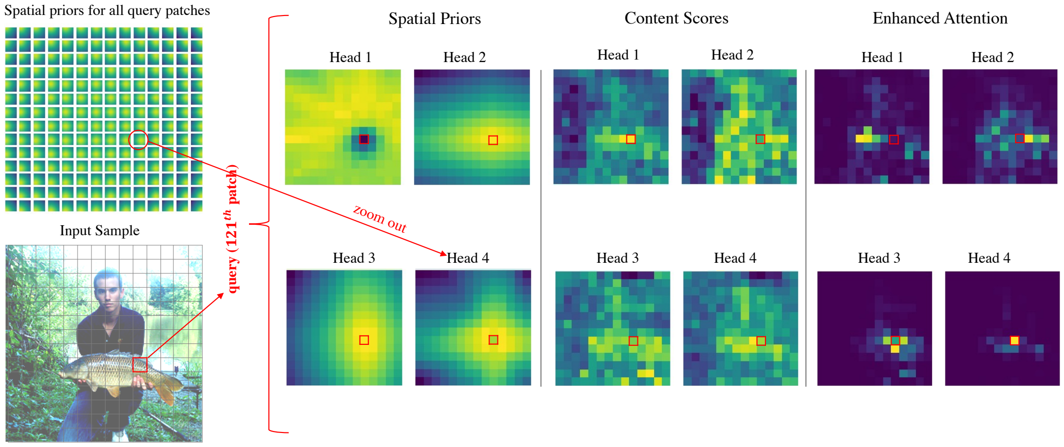


Figure IV.3.: Visualization of the learned 2D SPs, content scores and the enhanced attention. The input image is shown in the bottom-left and the query patch is marked in red. Different SPs are learned, including horizontal and vertical (head 2 and 3), non-local (head 1), as well as cross-shaped (head 4). The attention scores at each head are obtained within the context of a certain type of spatial relations. The original attention is distracted by background objects, whereas our Spatial Priors help the model to focus on the object of interest.

2. Related Work

Vision Transformers Recently, Dosovitskiy et al. [63] showed that purely attention-based transformers can achieve state-of-the-art performance in image classification, when pre-trained on large-scale datasets. Since then, a vast amount of efforts have been made to improve ViTs. Some works [120, 85] find it effective to add additional losses or regularization terms, while others propose new patch embedding blocks [93] or scale-up methods [222, 273]. [147, 62, 269, 231] propose to utilize multi-scale information, where local attention are is adopted to reduce the overall computation. It is noteworthy that an cross-shaped 2D structure, similar to the design in CSWin-Transformer [62], is also learned by our model.

Inductive Biases for ViTs ViTs' performance degrades rapidly with a reduced amount of training data. To alleviate this issue, many studies focus on emphasizing local correlations by introducing a convolutional inductive bias into ViTs, either by restricting SA to local windows [183, 147, 62], combining vanilla transformers with implicit or explicit convolutional operations [255, 49, 256, 239, 88, 44], knowledge distillation [221], or convolutional initialization [48]. Our work also incorporates inductive biases into ViTs, but they are not locally restricted and are automatically learned by the model. Indeed, as shown in Fig. IV.2(b), patterns that focus solely on local or remote regions are both present in the learned SPs.

Relative Spatial Information Transformers are by their very nature permutation equivalent, thus extra spatial information is often supplied to better handle ordered input data. Besides the common absolute positional embedding, the relative spatial information is also considered in Swin-Transformer [147] by an trainable bias term called relative positional bias. ConViT [48] also introduces a function based on coordinates relative to force the attention to be within a local region. In comparison to ViTs, using relative positional information is more common in NLP transformers. The relative positional embedding [201] is built on the distances between tokens and has been improved in XL-Net [251] and DEBERTA [99]. It can be extended to 2D for ViTs with little effort, and is proved to be effective in [240]. The essential difference of our method is the focus on various learned spatial relations at each head, which proves to be beneficial in Section 4.3.

3. Method

3.1. Spatial Prior-enhanced Self-Attention

Motivated by the observation that certain inductive biases on spatial relations can be beneficial to transformers, we propose an extension of self-attention enhanced by a combination of learned 2D Spatial Priors (SPs), called Spatial Prior–enhanced Self-Attention (SP-SA). Each SP $\Omega \in \mathbf{R}^{N\times N}$ forms a specific spatial context for computing attention scores $\mathbf{A} \in \mathbf{R}^{N\times N}$, and it is derived from coordinate spatial relations between input tokens, i.e. relative positions between the key and query patches for ViTs. Thus an SP has exactly the same form of attention scores and we can simply integrate it in the equation of vanilla SA [224] by multiplicative interaction:

$$A_{ij} = \frac{\exp(e_{ij} \cdot \Omega_{ij})}{\sum_{k=1}^{n} \exp(e_{ik} \cdot \Omega_{ik})}, \tag{IV.1}$$

with

$$e_{ij} = \frac{(\vec{x}_i^\top \mathbf{W}^Q)(\vec{x}_j^\top \mathbf{W}^K)^\top}{\sqrt{d_z}}, \tag{IV.2}$$

where $\vec{x}_i$ and $\vec{x}_j$ are the i^{th} and j^{th} input tokens.

Learnable 2D Spatial Priors

Taking query patch i as the reference point, we can obtain a relative coordinate $\vec{r}_{ij} \in \mathbf{R}^2$ for image patch j. Then we employ a shared mapping f_p for all query and key patch pairs, named spatial relation function:

$$\Omega_{ij} = f_p(\vec{r}_{ij}), \tag{IV.3}$$

the outputs together form the so-called 2D SP Matrix Ω.

To enable the model to learn desirable inductive biases automatically, we employ Multilayer Perceptron (MLP) to parameterize the mapping from 2D relative coordinates to Ω. Thereby, we allow Ω to learn a weighting for the attention scores for query $\vec{x}_i$ and key $\vec{x}_j$ which depends solely on their relative coordinates and is applied in a non-linear way, i.e. before the softmax. We extend SP-SA to its multi-head version by adding a unique network to each head. This design follows the same motivation as multi-head self-attention and assumes that a combination of different SPs should boost the performance.

3.2. Relation to Other Methods

In the following, we discuss the relation of SP-SA to the most related work.

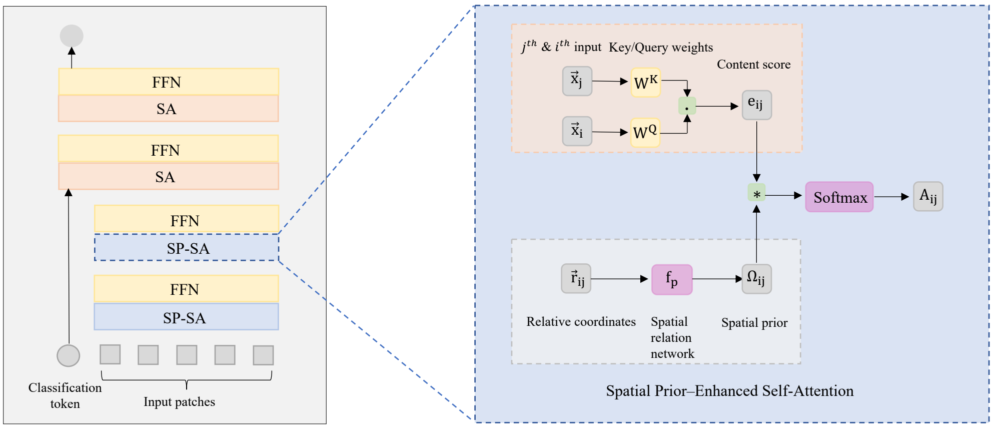


Figure IV.4.: The schema of SP-ViT. SP-SA can be used as a drop-in replacement for the vanilla SA layer at a range of depths. Because the classification token does not have a valid 2D relative coordinate, it is simply concatenated with the hidden representation after the last SP-SA layer. FFN: feedforward network (2 linear layers separated by a GeLU activation).

Relation to Local Windows The square and cross-shaped windows used in [147, 62] can be seen as a special form of our proposed spatial relation functions in practice:

$$f_p(\vec{r}_{ij}) = \begin{cases} 1, & \text{if } \|(\vec{r}_{ij} - \vec{\Delta}) \odot (a, b)\|^{\infty} <= 1 \\ & \text{or } \|(\vec{r}_{ij} - \vec{\Delta}) \odot (b, a)\|^{\infty} <= 1, \\ 0, & \text{else} \end{cases} \tag{IV.4}$$

where $\vec{\Delta}$, a and b control the shift, window width and height respectively. If $a = b$, it generates a square window, otherwise it results in a cross-shaped window. Both works only adopt some hard-coded patterns for the whole network, while our method proposes to benefit from a variety of beneficial 2D structures.

Relation to PSA The Positional Self-Attention (PSA) proposed in [48] can also be regarded as a manually designed family of spatial relation functions:

$$f_p(\vec{r}_{ij}) = \alpha(\|(\Delta_x, \Delta_y)\|^2 - \|\vec{r}_{ij} - (\Delta_x, \Delta_y)\|^2), \tag{IV.5}$$

where the parameters Δ_x and Δ_y are specially initialized to approximate convolution effect.

Note that their main contribution is the so-called local/convolutional initialization, which restricts the number of heads to the square of integer numbers, and the initial values of both α and $\vec{\Delta}$ require extra hyperparameter tuning. In order to compare with their method, we adopt a ViT baseline with 9 heads for ablation analysis as in [48].

Relation to Relative Positional Embeddings Shaw et al. [201] introduce the so-called 1D Relative Positional Embedding (RPE) for transformers to take relative distances into account:

$$A_{ij} = \frac{\exp(e_{ij} + (\vec{x}_i \mathbf{W}^Q) T_{r_{ij}})}{\sum_{k=1}^{n} \exp(e_{ik} + (\vec{x}_i \mathbf{W}^Q) T_{r_{ij}})}, \tag{IV.6}$$

where T is a learnable embedding table from which the RPE is taken. Then it interacts multiplicatively with the query. If extended to 2D, it is equivalent to applying a linear transformation to one-hot representations of relative distances. For one-hot representations, the magnitude of distances is neglected, while this is not the case for relative coordinates.

The main difference of our approach to all previous methods is the combination of complementary spatial priors at each layer. As shown in Table 6, performance drops form 83.6% to 82.1% with the same spatial priors per layer. In addition, we can see in Figure 2(b) that different spatial foci (local and non-local) are learned for each layer. With such spatial foci, our model is less distracted by noises w.r.t. a certain context. For example, as discussed in Section 4.1, our SP-ViT shows a significantly diminished class activation in background regions compared to DeiT. We also confirmed experimentally that our SP-SA outperforms these methods, see Table IV.3.

4. Experiments

We first provide an experimental evaluation of the proposed SP-SA in the context of image classification on the ImageNet-1k dataset and show that SP-ViTs achieve state-of-the-art results for training without extra data. Further, we provide an extensive ablation study to analyze the impact of all proposed model details.

Table IV.1.: Comparing to state-of-the-art models trained on ImageNet-1k 224 × 224 resolution. Models are by default trained and tested on 224 × 224 resolution if not specified. ↑ plus size denotes the model is trained on 224 × 224 resolution then fine-tuned and tested on size × size resolution. The performance of LV-ViT-L trained on 224 × 224 resolution is not available in [120]. And LV-ViT-L trained on 288 × 288 resolution has a lower accuracy of 85.3%.

Network	Top-1 (%)	Parameters	FLOPs
DeiT-S [221]	79.9	22M	4.6B
CaiT-XS-24 [222]	82.0	27M	5.4B
LV-ViT-S [120]	83.3	26M	6.6B
Our SP-ViT-S	**83.9**	26M	6.6B
DeiT-B [221]	81.8	86M	17.5B
Swin-B [147]	83.3	88M	15.4B
CaiT-S-24 [222]	83.5	47M	9.4B
LV-ViT-M [120]	84.1	56M	12.7B
Our SP-ViT-M	**84.9**	56M	12.7B
CaiT-M-24 [222]	84.7	186M	36.0B
Our SP-ViT-L	**85.5**	150M	34.7B
LV-ViT-S↑384 [120]	84.4	26M	22.2B
SP-ViT-S↑384	**85.1**	26M	22.2B
CaiT-S-24↑384 [222]	85.1	47M	32.2B
LV-ViT-M↑384 [120]	85.4	56M	42.2B
Our SP-ViT-M↑384	**86.0**	56M	42.2B
CaiT-M-24↑384 [222]	85.8	186M	116.1B
Our SP-ViT-L↑384	**86.3**	150M	110.6B

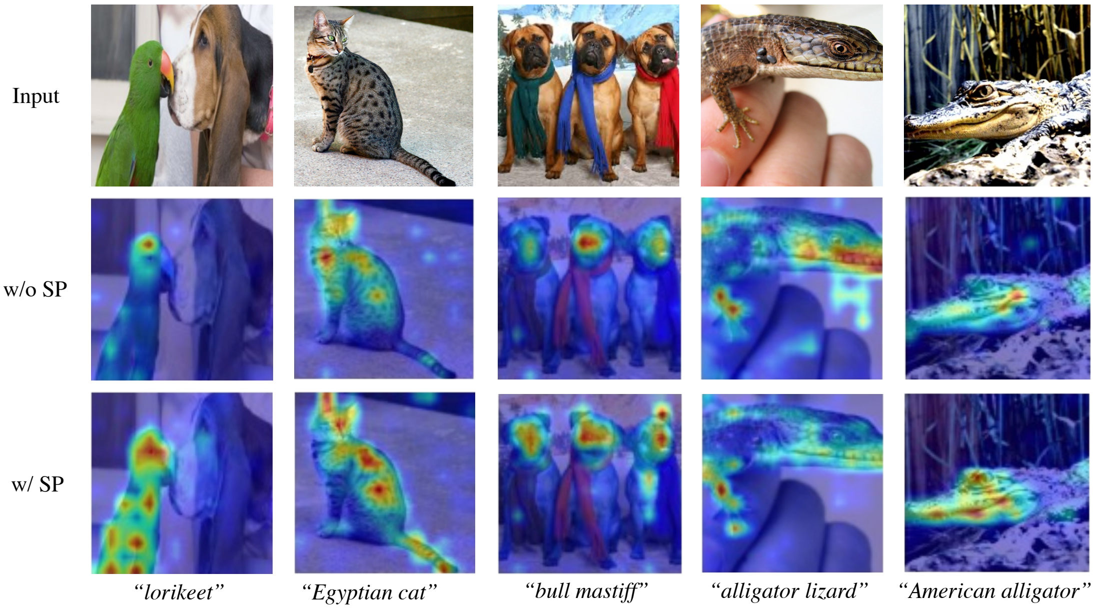


Figure IV.5.: Visualization using Transformer Explainability [31]. The second row are results of DeiT baseline w/o SP layers. The Last row are results of SP-ViT. Our SP-ViT generate results with more focus on areas of interests and less distraction from background.

4.1. Image Classification on ImageNet-1K

Settings All models for ImageNet-1K classification are trained on a single machine node with 8 Tesla V100 GPUs. Our code is based on DeiT [221]. To obtain our SP-ViT, we replace the vanilla SA layers of the baseline with SP-SA till the last 2 layers and follow the training settings in [120] (with Token Labeling). We keep the vanilla SA in the last 2 layers, based on the ablation analysis conducted on a fraction of ImageNet, please refer to the Appendix for more details. When fine-tuning on higher resolution (indicated by ↑384 in Table IV.1), we set batch size to 512, learning rate to 5e-6, weight decay to 1e-8 and we fine-tune the model for 30 epochs.

Comparing to State-of-the-Art Models We compare our SP-ViT (based on LV-ViT) with other recent ViTs in Table IV.1. Within all groups of comparable model sizes, SP-ViT outperforms competing models. Our best result of 86.3% is achieved with SP-ViT-L↑384. It outperforms all previous models with about only 150M parameters as compared to 271M parameters of the second best CaiT-M-36↑384. Also note that our smaller SP-ViT-M↑384 already achieves 86.0% accuracy, on par with CaiT-M-36↑384 while reducing parameters from 271M to 56M (by a factor of about 4.8).

Qualitative results We present visualizations of target class activation maps using the recent technique [31] in Figure IV.5 to showcase the behavior of SP-ViT. While the DeiT model only shows class activations on small parts of the target class regions, for example on the head of the "Lorikeet", the fur of the "Egyptian cat" or the jaw of the "American alligator", the proposed SP-ViT model shows class activations on wider target class regions. Thereby, it follows well class specific image regions such as the pointy ears as well as the tail of the "Egyptian cat", and the dogs' ears in the "Bull mastiff" class. The "Alligator lizard" example as well as the "American alligator" further show a significantly diminished class activation in background regions compared to DeiT. In summary, we make two observations: 1) The results generated by SP-ViT focus more on areas of target class objects comparing to DeiT. In “Lorikeet", “Bull mastiff", “Egyptian cat" and “American alligator", SP-ViT’s activation maps clearly have a better coverage of target class; 2) The distraction by background is better suppressed, e.g. in “Alligator lizard", resulting in a cleaner activation map.

4.2. Semantic Segmentation

Following [120] and [147], we utilize UperNet as our base framework and our SP-Vit trained ImageNet1K as the backbone to perform semantic segmentation on ADE20K. We adopt the same training setup as [13] and [15] and obtain 49.8 mIoU, which improves the result of LV-ViT-S by 1.2 mIoU. This shows that our proposed SPs benefit downstream tasks as well.

Table IV.2.: Performance of our proposed SP-ViT in the downstream semantic segmentation task. SP-ViT improves over its baseline on both single-scale (SS) and multi-scale (MS) setups on the validation set.

Method	mIoU (SS)	P.Acc. (SS)	mIoU (MS)	P.Acc. (MS)
LV-ViT-S	47.9	82.6	48.6	83.1
SP-ViT-S	49.0	83.0	49.8	83.4

4.3. Ablation Analysis

For ablation, we employ a small DeiT model as the baseline with 12 layers, 9 heads and 432 embedded dimensions. The choice of head numbers is simply for a fair comparison with other methods, because Positional Self-Attention (PSA) introduced by d’Ascoli et al. [48] requires such specific numbers (square of integer numbers) of heads. Due to limited available computation resources, we train all model variants on the first 100 classes of ImageNet-1K called ImageNet-100 for 300 epochs, following the setup in [48]. In this section, we simply take the accuracy at the last epoch for all models. This should be a fair comparison, since we adopt the same hyperparameters for different models without tuning. For all experiments in this section, we train the models on 4 NVIDIA P100 GPUs and adopt a batch size of 256. The rest of settings are kept the same as DeiT’s w/o knowledge distillation in [221].

Comparing to Related Approaches SP-SA Additive is obtained by replacing the multiplication in Eq. (IV.1) with a summation. It is more comparable to other methods which also employ additive interaction between spatial information and content scores. As shown in Table IV.3, our SP-SA has much higher Top-1 accuracy than all previous methods. The advantage of our SP-SA can be largely credited to the combination of different spatial foci at each head, see Section 4.3. As opposed to our method, the Relative Positional Bias [147] directly adds a univariate bias term to the content score before applying softmax, and the bias term is taken from a parameter table based on the relative coordinates. Adding such a bias term is a straightforward idea to include relative spatial information,

Table IV.3.: Comparing to SA with Relative Positional Bias [147], Positional SA[48], SA with the 2D extension of Relative Positional Embedding (RPE) [201] as well as a more advanced version proposed in DEBERTA [99] on ImageNet-100.

Method	Top-1 acc (%)
2D RPE [201]	79.9
Improved 2D RPE [99]	82.8
Relative Positional Bias [147]	81.3
Positional Self-Attention [48]	82.5
SP-SA Additive	83.5
SP-SA	**83.6**

but it is neither based on the idea of nor capable of learning complex 2D spatial priors, as reflected in Table IV.3.

We have also compared SP-SA to Positional Self-Attention [48] with hand-crafted spatial relation function. Our method delivers better performance, which shows that the effort in such a manual design process can be saved by our learnable SP.

Table IV.4.: The effect of unique Spatial Priors (SPs) per head. This setting performs best.

SP-SA	Top-1 (%)
shared SP	82.6
unique SPs per layer	82.1
unique SPs per layer&head (default)	**83.6**

Single vs Multiple Spatial Priors To validate the benefit of combining various learned SPs, we compare SP-SA to two variants: one only adopting a single SP for each layer, the other learning the same SP for the whole network. As shown in Table IV.4, a shared SP for the whole network provides better results than a single SP for each layer. However, the proposed setting with a unique SP per layer&head performs best, providing evidence of the benefit of combining different SPs.

5. Conclusion

In this chapter, we introduce a variant of self-attention (SA) named Spatial Prior-enhanced Self-Attention (SP-SA) to facilitate vision transformers with automatically learned spatial priors. Based on the SP-SA, we further proposed SP-ViT and experimentally demonstrate the effectiveness of our method. Our proposed SP-ViTs establish state-of-the-art results for models trained on ImageNet-1K only. For example, SP-ViT-M achieves a 0.8% higher accuracy comparing to the previous state-of-the-art LV-ViT-M. We hope that our powerful SP-SA can stimulate more studies on designing appropriate inductive biases for ViTs.

Chapter V.

Incorporating Structural Prior into Attention for Skeleton-Based Action Recognition

Skeleton-based action recognition offers a compact, robust representation of human motion that is less sensitive to environmental changes, making it ideal for real-world applications. Its graph-structured data naturally aligns with neural probabilistic modeling, enabling the integration of structural priors into attention mechanisms. While GCNs use fixed skeletal connectivity, their limited flexibility motivates more adaptive models. This chapter advances the thesis by enhancing neural probabilistic scoring with attention models that flexibly encode structural priors, improving performance and generalization.

Building on our prior work [278], this chapter introduces Hypergraph Self-Attention (HyperSA), which incorporates structural priors into transformers by using a novel relative positional encoding based on graph distances, as well as modeling higher-order joint groupings through hypergraphs. This approach enables the model to learn complex joint co-occurrences beyond pairwise relations, enhancing representational power without sacrificing efficiency.

Our Hyperformer achieves state-of-the-art accuracy and efficiency on skeleton-based action benchmarks, contributing to the thesis on neural probabilistic scoring by integrating structural priors into flexible attention models—advancing the balance between inductive bias and model capacity for structured data domains.

1. Introduction

Skeleton-based human action recognition has attracted increasing attention due to its computational efficiency and robustness to environmental variations and camera viewpoints. One of the key advantages of skeleton-based action recognition is that body keypoints can be easily acquired using sensors [268] or reliable pose estimation algorithms [23]. This offers a more reliable alternative to RGB or depth-based methods, making it a promising solution for various real-world applications.

Graph Convolution Networks (GCNs) have been widely used for modeling off-grid data. To our knowledge, Yan et al. [249] were the first to treat joints and their natural connections as nodes and edges of a graph, and employ a GCN [125] on such a predefined graph to learn joint interactions. Since then, GCNs have become the de facto standard of choice for skeleton-based action recognition. To further capture the interactions between physically unconnected joints, state-of-the-art GCNs [42, 38, 252, 208, 210, 211] adopt a learnable topology which merely uses the physical connections for initialization. Even so, they still need to rely on attention mechanisms to relax the restriction of the fixed topology, which is the key to their improved performances.

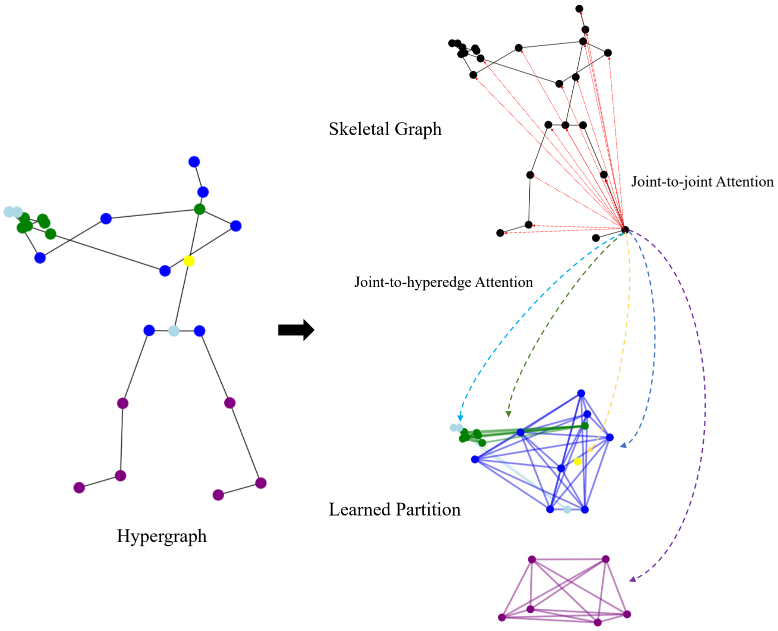


Figure V.1.: Illustration of our proposed HyperSA using a frame from the action class “Clapping Hands". HyperSA accommodates the additional high-order relations besides the skeletal interconnections.

Given these facts, it is natural to question whether a purely attention-based Transformer model would be a better candidate for skeleton-based action recognition. However, current research [176, 207] has shown that the performance of such models is far from satisfactory. This can be attributed to the fact that the formulation of the vanilla Transformer ignores the unique characteristics of skeleton data, i.e., the permutation equivalent attention operation is agnostic to the bone connectivity between human body joints. To address this issue, absolute positional embeddings have been used [224, 63, 221] , but they still lack the necessary structural information. In contrast, relative positional embeddings have been shown to be more effective for Transformers in various tasks, involving language [201, 50, 98], vision [147, 282, 240], and graph data [254, 277]. To incorporate the information of the bone connectivity, we also introduce a powerful relative positional embedding based on graph distance. Our embedding retains the information of skeletal structure during the entire training process, whereas GCNs merely use it for initialization.

Moreover, we reveal an underlying issue of graph models for this task in general. For human actions, each type of body joint has a unique physical functionality. As a result, certain re-occurring groups of body joints are often involved in specific actions, such as the subconscious hand movement for maintaining balance. Vanilla attention is incapable of capturing these underlying relationships that are independent from joint coordinates and go beyond pair-wise interactions. To compensate for this, we employ the concept of hypergraph [274, 71, 13] to accommodate the higher-order relations of body joints. With the hypergraph representation, we propose a novel variant of Self-Attention (SA) called **Hypergraph Self-Attention (HyperSA)**, which considers both pair-wise and high-order relations. Given a partition of the human body joints into different groups, a representation of each group is derived based on its assigned joints. The group representation is then linearly transformed and multiplied with joint queries, allowing joint-to-group interactions in addition to the vanilla joint-to-joint SA. Though HyperSA works well with empirical partitions, we additionally propose an approach to search the optimal partition strategy automatically, further improving its performance.

At the same time, Transformers spend a large portion of capacity on intra-token modeling via feed-forward layers. While this is important for complex tokens such as image patches or word embeddings, we analyze that such an expensive step is unnecessary for joint coordinates which are merely three-dimensional. This implies that the modeling of inter-token relations, or the so-called joint co-occurrences, is the key to successful action recognition. We thus suggest removing MLP layers for computation and memory reduction, and show in Section 5.3 that MLP layers are indeed negligible. This leads to a lightweight Transformer which is comparable to GCNs in model size and computation cost.

Our main contributions can be summarized as follows:

- We propose to incorporate the structural information of human skeleton into Transformer via a relative positional embedding based on graph distance, leveraging the gap between Transformer and state-of-the-art hybrid models.
- We devise a novel extension of Self-Attention (SA) called Hypergraph Self-Attention (HyperSA). To our best knowledge, HyperSA is the **first attention varaint on hypergraph** for skeleton-based action recognition.
- The resulting model, termed as Hyperformer, is the **first Transformer** which beats state-of-the-art models w.r.t. both efficiency and accuracy.

2. Related work

In this section, we highlight the most related work to ours regarding the spectacle of method and application.

2.1. Representation of skeleton data

Graph Representation Graph is the most prevalent choice for representing non-euclidean data and human skeleton can be naturally represented as graph. Comparing to other graph models [247, 84, 225], the Graph Convolutional Network (GCN) proposed by Kipf [125] is widely adopted for action recognition due to its simplicity and thus higher resistance to overfitting. Transformers have also achieved great performance in a variety of graph learning tasks [258, 254], although they often requires much higher computation budget.

Hypergraph Representation In real-world scenarios, relationships could go beyond pairwise associations. Hypergraph further considers higher-order correlations among data. Although hypergraphs can be modeled as a graph approximately via techniques such as clique expansion[274], such approximations fail to capture higher-order relationships in the data and result in unreliable performance [43, 140]. This motivates the study of learning on hypergraphs [101, 264, 71, 248]. Attention-based hypergraph models have also been proposed for multi-modal learning [123] and inductive text classification [61]. Our HyperSA is the first hypergraph attention designed for skeleton-based action recognition.

2.2. Skeleton-based action recognition

In early years, RNNs [64, 266] have been a popular choice to tackle the problem of skeleton-based human action recognition. The application of CNNs for this task [122, 145] is also well-studied. Nevertheless, the spatial interactions between joints are ignored in the above methods, and GCNs have become a more common choice in this field, by modelling the spatial configurations as graphs.

GCN-based approaches Yan [249] first introduced GCN [125] to model the joint correlations and demonstrated its effectiveness for action recognition. However, the limitation of assuming a fixed topology according to the natural connections is identified later, and most follow-up works adopt a learnable topology for action recognition. Many among them [40, 208, 38, 252] also employ attention or similar mechanisms to produce a data-dependent component of the topology (analogous to Graph Attention Networks [225]), boosting GCN's performance further.

Transformer-based approaches Attempts to tackle this problem with Transformers have been made recently. They mainly focus on handling the challenge brought by the extra temporal dimension. [176] propose a two-stream model consisting of spatial and temporal Self-Attention for modeling intra- and inter-frame correlations, respectively. Instead, [207] employ a Transformer which models the spatial and temporal dimension in an alternate fashion. Nevertheless, none of them achieved comparable results to state-of-the-art GCN-based approaches. Our work is the first to reveal the reason behind, i.e., vanilla Transformers fails to exploit the special characteristics of skeleton data, including high-order joint relations and skeletal connectivity. Notably, later work [228] proposes another extension of attention to hypergraph, which is shown to be inferior to our HyperSA.

3. Preliminaries

In this section, we recap the definition of Self-Attention and hypergraphs.

3.1. Self-Attention

Given an input sequence in the form of $X = (\vec{x}_1, ..., \vec{x}_n)$, each token $\vec{x}_i$ is first projected into *Key* $\vec{k}_i$, *Query* $\vec{q}_i$ and *Value* $\vec{v}_i$ triplets. Then the so-called attention score A_{ij} between two tokens is obtained by applying a softmax function to the dot product of $\vec{q}_i$ and $\vec{k}_j$ [224]:

$$A_{ij} = \vec{q}_i \cdot \vec{k}_j^\top , \tag{V.1}$$

the final output at each position is computed as the weighted sum of all Values:

$$\vec{y}_i = \sum_{j=1}^{n} A_{ij} \vec{v}_j . \tag{V.2}$$

An extension called Multi-Head Self-Attention (MHSA) is often adopted by Transformers in practice. It divides the channel dimension into subgroups and apply Self-Attention to each subgroup in parallel to learn different kinds of inter-dependencies. For simplicity, we omit the notation of MHSA in this chapter.

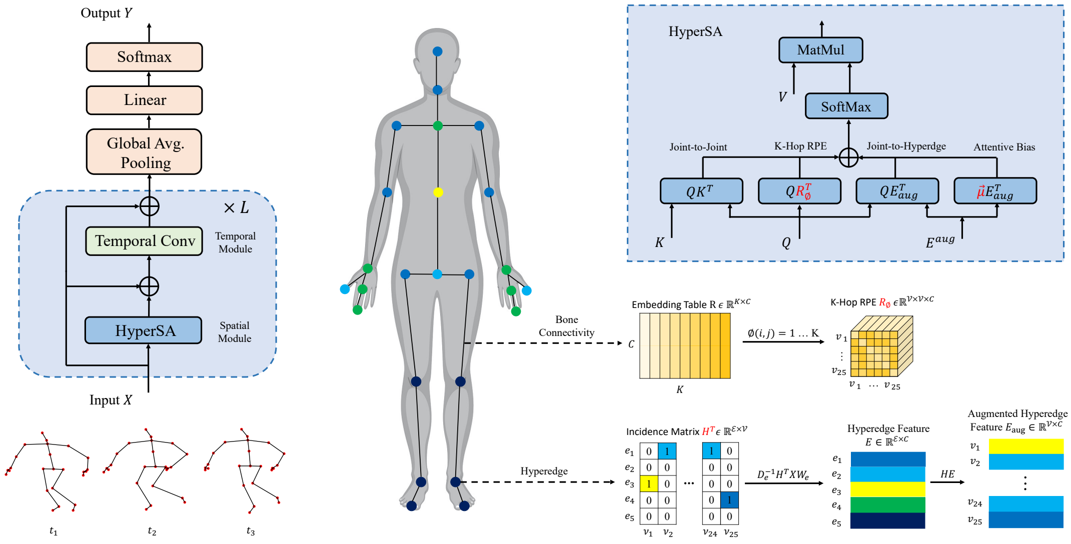
3.2. Hypergraph representation

Unlike standard graph edges, a hyperedge in a hypergraph connects two *or more* vertices. An unweighted hypergraph is defined as $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, which consists of a vertex set $\mathcal{V}$ and a hyperedge set $\mathcal{E}$. The hypergraph $\mathcal{H}$ can be denoted by a $|\mathcal{V}| \times |\mathcal{E}|$ incidence matrix H, with entries defined as follows:

$$h_{v,e} = \begin{cases} 1, & if \quad v \in e \\ 0, & if \quad v \notin e \end{cases} \tag{V.3}$$

The degree of a node $v \in \mathcal{V}$ is defined as $d(v) = \sum_{e \in \mathcal{E}} h_{v,e}$, and the degree of a hyperedge $e \in \mathcal{E}$ is defined as $d(e) = \sum_{v \in \mathcal{V}} h_{v,e}$. The degree matrices D_e and D_v are constructed by setting all the edge degrees and all the vertex degrees as their diagonal entries, respectively.

In this chapter, we consider the special case of $d(v) = 1$ for all vertices, i.e. , body joints are divided into $|\mathcal{E}|$ disjoint subsets, which is efficient in practice. Notably, the incidence matrix H is equivalent to a partition matrix in this case. Each row is a one hot vector denoting the group to which each joint belongs.



Output Y
Softmax
Linear
Global Avg. Pooling
$\times L$
Temporal Conv
Temporal Module
HyperSA
Spatial Module
Input X
t_1
t_2
t_3
Bone Connectivity
Hyperedge
HyperSA
MatMul
V
SoftMax
Joint-to-Joint
K-Hop RPE
Joint-to-Hyperdge
Attentive Bias
QK^T
$QR_\emptyset^T$
QE_{aug}^T
$\vec{\mu}E_{aug}^T$
K
Q
E^{aug}
Embedding Table R $\epsilon\ \mathbb{R}^{K \times C}$
C
K
$\emptyset(i,j) = 1 \dots K$
K-Hop RPE $R_\emptyset\ \epsilon \mathbb{R}^{\mathcal{V} \times \mathcal{V} \times C}$
v_1
v_{25}
$v_1 \dots v_{25}$
Incidence Matrix $H^T \epsilon\ \mathbb{R}^{\mathcal{E} \times \mathcal{V}}$
e_1 e_2 e_3 e_4 e_5
v_1 v_2 v_{24} v_{25}
$D_e^{-1}H^TXW_e$
Hyperedge Feature $E \in \mathbb{R}^{\mathcal{E} \times C}$
HE
Augmented Hyperedge Feature $E_{\text{aug}} \in \mathbb{R}^{\mathcal{V} \times C}$
v_1 v_2 v_{24} v_{25}

	v_1	v_2	…	v_{24}	v_{25}
e_1	0	1		1	0
e_2	0	0		0	0
e_3	1	0		0	0
e_4	0	0		0	1
e_5	0	0		0	0

Figure V.2.: Model architecture overview and illustration of our proposed HyperSA layer.

4. Method

As analyzed in Section 1, multiple specific joints often move cooperatively in an action, i.e., there are inherent higher-order relations beyond the pair-wise relation. Therefore, we propose to introduce the prior information of the intrinsic hyper connections into vanilla Self-Attention. Specifically, a novel **Hypergraph Self-Attention (HyperSA)** layer is introduced, which makes Transformers aware of extra higher-order relations shared by a subset of joints connected to each hyperedge.

4.1. Deriving the hyperedge feature

Given an incidence matrix H, we propose an effective approach to obtain the feature representation for each subset of joints connected to a hyperedge. Let C denote the number of feature dimensions, individual joint features $X \in \mathbb{R}^{|\mathcal{V}| \times C}$ are first aggregated into subset representations $E \in \mathbb{R}^{|\mathcal{E}| \times C}$ by the following rule:

$$E = D_e^{-1} H^\top X W_e, \tag{V.4}$$

where:

- The product of incidence matrix H and input X essentially sums up the belonging joint features of each subset.
- The inverse degree matrix of hyperedges are multiplied for the purpose of normalization.
- The projection matrix $W_e \in \mathbb{R}^{C \times C}$ further transforms the features of each hyperedge to obtain their final representations.

Then we construct an augmented hyperedge representation $E_{aug} \in \mathbb{R}^{|\mathcal{V}| \times C}$ by assigning hyperedge representations to the position of each associated joint:

$$E_{aug} = H D_e^{-1} H^\top X W_e. \tag{V.5}$$

4.2. Encoding human skeleton structure

Human body joints are naturally connected with bones and form, together with the latter, a bio-mechanical model. In such a mechanical system, the movement of each joint in an action is strongly influenced by their connectivities. Therefore, it is beneficial to take the structural information of human skeleton into account.

Analogous to the established Relative Positional Embedding (RPE) for image [240] and language [98, 201] Transformer, we propose a powerful k-Hop Relative Positional Embedding $R_{ij} \in \mathbb{R}^C$, which is indexed from a learnable parameter table by the Shortest Path Distance (SPD) between the i^{th} and j^{th} joints. In comparison to the learnable scalar spatial encoding in [254], it has larger capacity and interacts with the query additionally.

4.3. Hypergraph Self-Attention

With the obtained hyperedge representation and skeleton topology encoding, we now define our Hypergraph Self-Attention as follows:

$$A_{ij} = \underbrace{\vec{q_i} \cdot \vec{k_j}^\top}_{\text{(a)}} + \underbrace{\vec{q_i} \cdot E_{aug,j}^\top}_{\text{(b)}} + \underbrace{\vec{q_i} \cdot R_{\phi(i,j)}^\top}_{(c)} + \underbrace{\vec{u} \cdot E_{aug,j}^\top}_{(d)}, \tag{V.6}$$

where $\vec{u} \in \mathbb{R}^C$ is a learnable static key regardless of the query position.

- Term (a) alone is the vanilla SA, which represents joint-to-joint attention.
- Term (b) computes the joint-to-hyperedge attention between the i^{th} query and the corresponding hyperedge of the j^{th} key.

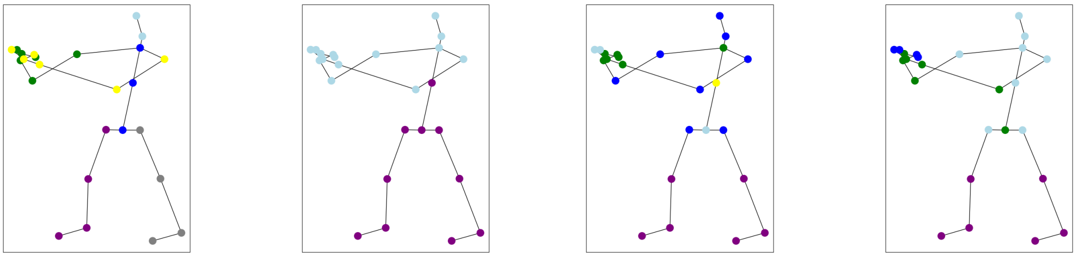
- Term (c) is the term for injecting the structural information of human skeleton with k-Hop Relative Positional Embedding.
- Term (d) is intended for calculating the attentive bias of different hyperedges independent of the query position. It assigns the same amount of attention to each joint connected to a certain hyperedge.

Note that terms (a) and (b) can be combined by distributive law and require merely an extra step of matrix addition. Moreover, term (d) has $O(|\mathcal{V}|C^2)$ complexity and thus requires negligible computation in comparison to term (a).

Relational Bias Transformers assume the input tokens to be homogeneous, whereas human body joints are inherently heterogeneous, e.g., each physical joint plays a unique role and thus has different relations to others. Taking the heterogeneity of the skeleton data into account, we propose to represent the inherent relation of each joint pair as a scalar trainable parameter B_{ij}, called Relational Bias (RB). It is added to the attention scores before aggregating the global information:

$$\vec{y_i} = \sum_{j=1}^{n}(A_{ij} + B_{ij})\vec{v}_j, \tag{V.7}$$

4.4. Partition strategy



(a) Empirical partition a.

(b) Empirical partition b.

(c) Learned partition a.

(d) Learned partition b.

Figure V.3.: Visualization of the empirical and learned partitions. Different node colors stand for different subgroups for each partition strategy.

Empirically, human skeletons could be divided into a number of body parts, which have been well studied in previous work [218, 111, 211]. We experimentally show that our Hyperformer with empirical partitions yields excellent performance. However, finding an optimal empirical partition strategy is laborious and the optimal partition strategy is restricted to a certain skeleton with a fixed number of recorded joints. In this chapter, we also provide an approach to automate the search process for an effective partition strategy.

To make the partition matrix learnable, we parameterize and relax the binary partition matrix to its continuous version by applying a softmax along its column axis:

$$\tilde{H} = \{\tilde{h}_{ve} = \frac{\exp(h_{ve})}{\sum_{e=1}^{|\mathcal{E}|}\exp(h_{ve})}; i = 1...|\mathcal{V}|, j = 1...|\mathcal{E}|\}. \tag{V.8}$$

The problem of finding an optimal discrete partition matrix H is thus reduced to learning an optimal continuous partition matrix $\tilde{H}$, which can be optimized jointly with Transformer parameters.

At the end of the optimization, a discrete partition matrix can be obtained by applying an argmax operation along each row of $\tilde{H}$:

$$H = argmax(\tilde{H}). \tag{V.9}$$

Note that a number of different proposals can be easily acquired by varying the initialization of $\tilde{H}$. We experimentally show that all the proposals prove to be reasonable. Interestingly, all the learned proposals are symmetric as shown in Fig. V.3, indicating that symmetry is an important aspect of inherent joint relations.

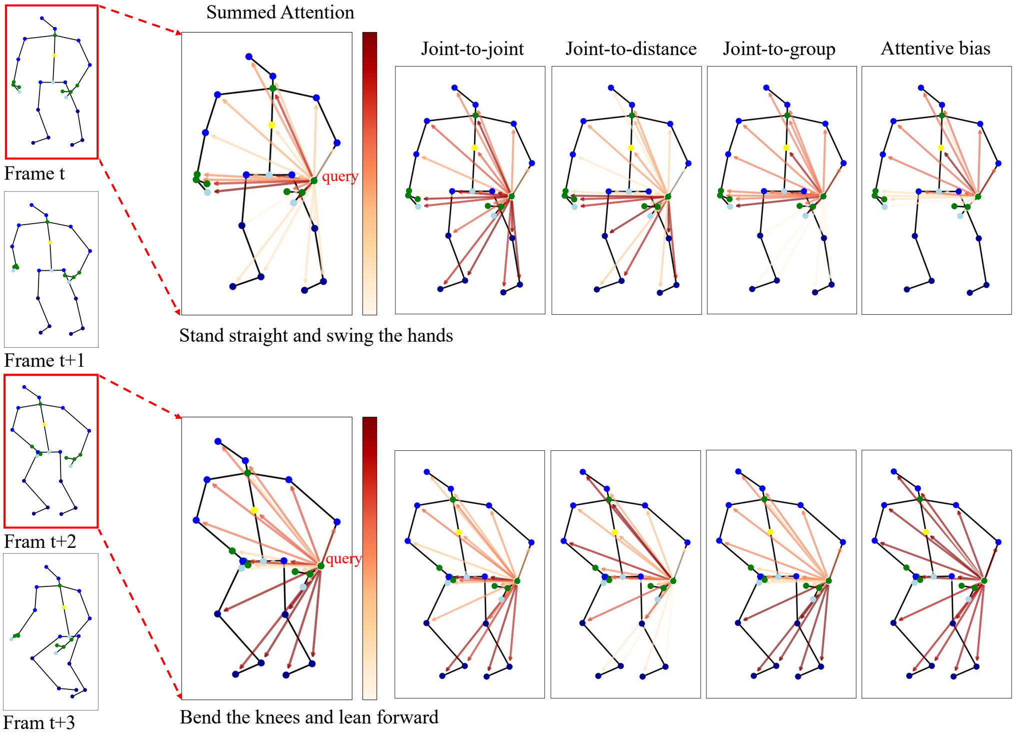


Figure V.4.: Visualization of the attention scores for the action class "Jump Up". The directed edges represent the attention weights w.r.t. the query joint of left wrist and range from light orange to dark red with the increase of the weights. The black edges stand for the bones and the joints are assigned different colors according to their connected hyperedges as in Fig. V.3 (c).

4.5. Model architecture

We first revisit the architectural design of Transformers for skeleton data. Then we built our Hyperformer based on our analysis. HyperSA is employed for spatial modeling of each frame and a lightweight convolutional module is adopted for temporal modeling, following the design of state-of-the-art models [42, 38] in this field.

Spatial Modeling We apply Layer Normalization (LN) before the multi-head HyperSA and add a residual connection to the output, following the standard Transformer architecture [224]. Based on our analysis in Section 1, we further remove the Multi-Layer-Perceptron

(MLP) layers. To introduce non-linearity, a ReLU layer is added after each block of spatial and temporal modeling modules instead.

Temporal Modeling To model the temporal correlation of the human pose, we adopt the Multi-Scale Temporal Convolution (MS-TC) module [38, 149, 42] for our final model. This module contains three convolution branches with a 1 × 1 convolution to reduce channel dimension, followed by different combinations of kernel sizes and dilations. The outputs of convolution branches are concatenated.

Hyperformer is constructed by stacking HyperSA and Temporal Convolution layers alternately as follows:

$$z^{(l)} = \mathrm{HyperSA}(LN(z^{(l-1)})) + z^{(l-1)} \tag{V.10}$$

$$z^{(l)} = \mathrm{TemporalConv}(LN(z^{(l)})) + z^{(l-1)} \tag{V.11}$$

$$z^{(l)} = \mathrm{ReLU}(z^{(l)}) \tag{V.12}$$

5. Experiments

In this section, we first compare our Hyperformer to state-of-the-art approaches on skeleton-based human action recognition benchmarks and show the superior performance of our model. Then we conduct an ablation study for a deeper understanding of our proposed HyperSA. Finally, we evaluate our approach qualitatively by visualizing each component of HyperSA.

5.1. Experimental settings

Datasets

We evaluate our proposed Hyperformer on three commonly adopted public datasets NTU-RGB+D [198], NTU-RGB+D120 [144] and Northwestern-UCLA [227] which are briefly introduced in the following.

NTU RGB+D [198] is a widely used dataset for skeleton-based human action recognition. There are two benchmarks for evaluation including Cross-Subject (X-Sub) and Cross-View (X-View) settings. For X-Sub, the training and test sets come from two disjoint sets of 20 subjects each. For X-View, the training set contains 37920 samples captured by the camera views 2 and 3, and the test set includes 18960 sequences captured by camera view 1.

NTU RGB+D 120 [144] is an extension of NTURGB+D dataset with additional skeleton sequences over 60 additional action classes. It is currently the largest available dataset with 3D joint annotations for human action recognition and contains 32 setups, each of which represents a different location and background.

Northwestern-UCLA [227] dataset is recorded by three Kinect sensors from different viewpoints. It includes 1494 video sequences of 10 action categories.

Implementation details

All experiments are conducted with the PyTorch [171] deep learning library. We train the model for a total number of 140 epochs with standard cross-entropy loss. The learning rate is initialized to 0.025 and reduced at 110 and 120 epochs by 0.1, basically following the strategy in [42]. For NTU RGB+D and NTU RGB+D 120, the batch size is set to 64, each sample is resized to 64 frames, and we adopt the code of [267] for data pre-processing. For Northwestern-UCLA, we use a batch size of 16, and follow the data pre-processing in [41, 38]. Our code is based on the official implementations of [221], [38] and [267]. We employ a model with a total number of 10 layers and 216 hidden channel dimensions for all the experiments.

Table V.1.: Action classification performance on the NTU RGB+D and NTU RGB+D 120 dataset. Following the common setup, we report results using 4 modalities for a fair comparison. We denote the methods that are not directly comparable with * (rely on additional supervision signal or change the standard input data) and mark the methods of which the code is unavailable for reproduction with gray. Please refer to Section 5.2 for more details. InfoGCN [42] reports their results by ensembling 6 modalities, so we use our reproduced results using 4 modalities for a fair comparison.

Category	Methods	Model Type	Loss	Modalities	Parameters	FLOPs	NTU RGB+D 60		NTU RGB+D 120	
							X-Sub(%)	X-View(%)	X-Sub(%)	X-Set(%)
Model	Shift-GCN [41]	GCN	Cross-Entropy loss	J+B+JM+BM	2.8M	10.0G	90.7	96.5	85.9	87.6
	DC-GCN+ADG [40]	GCN		J+B+JM+BM	4.9M	25.7G	90.8	**96.6**	86.5	88.1
	MS-G3D [149]	GCN		J+B+JM+BM	2.8M	48.8G	91.5	96.2	86.9	88.4
	MST-GCN [39]	GCN		J+B+JM+BM	12.0M	408.7G	91.5	**96.6**	87.5	88.8
	EfficientGCN-B4 [210]	Hybrid		J+B+JM+BM	2.0M	15.2G	91.7	95.7	88.3	89.1
	CTR-GCN [38]	Hybrid		J+B+JM+BM	1.5M	11.2G	92.4	96.4	88.9	90.4
	ST-TR [176]	Transformer		J+B+JM+BM	12.1M	259.4G	89.9	96.1	82.7	84.7
	DSTA [207]	Transformer		J+B+JM+BM	4.1M	64.7G	91.5	96.4	86.6	89.0
	3Mformer (3rd-order) [228]	Transformer	-	J+B+JM+BM	2.1M	35.5G	91.3	**97.0**	87.5	89.7
	Hyperformer	Transformer		J+B+JM+BM	2.6M	14.8G	**92.9**	96.5	**89.9**	**91.3**
Training	InfoGCN* [42]	Hybird	+ MMD losses	J+B+JM+BM	1.6M	-	92.3*	96.5*	89.2*	90.6*
	CTR-GCN* [112]	Hybrid	+ Graph Contrastive Learning	J+B+JM+BM	1.5M	-	**93.1***	**97.0***	89.5*	91.0*
	CTR-GCN* [243]	Hybrid	+ Language Supervised Training	J+B+JM+BM	1.5M	-	92.9*	**97.0***	**89.9***	**91.1***
	InfoGCN* [42]	Hybird	+ MMD losses	J	1.6M	-	89.4*	95.2*	84.2*	86.3*
	CTR-GCN [38]	Hybrid		J	1.5M	11.2G	89.8	94.8	84.9	86.7
	Hyperformer	Transformer		J	2.6M	14.8G	**90.7(+0.9)**	**95.1(+0.3)**	**86.6(+1.7)**	**88.0(+1.3)**

Table V.2.: Action classification performance on the Northwestern-UCLA dataset.

Type	Methods	Acc (%)
CNN	VA-CNN (aug.) [265]	90.7
	Ta-CNN [246]	96.1
GCN	4s-shift-GCN [41]	94.6
	DC-GCN+ADG [40]	95.3
Hybrid	CTR-GCN [38]	96.5
	InfoGCN [42] (*with MMD losses)	96.6*
Transformer	**Hyperformer**	**96.7**

5.2. Comparison with state-of-the-art approaches

Following most recent state-of-the-art approaches [41, 252, 38, 39], we adopt a multi-stream fusion strategy, i.e., there are 4 streams which take different modalities including **joint**, **bone**, **joint motion** and **bone motion** as input respectively. Joint modality refers to the original skeleton coordinates; bone modality represents the differential of spatial coordinates; joint motion and bone motion modalities use the differential on temporal dimension of joint and bone modalities, respectively. The softmax scores of 4 streams are added to obtain the fused score.

The comparison on the three datasets is shown in Table V.1 and Table V.2, respectively. As shown in Table V.1, our model reaches state-of-the-art results on all benchmarks, except the Cross-View Setup of NTU RGB+D dataset. For Cross-View bencharmk, our Hyperformer performs slightly worse than DC-GCN [40] and MST-GCN [39]. Nevertheless, these models have much larger sizes and computation budgets. The methods which rely on additional losses are also not directly comparable. For example, InfoGCN [42] relies on two additional MMD loss terms and a number of associated hyperparameters, including loss coefficients, noise ratio and z prior gain. Language Supervised Training relies on pretrained LLMs (GPT-3&CLIP) to leverage the correlations between lables for extra supervision. NTU RGB+D 120 is a more challenging dataset, on which the results of previous Transformer-based approaches are not satisfying. On the contrary, our model achieves the best results among all model categories. This conforms to the findings that attention mechanism benefits more from a large amount of data than Convolution [63, 221]. It is noteworthy that later work 3Mformer [228] proposes another kind of High-order Attention, but achieves inferior performance comparing to our Hyperformer and is computationally heavier. Moreover, their code is unavailable for reproduction. The Northwestern-UCLA dataset is particularly challenging since it contains much fewer training samples, making it even harder for Transformer-based models to compete. With the prior knowledge of bone connectivity and underlying high-order relations of joints, our Hyperformer still yields state-of-the-art results in such a low-data regime.

5.3. Ablation study

In this section, we revisit the role of MLP layers for this task and compare Hyperformer with the vanilla Transformer to show the effectiveness of our proposed model. We also analyze the contribution of each component of our HyperSA. In addition, we compare our learned partition strategy with empirical ones to show its effectiveness. For the ablation study, all experiments are conducted on the X-sub benchmark of NTU RGB+D using the joint modality as input only.

The design of Hyperformer

Before replacing standard SA layers with our HyperSA, we removed the MLP layers based on the results in Table V.3. As can be seen, our HyperSA layers contribute most to the final performance, with an significant improvement of 2.8% absolute accuracy. The MS-TC module further improves over vanilla TC by 0.4%. In Table V.3, it can be seen that our Hyperformer achieves significantly higher accuracy than the vanilla baseline with fewer parameters thanks to the listed design choices. We provide more detailed results in Table V.4, validating the effectiveness of each individual HyperSA components.

Table V.3.: Constructing Hyperformer from the baseline. Note that the MS-TC module in our final model has fewer parameters than vanilla Temporal Convolution (TC) due to the dimension reduction via 1x1 convolutions, see Fig. V.2

Model	Parameters	FLOPs	Acc(%)
SA + MLP + TC	7.2M	25.6G	88.3
SA + TC	3.6M	14.1G	87.5
HyperSA + TC	4.1M	16.7G	90.3
HyperSA + MS-TC	2.6M	14.8G	90.7

Table V.4.: The effectiveness of the HyperSA components.

Model	Acc(%)
SA + TC	87.5
SA + TC + Joint-to-hyperedge attn	89.6
SA + TC + K-Hop RPE	89.5
SA + TC + Hyperedge Attentive Bias	89.6
Full HyperSA + TC	90.3

Effect of different partition strategies

Table V.5.: The effect of different partition strategies

Partition Strategy	Acc(%)
Body Parts	90.5
Upper and Lower Body	90.4
Learned a	90.7
Learned b	90.7

In Table V.5, we compared empirical partitions (see Fig. V.3) to the learned partitions. Note that we obtain different partition proposals when the model is trained with different seeds. All proposals prove to be effective. Overall, Hyperformer delivers stable performance and achieves the best result with a learned partition using the approach described in Section 4.4.

5.4. Qualitative results

In order to showcase the effectiveness of our approach, we visualize the attention scores of HyperSA and the four decomposed terms in Eq. (V.6) at the first layer in Fig. V.4. More specifically, we draw the attention scores w.r.t. the query joint of left wrist for two single frames respectively. The directed edges represent the attention weights and range from light orange to dark red as attention score increases. The black edges stand for the bones and the joints are assigned different colors according to their connected hyperedges.

At Frame t, the person stands straight and starts to swing the hands, preparing to jump up. The joint-to-joint attention is distracted by a large number of joints, whereas the joint-to-group attention concentrates on the upper body. As the sum of the four terms, the final attention reasonably focuses on the hands and neck.

At Frame $t+2$, the person bends the knees and leans the upper body forward to squeeze the leg muscles. Although the joint-to-joint attention is successfully attached to the feet and waist, the knees and heels are less valued. This incomplete and unstable attention is unavoidable due to the pairwise mechanism. However, our joint-to-group attention solves this issue by exploiting the underlying group relation.

6. Conclusion

In this chapter, we successfully incorporate the information of skeletal structure into Transformer by proposing a relative positional embedding based on graph distance. This is a more elegant solution than previous hybrid models. Moreover, we identified a limitation of graph models for the task of skeleton-based action recognition, i.e., high-order joint relations are ignored. Therefore, we propose a novel HyperSA layer to make Transformers aware of these inherent relations. The resulting Hyperformer is the first Transformer that establishes the state-of-the-art performance.

Chapter VI.

Extending the Structural Prior with Topological Analysis beyond Connectivity

This chapter extends the structural encoding approach introduced previously by incorporating topological features derived from persistent homology. While earlier methods focused on capturing joint connectivity through graph distances, we now extract stable, multi-scale topological summaries that reflect invariant structural patterns in skeletal data. This shift moves beyond local connectivity to more global, shape-informed representations.

Building on our published work [283], we integrate these topological priors into Graph Convolutional Networks for action recognition, achieving state-of-the-art results in skeleton-based tasks. The findings demonstrate that our topological encoding offers a more expressive representation than traditional connectivity-based priors.

1. Introduction

Skeleton-based action recognition has undergone a significant transformation, driven by the need for computational efficiency and adaptability to varying environmental conditions, particularly in fields such as medical applications. Early pioneering efforts predominantly utilized Recurrent Neural Networks (RNNs) [64, 212, 266] and Convolutional Neural Networks (CNNs) [122, 145], extracting features or pseudo-images from human joint data to make predictions. Despite reasonable performance, these approaches were inherently constrained in modeling the intricate inter-dependencies between joints, which is crucial for fine-grained action recognition.

Graph Convolutional Networks (GCNs) [52, 125, 193] have the potential to learn the topology, but they fail to fully exploit the inherent skeletal structure due to two key limitations: (1) The topology is initialized based on physical connections, but this vital knowledge decays during training, limiting the retention of skeletal information. (2) The single static topology struggles to capture diverse joint relationships that emerge across complex actions. Therefore, while the graph modeling of GCNs is better suited to handle skeletal data than CNNs or RNNs, they have difficulty fully capturing the intricate multi-scale relationships within the human topology, which are crucial for sophisticated skeleton-based action recognition.

Moreover, the single static graph topology in GCNs has limited expressivity to encapsulate the multi-scale semantic relationships that emerge through the hierarchical representation learning process. Recent advanced methods have sought to mitigate this issue by incorporating learnable topologies with impressive adaptability (e.g., [38, 40]). However, as evidenced by empirical analysis, such techniques still tend to lose valuable

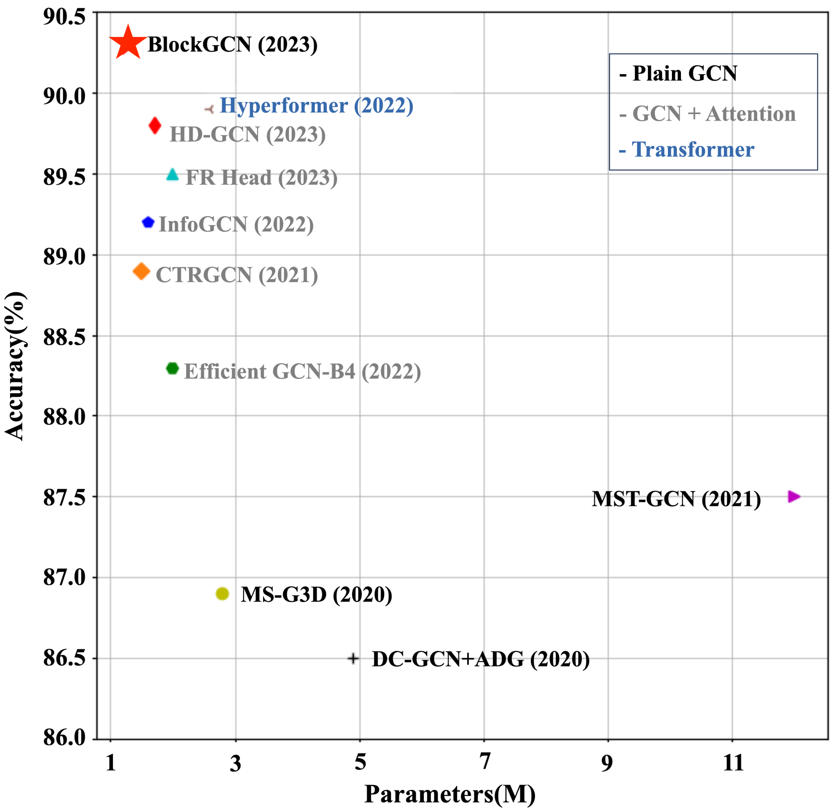
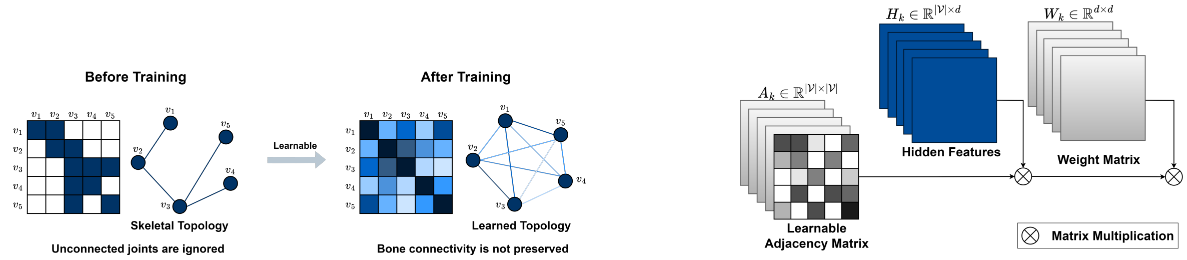


Figure VI.1.: Performance vs. Model Size on NTU RGB+D 120 Cross-Subject. Our BlockGCN improves over previous methods w.r.t. both performance and efficiency. [Best viewed zoomed in]

topological knowledge acquired from the physical connections during network training. While learnable topologies provide modeling flexibility, vital inductive biases from the inherent skeletal topology are not effectively retained. The enriched semantics captured in the optimized topology tend to deviate from the underlying physical connections, leading to detrimental topological knowledge forgetting.

To remedy the topology fading issue, we propose a novel Topological Encoding approach that represents the skeletal structure through relative distances between joint pairs on the skeletal graph (Sec.3.2). This enables a more robust characterization of the physical connections. Complementing this static encoding, we introduce an action-specific scheme using persistent homology analysis – the resulting topological descriptor provides vital insights into the skeletal dynamics across actions (Sec.3.2). Furthermore, we demonstrate the redundancy in existing GCNs for multi-relational modeling (Sec. 3.1). To capture substantial joint relationship variations across complex actions, current state-of-the-art GCNs widely adopt ensemble convolutions and attention mechanisms, at the cost of increased computation. To further address this inefficiency, we propose BlockGC, a significant refinement to the standard Graph Convolution (Sec. 3.3). BlockGC proves to be highly effective and efficient for multi-relational reasoning, reducing parameters by over 40% while elevating performance beyond original GCNs. The key contributions of the work presented in this chapter are summarized as three-fold:

(i) Identifying and restoring the overlooked skeletal topology in advanced GCNs via novel topological encoding schemes. This includes a static encoding using graph distances to retain bone connectivity and a dynamic encoding based on persistent homology to capture action-specific topology.


Before Training
v_1 v_2 v_3 v_4 v_5
Skeletal Topology
Unconnected joints are ignored
Learnable
After Training
Learned Topology
Bone connectivity is not preserved
$H_k \in \mathbb{R}^{|\mathcal{V}|\times d}$
$W_k \in \mathbb{R}^{d\times d}$
$A_k \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$
Hidden Features
Weight Matrix
Learnable Adjacency Matrix
Matrix Multiplication

(a) Skeletal information is lost after training. Darker color is larger weight.

(b) Ensemble of GCs (default choice of SOTA models).

Figure VI.2.: We reveal the remaining issues of previous GCNs, namely "catastrophic forgetting" of skeletal topology with learnable topology (see Fig. VI.2a) and inefficient modeling of multi-relational joint co-occurrences (see Fig. VI.2b).

(ii) Devising BlockGC, an efficient and powerful graph convolutional block that reduces parameters by over 40% while elevating modeling capabilities beyond original GCNs, enabled by its block diagonal weight matrix.

(iii) Establishing new state-of-the-art performance on standard benchmarks without reliance on extra supervision or attention. Our method demonstrates consistent improvements averaging over 0.8% in accuracy against previous best-performing approaches.

2. Related Work

2.1. Skeleton-based Action Recognition

Early approaches to skeleton-based action recognition relied on Recurrent Neural Networks (RNNs) due to their ability to handle temporal dependencies [64, 212, 266]. Convolutional Neural Networks (CNNs) were also employed, but they were found to be less effective in explicitly capturing spatial interactions among body joints [122, 145]. Consequently, the focus shifted to Graph Convolutional Networks (GCNs), which extend convolution operations to non-Euclidean spaces and enable the explicit modeling of joint spatial configurations [84, 247]. In the following, we primarily focus on these graph-based models as they more comprehensively capture spatial relationships.

2.2. GCNs for Skeleton-based Action Recognition

Graph Convolutional Networks (GCNs) have significantly impacted skeleton-based action recognition. We discuss previous GCNs in terms of the following aspects:
Adjacency Matrix: The choice of adjacency matrix in GCNs is crucial. Early works, such as [249], used a fixed topology based on bone connectivity, demonstrating the effectiveness of GCNs in action recognition. However, this rigid topology has inherent limitations. Recent approaches have explored learnable adjacency matrices to capture relationships between physically connected and disconnected joints [38, 40, 42, 78, 149, 208, 210, 242, 172, 178]. Our work builds on this idea and addresses the Catastrophic Forgetting associated with learnable adjacency matrices, proposing a method to preserve bone connectivity.

Multi-relational Modeling Methods	Complexity	Parameters
Vanilla GC (Baseline)	$\mathcal{O}(\lvert\mathcal{V}\rvert d^2)$	$d^2 + \lvert\mathcal{V}\rvert^2$
Ensemble of GCs	$\mathcal{O}(K\lvert\mathcal{V}\rvert d^2)$	$Kd^2 + K\lvert\mathcal{V}\rvert^2$
Ensemble of Adjacency Matrices	$\mathcal{O}(\lvert\mathcal{V}\rvert d^2)$	$d^2 + K\lvert\mathcal{V}\rvert^2$
Proposed BlockGC	$\mathcal{O}(\frac{\lvert\mathcal{V}\rvert d^2}{K})$	$\frac{d^2}{K} + K\lvert\mathcal{V}\rvert^2$

Table VI.1.: Comparison of different approaches for multi-relational modeling. We denote the number of body joints, the number of hidden dimensions, and the number of groups/ensembles with $|\mathcal{V}|$, d and K respectively, where d is much larger than $\mathcal{V}$. Our BlockGC has the least complexity and parameters but achieves the best performance.

Relative Positional Encodings: Relative positional information has proven important in various domains, including Natural Language Processing [50, 98, 201] and Computer Vision [240, 282, 147, 138, 139]. While relative positional encoding has been demonstrated to be beneficial for Transformers on graph data [254], its significance for GCNs, especially in the field of skeleton-based action recognition, remains unexplored. Our work aims to fill this gap by proposing a novel method for relative positional encoding that preserves the essential topological invariances in skeleton data.

Multi-Relational Modeling: Capturing multiple semantic relations with a single adjacency matrix is challenging. Previous studies have proposed strategies to overcome this limitation. One approach is the ensemble of GCs, as employed by Yan et al. [249], where three parallel GCs at each layer are intended to operate on different partitions of joints according to their distances to a reference node. However, we observed that each adjacency matrix tends to become fully connected after learning, rendering the handcrafted partitions ineffective. This setup is equivalent to ensembling multiple GCs at each layer, a technique adopted in subsequent works [38, 40, 42, 149, 208, 249, 267]. Moreover, DecouplingGCN [40] uses multiple adjacency matrices and a shared weight matrix for different feature subsets, improving efficiency but reducing expressiveness.

Another approach is attention-based adaptation, as employed in recent works [38, 42, 78, 208, 278], which incorporate attention mechanisms or similar techniques to create a data-dependent component of the topology, similar to Graph Attention Networks [225] and Graphormer [254]. This approach allows for the dynamic adjustment of joint connections based on relevance but is computationally heavy and requires extensive data for optimal performance. In contrast to the above-mentioned approaches, our proposed BlockGC enables the full power of multi-relational modeling by assigning a unique subset of weights to each feature group, while being the most efficient thanks to its sparse convolution weight matrix.

3. Method

In this section, we initially compare Graph Convolutional Networks (GCNs) that utilize learnable adjacency matrices with Fully Connected Networks (FCNs). Through a combination of theoretical and experimental analyses, we identify two primary challenges: 1) catastrophic forgetting of skeletal topology and 2) inefficient multi-relational modeling(Sec. 3.1). To combat these limitations, we introduce a series of enhancements: 1) Topology Encoding aimed at retaining key skeleton properties (Sec. 3.2), and *2) an enhanced graph convolution*, termed BlockGC, designed to capture the implicit relations within joints at minimum cost (Sec. 3.3). The above innovations lead to the core building block of our Model, as shown in (see Fig. VI.4a).

3.1. Problem Formulation

Within the realm of skeleton-based action recognition, the skeletal topology is inherently defined as a graph $\mathcal{G}_\mathcal{S} = (\mathcal{V}, \mathcal{E})$, where the vertices $\mathcal{V}$ represent the body's joints, and the edges $\mathcal{E}$ illustrate the connections between joints through bones. As a result, nearly all cutting-edge methods [38, 40, 149, 208, 210, 242] consistently adopt the graph convolution proposed by by [125], due to its simplicity and strong resistance to over-fitting:

$$H^{(l)} = \sigma(A^{(l)} H^{(l-1)} W^{(l)})\,, \tag{VI.1}$$

where $A^{(l)} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ is the adjacency matrix employed for spatial aggregation, $H^{(l)} \in \mathbb{R}^{|\mathcal{V}|\times T\times d}$ symbolizes the hidden representation, and $W^{(l)} \in \mathbb{R}^{d\times d}$ is the weight matrix utilized for feature projection. Here, $|\mathcal{V}|$, T, and d denote the number of joints, frames, and hidden features, respectively. σ is the non-linear ReLU activation function, and the superscript l indicates the layer number. Despite GCNs seeming adept at learning human skeleton characteristics effectively, our experimental validation shows that this is not entirely the case. To sum up, there are two main issues in existing GCNs, which will be analyzed below.

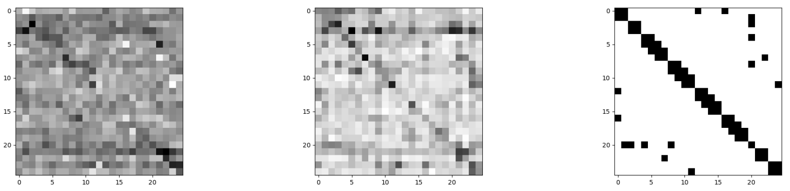
P1: Catastrophic Forgetting of skeletal topology: Prior research can generally be categorized into two groups: one [249] where the adjacency matrix is fixed to portray the skeleton topology, and the other [38, 40, 42, 208] where the adjacency matrix is optimized during training via gradient backpropagation[1]. Despite these advancements, GCNs (Eq. VI.1) have been observed to struggle with accurate recognition of complex actions [40]. We hypothesize that this performance bottleneck is related to the adjacency matrix A, as it "catastrophically forgets" the skeleton topology during training. Our goal is to validate this hypothesis through both theoretical and experimental approaches.

Theoretically, Graph Convolution with a learnable adjacency matrix can be interpreted as a fully connected layer with a weight matrix $W_{spatial} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$. In this light, GCNs resemble ResMLP [220] and MLP-Mixer [219], which belong to a special type of Fully-Connected-Networks (FCNs) for image classification. Similarly, GCNs with a learnable adjacency matrix suffer from catastrophic forgetting [169] as FCNs during training, resulting in the inability to preserve the original topological representation in the adjacency matrix A, see Fig. VI.2a.

From an experimental perspective, we have rigorously confirmed the catastrophic forgetting of skeleton topology (Table VII.1). Our results demonstrate that GCNs' performance remains similar irrespective of the initialization states, suggesting that existing GCNs entirely fail to maintain the topological skeleton in the adjacency matrix A.

To validate our analysis that the information of bone connectivity is lost after training. We also examined the learned weights of adjacency matrices at each layer of the GCN baseline model. The statistics are provided in Fig. VI.3. As shown in the figure, the learned adjacency matrices are totally different from each other at each layer, although they are all initialized according to the bone connections.

[1] For details, please refer to related work.



(a) Mean of A. (b) Std of A. (c) Bone connections.

Figure VI.3.: The statistics (mean and standard deviation) of the learned adjacency matrices of GCN (Darker colors stand for larger weights). It can be seen that the learned weights vary dramatically among different layers and deviate far from the bone connections, which are used for initialization at the beginning of training.

P2: Inefficient multi-relational modeling: The interactions between joints are action-dependent. For instance, during running, the movement of hands and feet primarily serves to maintain balance, whereas when removing shoes, hands and feet interact more directly and play a dominant role. Therefore, it is clear that a single adjacency matrix A in a classic GCN (Eq. VI.1) cannot capture more than one type of interaction.

To overcome this issue, previous work has proposed the use of a layer-wise ensemble of GCs or adjacency matrices (see Fig. VI.2b) and attention-based adaptation. For ensembles of GCs, both parameters and computation increase linearly with the number of ensembles, causing the model to become excessively large with many ensembles and to suffer from over-fitting. As a result, the number of ensembles is typically limited to three.

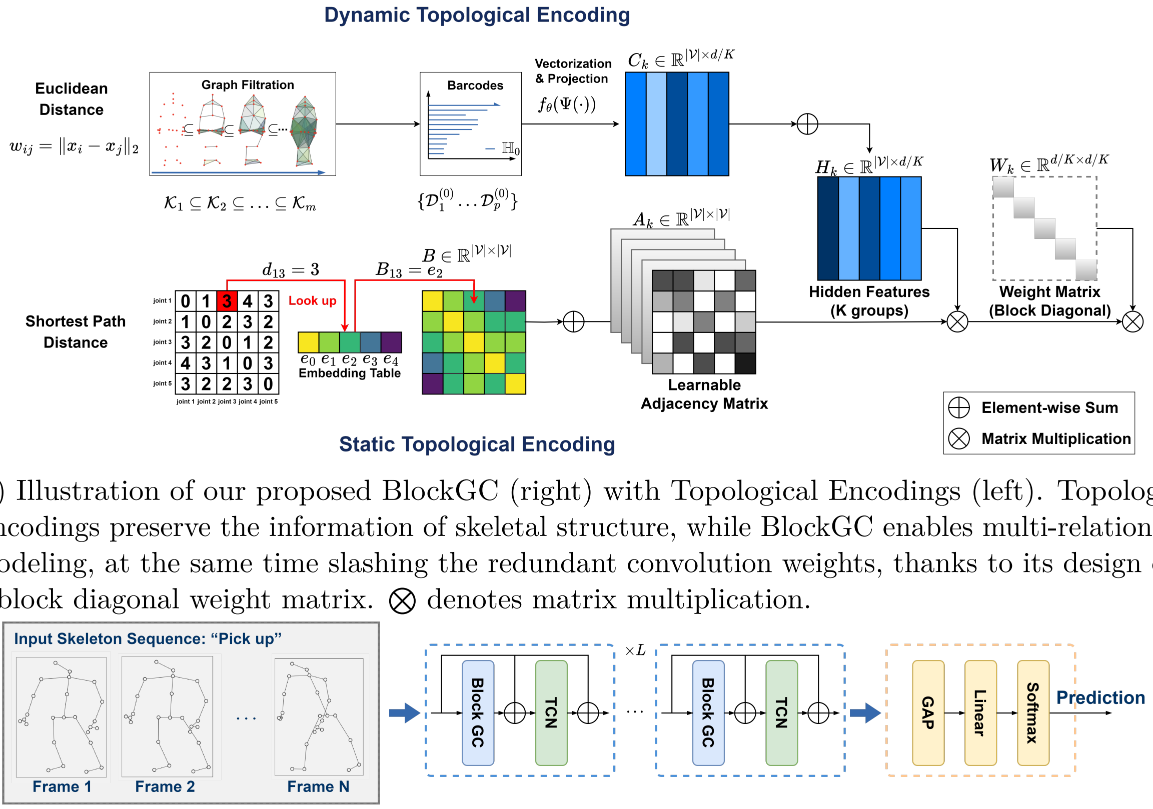
For the ensemble of adjacency matrices [40] and attention-based adaptation [38, 42], a single weight matrix is applied across the entire feature dimension, which constrains the modeling capacity. Furthermore, our experimental results demonstrate that a significant portion of the weight matrix is redundant (see Table VI.6).

3.2. Topological Encoding

GCNs with trainable adjacency matrices A tend to become insensitive to the underlying skeletal topology, i.e., the bone connections, after training. However, incorporating bone connections is beneficial as they convey substantial information about the action being performed, such as how the bone connections physically constrain joint movements. To address this issue, we introduce a method termed Topological Encoding, which efficiently preserves such static information during training. Additionally, we consider the dynamic topological features of the input pose sequence through persistent homology analysis, which further illustrates the self-organizing dynamics in each action class. These two complementary modules provide rich skeleton descriptions to enhance the representation ability of GCNs. Theoretical explanations and intuitive descriptions of the persistent homology analysis are provided in the supplementary material for further reference.

Static Topological Encoding

Bones connect the joints of the human body, physically restricting each joint's movement during an action. It is crucial to integrate this bone connectivity information to accurately recognize the action. We propose a Static Topological Encoding to describe the skeletal connection. This method encodes the relative distance between two joints on the skeletal graph $\mathcal{G}_\mathbf{S}$, using different distance measures such as Shortest Path Distance (SPD) or


Dynamic Topological Encoding
Euclidean Distance
$w_{ij} = \|x_i - x_j\|_2$
Graph Filtration
$\mathcal{K}_1 \subseteq \mathcal{K}_2 \subseteq \ldots \subseteq \mathcal{K}_m$
Barcodes
$\mathbb{H}_0$
$\{\mathcal{D}_1^{(0)} \ldots \mathcal{D}_p^{(0)}\}$
Vectorization & Projection
$f_\theta(\Psi(\cdot))$
$C_k \in \mathbb{R}^{|\mathcal{V}| \times d/K}$
$H_k \in \mathbb{R}^{|\mathcal{V}| \times d/K}$
Hidden Features (K groups)
$W_k \in \mathbb{R}^{d/K \times d/K}$
Weight Matrix (Block Diagonal)
$A_k \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$
Learnable Adjacency Matrix
Shortest Path Distance
$d_{13} = 3$
$B_{13} = e_2$
$B \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$
Look up
joint 1 0 1 3 4 3
joint 2 1 0 2 3 2
joint 3 3 2 0 1 2
joint 4 4 3 1 0 3
joint 5 3 2 2 3 0
joint 1 joint 2 joint 3 joint 4 joint 5
e_0 e_1 e_2 e_3 e_4
Embedding Table
Static Topological Encoding
Element-wise Sum
Matrix Multiplication

(a) Illustration of our proposed BlockGC (right) with Topological Encodings (left). Topology Encodings preserve the information of skeletal structure, while BlockGC enables multi-relational modeling, at the same time slashing the redundant convolution weights, thanks to its design of a block diagonal weight matrix. $\otimes$ denotes matrix multiplication.

Input Skeleton Sequence: "Pick up"
Frame 1
Frame 2
...
Frame N
Block GC
TCN
$\times L$
...
Block GC
TCN
GAP
Linear
Softmax
Prediction

(b) Model architecture of our BlockGCN. L denotes the number of stacked layers.

Figure VI.4.: Visualization of our proposed approach.

distance in a level structure [59]. We adopt SPD for our final model due to its simplicity.

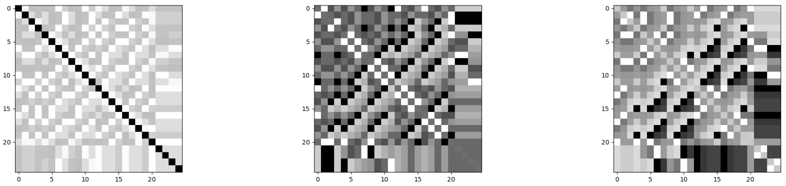
$$\begin{aligned} B_{ij} &= e_{d_{i,j}} \quad \text{with} \\ d_{i,j} &= \min_{P \in Paths(\mathcal{G}_\mathbf{S})} \{|P|, P_1 = v_i, P_{|P|} = v_j\} \end{aligned}, \tag{VI.2}$$

where P_1 and $P_{|P|}$ indicate the first and last vertex on the path P, and the weight parameter B_{ij} is retrieved from a trainable parameter table $E = \{e_{\text{index}}\}$ and then assigned to each joint pair according to their shortest path distances $d_{i,j}$ through bone connections, as shown in Fig. VI.4a. In this way, only the embedding weights, instead of adjacency matrices, are optimized during training, ensuring that the bone connectivity information represented by joint distances is preserved. The learned static topological encoding is shown in Fig. VI.5. By incorporating the Static Topological Encoding, our model effectively captures the essential structural information of the skeleton, leading to improved action recognition performance.

Dynamic Topological Encoding

The algebraic topology tool of persistent homology [68] was proposed to extract characteristics of topological objects of connected components and cycles of graph persist across multiple scales [5], showing to be efficient in graph representation extraction [271, 186]. By encoding graphs into simplicial complexes, novel descriptors are exposed, which include essential information on action-specific dynamics.

Given an input pose sequence, a weighted dynamic graph $\mathcal{G}_\mathbf{D}$ is composed by using skeleton joints as nodes and the *Euclidean distance* between each joint pair as weights



(a) Layer 1. (b) Layer 2. (c) Layer 3.

Figure VI.5.: The learned Static Topological Encoding. It shows that the learned weights are diverse and adapted to different levels of semantics at each layer.

denoted by w_{ij}. The key idea for persistent homology analysis is that we consider the *filtration* of $\mathcal{G}_{\mathbf{D}}^{\epsilon_1} \subseteq \mathcal{G}_{\mathbf{D}}^{\epsilon_2} \subseteq \ldots \mathcal{G}_{\mathbf{D}}^{\epsilon_m} = \mathcal{G}$ instead of a single object of $\mathcal{G}_{\mathbf{D}}$. With $\mathcal{K}$ representing the abstract simplicial complexes for each graph and $\mathcal{K}_i = \mathcal{G}_{\mathbf{D}}^{\epsilon_i}$ (in which $i = 1, 2, \ldots, m$ denotes one of the subgraph or subcomplex), the *graph filtration* is defined as:

$$\emptyset = \mathcal{K}_0 \subseteq \mathcal{K}_1 \subseteq \mathcal{K}_2 \ldots, \subseteq \mathcal{K}_m = \mathcal{K} \tag{VI.3}$$

We apply *Vietoris-Rips complex* [5] to build simplicial complexes from the graphs due to its computational advantages. Through the graph filtration construction, the birth-death barcodes of different topological objects are extracted as summaries of the graph topology. The corresponding *persistence diagram* of the barcodes is presented as a multi-set in $\mathbb{R}^2$ of $\{\mathcal{D}_1^0, \mathcal{D}_2^0, \ldots, \mathcal{D}_p^0\}$ in which $\mathcal{D}_i^0 = \{(b_i^0, d_i^0)\}$ and k means the number of connected components (here the superscript 0 denotes the 0-dimensional homology named *connected components*, while 1-dimensional homology for *cycles*). As shown in Fig. VI.6, the obtained barcodes reveal clear inter-action similarities and intra-action differences.

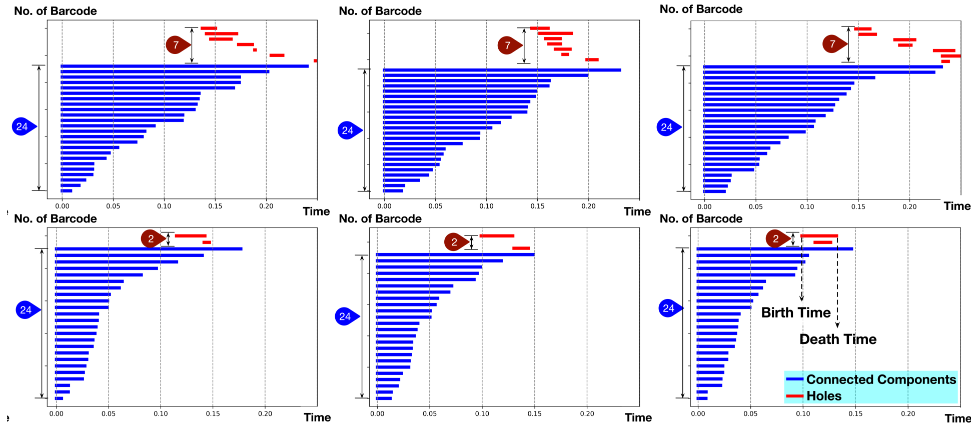


Figure VI.6.: Barcodes of "brush hair" (top) and "shake hands" (bottom). Large Inter-action similarities and intra-action differences can be observed among different samples in each group.

Then we adopt the differentiable vectorization [106] $\Psi^0 : \{\mathcal{D}_1^0, \mathcal{D}_2^0, \ldots, \mathcal{D}_p^0\} \to \mathbb{R}^{|\mathcal{V}| \times d'}$ on the barcodes, and project the obtained representation to GCN hidden layers' feature space through a mapping $f_\theta : \mathbb{R}^{|\mathcal{V}| \times d'} \to \mathbb{R}^{|\mathcal{V}| \times d}$ at each layer:

$$C = f_\theta \left(\Psi^0 \left(\mathcal{D}_1^0, \mathcal{D}_2^0, \ldots, \mathcal{D}_p^0 \right) \right), \tag{VI.4}$$

where f_θ is parameterized by a linear layer. The procedure is depicted in Fig. VI.4a. This encoding is input-dependent and hence termed "dynamic".

Finally, we add the obtained static and dynamic topological encoding to the adjacency matrix and hidden feature at each layer (the superscript (l) denotes the layer number), respectively, to obtain the final formulation of spatial aggregation:

$$H^{(l)} = \sigma((A^{(l)} + B^{(l)})(H^{(l-1)} + C^{(l)})W^{(l)}). \tag{VI.5}$$

3.3. Efficient Multi-Relational Modeling

Joint co-occurrences inherently involve multiple relations, as discussed in Section 3.1, which necessitate modeling various semantics. A single adjacency matrix is insufficient to handle such complexity. Previous approaches, detailed in Section 2, have limitations in computational efficiency or theoretical constraints, preventing the full potential of GCNs from being realized. To overcome this, we propose BlockGC, which allows efficient modeling of different high-level semantics. Our proposed BlockGC not only reduces computation and parameters but also proves to be more effective than previous methods.

As illustrated in Fig. VI.4a (top right), the feature dimension is first divided into K groups, and then spatial aggregation and feature projection are applied in parallel within each k^{th} group. The corresponding formula is as follows:

$$H^{(l)} = \sigma(\begin{bmatrix} (A_1 + B_1)(H_1^{(l-1)} + C_1^{(l-1)}) \\ \dots \\ (A_K + B_K)(H_K^{(l-1)} + C_k^{(l-1)}) \end{bmatrix} \begin{bmatrix} W_1^{(l)} & & \\ & \dots & \\ & & W_K^{(l)} \end{bmatrix}) \tag{VI.6}$$

where $H_k \in \mathbb{R}^{|\mathcal{V}| \times T \times d/K}$ and $W_k \in \mathbb{R}^{d/K \times d/K}$. $\{W_k, k = 1, ..., K\}$ are arranged as a block diagonal matrix, leading to parameter reduction and making the projected feature groups independent from each other. This is a desired property, as each group is intended to model a kind of semantics that are also independent from each other. Thanks to the decoupled feature projection, our method enables GCN the full power for multi-relational modeling. Compared to DecouplingGCN [40] and attention-based adaptation of adjacency matrix, our BlockGC not only significantly reduces parameters and computation (BlockGC $\mathcal{O}(\frac{|\mathcal{V}|d^2}{K})$, GC $\mathcal{O}(|\mathcal{V}|d^2)$, Decoupling GC $\mathcal{O}(|\mathcal{V}|d^2)$), but also leads to improved performance.

3.4. Model Architecture

We built our final model, named BlockGCN, based on the above-described Topological Encodings and BlockGC. To model the temporal correlation of the skeleton sequences, we employ the multi-scale temporal convolution module [38, 42, 149]. It consists of three convolution branches with a 1×1 convolution for dimension reduction and different combinations of kernel sizes and dilations. The outputs of convolution branches are concatenated as the final output.

The final model is constructed by stacking our BlockGC and the multi-scale temporal convolution modules alternately 10 times as shown in Fig. VI.4b (the Topological Encodings are omitted for simplification). The final output of our model is produced by applying a global pooling operation over both the joint and temporal dimensions, followed by a softmax operation over the class dimension.

Table VI.2.: Action classification performance on the NTU RGB+D and NTU RGB+D 120 dataset. Following the common setup, we report results using 4 modalities for a fair comparison. We denote the methods that are not directly comparable with * (rely on additional supervision signal or change the standard input data) and mark the methods of which the code is unavailable for reproduction with gray. Please refer to Section 4.2 for more details. InfoGCN [42] reports their results by ensembling 6 modalities, so we use the reproduced results using 4 modalities by [112] for a fair comparison. In this table, we also omit the results without publicly available code for reproduction.

Methods	Publication	Category	Extra Loss/Data	Modalities	Parameters	FLOPs	NTU RGB+D 60		NTU RGB+D 120		NW-UCLA
							X-Sub(%)	X-View(%)	X-Sub(%)	X-Set(%)	
DC-GCN+ADG [40]	ECCV 2020	GCN		J+B+JM+BM	4.9M	1.83G	90.8	96.6	86.5	88.1	95.3
MS-G3D [149]	CVPR 2020	GCN		J+B+JM+BM	2.8M	5.22G	91.5	96.2	86.9	88.4	-
MST-GCN [39]	AAAI 2021	GCN		J+B+JM+BM	12.0M	-	91.5	96.6	87.5	88.8	-
CTR-GCN [38]	ICCV 2021	Hybrid		J+B+JM+BM	1.5M	1.97G	92.4	96.4	88.9	90.4	96.5
EfficientGCN-B4 [210]	TPAMI 2022	Hybrid		J+B+JM+BM	2.0M	15.2G	91.7	95.7	88.3	89.1	-
InfoGCN [42]	CVPR 2022	Hybrid		J+B+JM+BM	1.6M	1.84G	92.3	96.7	89.2	90.7	96.6
FR Head [275]	CVPR 2023	Hybrid		J+B+JM+BM	2.0M	-	92.8	96.8	89.5	90.9	96.8
BlockGCN		GCN		J+B+JM+BM	**1.3M**	1.63G	**93.1**	**97.0**	**90.3**	**91.5**	**96.9**
CTR-GCN* [243]	ICCV 2023	Hybrid	+ Language Supervison	J+B+JM+BM	-	-	92.9*	97.0*	89.9*	91.1*	97.2*
HDGCN* [134]	ICCV 2023	Hybrid	Without Motion Modality	J+B+J'+B'	1.7M	1.77G	93.0*	97.0*	89.8*	91.2*	96.9*
InfoGCN [42]	CVPR 2022	Hybrid		J	1.6M	1.84G	89.8	95.2	85.1	86.3	-
HDGCN [134]	ICCV 2023	Hybrid		J	1.7M	1.77G	-	-	85.7	87.3	-
FR Head [275]	CVPR 2023	Hybrid		J	2.0M	-	90.3	95.3	85.5	87.3	-
BlockGCN		GCN		J	**1.3M**$^{\downarrow 0.7}$	1.63G	**90.9**$^{\uparrow 0.6}$	**95.4**$^{\uparrow 0.1}$	**86.9**$^{\uparrow 1.4}$	**88.2**$^{\uparrow 0.9}$	95.5

4. Experiments

In this section, we comprehensively evaluate our proposed BlockGCN on standard benchmarks for skeleton-based action recognition. Our empirical results showcase that our model exceeds the performance of existing state-of-the-art methods. Furthermore, we present an intricate analysis exploring the significance of topological information within GCN-based models for action recognition. We also carried out an ablation study to assess the efficacy of our novel Topological Encodings and BlockGC. Remarkably, we employ the standard cross-entropy loss in all our experiments to ensure an impartial assessment of our architecture and to uphold direct comparability with prior works. We gauge the performance of our BlockGCN on three widely-used benchmark datasets for skeleton-based human action recognition: NTU RGB+D [198], NTU RGB+D 120 [144], and Northwestern-UCLA [227].

4.1. Implementation Details

Our implementation is mainly based on a Tesla V100 GPU. The model was optimized via Stochastic Gradient Descent (SGD) with Nesterov momentum set at 0.9 and a weight decay of 0.0004 for NTU RGB+D and NTU RGB+D 120, and 0.0002 for Northwestern-UCLA. Our experiments employed cross-entropy loss and initiated the learning rate at 0.05, reducing it by a factor of 10 at epochs 110 and 120. For NTU RGB+D and NTU RGB+D 120, we opted for a batch size of 64 and resized each sample to 64 frames. For Northwestern-UCLA, we selected a batch size of 16. Our implementation builds upon the official code [38] and our training setup follows the strategy used in [42, 38].

4.2. Comparison with State-of-the-art

To establish a fair comparison, we employed the commonly accepted 4-Stream fusion approach in our experiments. In particular, we input four different modalities: *Joint*, *Bone*, *Joint Motion*, and *Bone Motion*. The joint and bone modalities denote the original

Table VI.3.: Ablation on the adjacency matrix initialization. Note that SkeletonGCL [112] proposes a novel framework for training GCNs with additional contrastive loss.

Model	Initialization			
	Bone Connection	Identity Matrix	All Ones	Kaiming Uniform [97]
DecouplingGCN [40]	82.0	81.9	81.7	**82.1**
SkeletonGCL [112]	84.4	84.3	**84.8**	84.3
CTRGCN [38]	84.9	**85.0**	84.8	84.8

skeleton coordinates and their derivatives with respect to bone connectivity, respectively. The joint and bone motion modalities compute the temporal differential of the joint and bone modalities. Subsequently, we amalgamate the predicted scores of each stream to produce the final fused results. For a fair evaluation, we only consider the results of previous methods (e.g., InfoGCN [42], HDGCN [134]) that utilize four modalities.

We compare our BlockGCN with state-of-the-art methods on NTU RGB+D, NTU RGB+D 120, and Northwestern-UCLA in Table VI.2. It is noteworthy that the recently published works [65, 243, 134] are not directly comparable to our method. PoseC3D [65] achieves improved results by incorporating additional RGB input, but this necessitates significant computational overhead. GAP [243] relies on pre-trained LLMs (GPT-3&CLIP) to leverage the correlations between labels for extra supervision. HDGCN [134] replaces the weaker motion modalities with their hand-crafted modalities, which contributes a lot to their final results. As can be seen in the results using joint modality alone, the performance of HDGCN [134] is indeed on par with FR Head [275].

Our BlockGCN establishes new state-of-the-art performance on the common benchmarks, showing an improvement of 0.5% accuracy on average over the previous state-of-the-art FR Head [275]. Notably, this is achieved with 35% fewer parameters. The performance gaps seem mild for all recent approaches mainly because the reported results use an ensemble of 4 modalities. The actual improvement of our method is remarkable (0.8%) when using the single joint modality alone.

4.3. Ablation Analysis

We delve into an experimental evaluation of the effectiveness of each component of our method. All ablation studies are carried out on the X-sub benchmark of NTU RGB+D 120, utilizing a single joint modality, if not specified. We initiate the study by examining the impact of different initializations for the adjacency matrix.

Implications of Adjacency Matrix Initialization: We scrutinize various strategies for initializing the adjacency matrix, ranging from special initialization leveraging physical connections as adopted by previous work, to more topology-agnostic approaches. For this experiment, we engage three different GCNs, as shown in Table VI.3. Our results suggest that simply initializing the adjacency matrix based on physical connections does not suffice to exploit the skeletal topology effectively, thereby inspiring our proposed Topological Encodings to preserve such information.

Effectiveness of Individual Components: We enhance the vanilla GCN baseline by supplanting the vanilla GC with our BlockGC layers and incorporating the topological encodings. Our BlockGC substantially reduces the parameters by 43% ($0.9M$), while

Table VI.4.: Ablation on BlockGC and Topological Encodings using vanilla GCN as the baseline. PE denotes the learnable absolute positional embedding [42] added to the input pose sequence before the first layer of our network.

GC	BlockGC	PE	Encoding		Params	Acc(%)
			dynamic	static		
✓	-	-	-	-	2.1M	85.2
-	✓	-	-	-	1.2M (-0.9M)	85.8
-	✓	✓	-	-	1.2M	86.0
-	✓	✓	-	✓	1.2M	86.2
-	✓	✓	✓	-	1.3M (+0.1M)	86.7
-	✓	✓	✓	✓	1.3M	**86.9**

simultaneously improving the performance. The introduction of dynamic topological encoding marginally increases the parameter count but significantly bolsters performance by 0.7%. By integrating our BlockGC with both topological encodings, we outperform the baseline model by 1.7%, while concurrently reducing the parameters by approximately 38% as listed in Table VI.4.

Shared vs. Feature-wise Encodings: In comparison to a shared encoding for all feature dimensions, feature-wise encoding provides a larger capacity at the expense of an increase in parameters. For our static topological encoding, given the simplicity of the graph distance (discrete and one-dimensional), a shared encoding is adequate. Consequently, we simply employ a shared static topological encoding. In contrast, Euclidean distance is continuous and spans three dimensions, necessitating a larger capacity to retain such information. As demonstrated in Table VI.5, the effectiveness of shared dynamic topological encoding is restricted.

Table VI.5.: Feature-wise vs. shared Encoding.

Toplogical Encoding	Encoding Dimension		Acc(%)
	shared	feature-wise	
Dynamic	✓	-	86.5
	-	✓	86.9
Static	✓	-	86.9
	-	✓	86.7

Contrasting BlockGC with DecouplingGC: We pit our BlockGC against DecouplingGC [40] in Table VI.6. As analyzed in Table VI.1, DecouplingGC is essentially using an ensemble of adjacency matrices, which has theoretically larger complexity and more parameters than our BlockGC. Here, we validate the parameter reduction and performance improvement of BlockGC through experiments. Notably, the count of spatial weight parameters inversely correlates with the number of groups, while the number of adjacency matrices increases concurrently. As a result, our BlockGCs with varying groups possess a similar number of parameters. BlockGC significantly trims down the parameters compared to vanilla GC by almost half (43%), yet it still attains a substantial improvement against the baseline (approximately 0.6%). This result is noteworthy as it highlights the redundancy in the extensive parameters in the convolution weight matrix for feature projection and corroborates our analysis in Section 3.3 that the decoupling of features across different groups is a beneficial attribute.

Table VI.6.: BlockGC vs. DecouplingGC [40].

Layer	Groups	Parameters	Acc(%)
Vanilla GC (Baseline)	1	2.1M	85.2
DecouplingGC	4	2.1M	85.5
	8	2.2M	85.6
	16	2.3M	85.4
BlockGC (ours)	4	1.2M	85.7
	8	1.2M (-0.9M)	**85.8**
	16	1.2M	85.6

Numbers of Inserted Topological Encoding: We study the effectiveness of the number of layers that are integrated with our topological encoding, and the results are shown in Table VI.7. It shows that repetitively providing the topological information of the input pose sequence to the hidden layers benefits the classification performance.

Table VI.7.: Number of inserted layers.

Layers	Acc. (%)
0	86.0
1	86.4
5	86.7
10	86.9

Table VI.8.: Impacts of Barcodes' Vectorization Feature Size.

Dimension	Parameters	Acc. (%)
32	1.26M	86.5
64	1.32M	86.9
128	1.44M	86.4

Barcodes' Vectorization Feature Size: We evaluate the impacts for the dimension of the Barcode vectorization feature (namely d' as the output of $\Psi(\cdot)$) (see Table VI.8). The performance improves with increased dimension, and decreases again with the dimension of 128. Therefore, we adopt 64 as the size of the vectorization feature vector.

5. Discussion

Broader Impact. Skeleton-based action recognition is computationally more efficient compared to video-based action recognition and therefore finds its application in a broad range of real-world scenarios with limited resources. Additionally, skeleton data erases the identities of human subjects, such that there's a special advantage regarding privacy protection, e.g., for medical purposes and violent intent detection.

Part II.

Part Two: Probabilistic Scoring in Prediction

Accurate and reliable probabilistic scoring at the output layer is fundamental to both discriminative and generative models. In classification tasks, the widespread use of Softmax combined with cross-entropy loss has become a standard paradigm, providing a normalized predictive distribution over classes. While effective, this approach often leads to overconfident predictions and poor calibration. To mitigate overconfidence, label smoothing has been widely adopted by assigning soft targets. However, despite its popularity, label smoothing suffers from representation collapse and the reinforcement of erroneous predictions with unjustified confidence.

To address these challenges, we propose a novel regularization strategy designed to reduce overconfidence in erroneous predictions and preserve feature diversity, thereby improving classification performance and transferability.

Beyond classification, probabilistic scoring plays a crucial role in generative tasks such as open-ended text generation. Direct sampling from the Softmax distribution risks producing low-quality outputs, while deterministic decoding methods like greedy decoding or beam search often yield repetitive or overly predictable results. Balancing diversity and coherence in generated outputs remains a fundamental challenge.

In this context, the thesis presents a systematic framework for the fair evaluation of various adaptive sampling methods in large language models. This evaluation compares their inherent capacities to balance diversity and risk, independent of parameter tuning. Based on these insights, practical guidelines are provided to assist users in selecting suitable sampling methods and parameter settings, making these approaches more accessible and effective in practice.

Chapter VII.

Overcoming the Error-Enhancement Defect in Label Smoothing for Image Classifiers

This chapter contributes to the overarching goal of this thesis—advancing neural probabilistic scoring—by tackling a fundamental challenge in discriminative classification. Whereas earlier chapters focused on enhancing representation learning through improved attention mechanisms, here we investigate how training targets influence model confidence and the structure of learned features.

Label Smoothing (LS) is widely used to improve model calibration and accuracy. However, it can unintentionally encourage overconfidence in misclassified samples and lead to overly compressed intra-class features, limiting generalization and transferability.

To address these issues, we introduce Max Suppression (MaxSup), a novel regularization technique that retains the beneficial effects of LS while removing its error-enhancement component. MaxSup promotes richer intra-class variation and more robust feature representations, resulting in improved classification accuracy and downstream task performance.

By exposing and rectifying the hidden pitfalls of LS, this chapter provides a crucial advancement toward principled probabilistic scoring methods that enhance both predictive reliability and the quality of learned representations.

1. Introduction

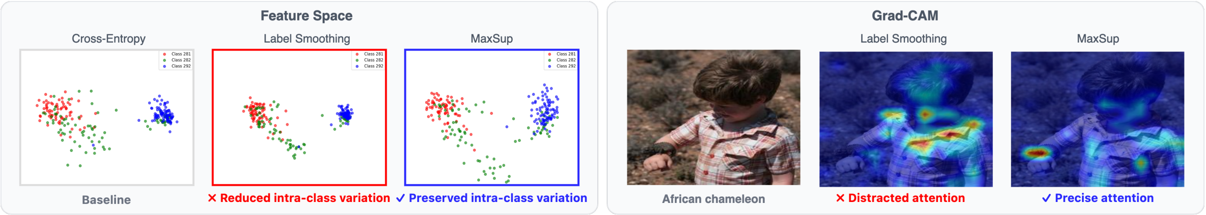


Figure VII.1.: Comparison of Label Smoothing (LS) and MaxSup. **Left:** MaxSup mitigates the intra-class compression induced by LS while preserving inter-class separability. **Right:** Grad-CAM visualizations show that MaxSup more effectively highlights class-discriminative regions than LS.

Multi-class classification [189, 132] conventionally relies on one-hot labels, implicitly treating each class as if it were completely orthogonal to every other. In reality, however, classes often share low-level attributes [260, 209] or exhibit high-level semantic similarities [36, 253, 166], which makes the strict orthogonality assumption overly simplistic. This mismatch can lead to *over-confident* classifiers, ultimately reducing generalization [91].

To address overconfidence, Szegedy et al. [216] introduced **Label Smoothing** (LS), blending a uniform distribution with the hard label to reduce the model's certainty in the target class. LS has become a mainstay in both image recognition [96, 221, 147, 282] and neural machine translation [80, 6], often improving accuracy and calibration [161]. However, studies have also revealed that LS can produce *overly tight clusters* in the feature space [127, 191, 245], thereby lowering intra-class diversity and harming transferability [72]. Meanwhile, Zhu et al. [284] reported that LS inadvertently increases confidence in *incorrect* predictions, though the exact cause remained unclear.

In this chapter, we show that LS's training objective inherently contains an *error-enhancement* term that amplifies misclassified predictions, thus causing overconfident errors and tighter feature clusters (Section 3.1, Table VII.1). Extending Zhu et al. [284], we define "overconfidence" in terms of the network's top-1 prediction rather than calibration-based criteria. Our analysis further demonstrates that penalizing the ground-truth logit in misclassifications compresses the feature space, reducing intra-class variation (Table VII.2), as corroborated by Grad-CAM visualizations (Fig. VII.3).

To overcome these limitations, we propose **Max Suppression (MaxSup)**, which retains LS's desirable *regularization* effect while eliminating its *error-enhancement* component. Rather than penalizing the ground-truth logit, MaxSup penalizes the *largest* logit, thus providing consistent regularization regardless of prediction correctness. By preventing ground-truth suppression during misclassification, MaxSup preserves richer intra-class variation and improves inter-class separability. As illustrated in Figure VII.1, this alleviates the compression and attentional shortcomings introduced by LS, leading to more robust feature representations. Extensive experiments on both image classification and semantic segmentation confirm that MaxSup not only alleviates intra-class collapse but also boosts final accuracy, enhances generalization, and strengthens transfer performance.

Our contributions are summarized as follows:

- We present a **logit-level analysis of Label Smoothing** that unearths an 'error-enhancement" term, revealing how LS inadvertently reinforces overconfidence in misclassified samples.
- We propose **Max Suppression (MaxSup)**, which preserves LS's desired regularization while removing its detrimental error-enhancement component, thereby reducing intra-class compression and boosting both classification and downstream task performance.

2. Related Work

We survey regularization techniques before focusing on Label Smoothing (LS) and highlighting how *MaxSup* differs.

2.1. Regularization

Regularization enhances the generalization of deep neural networks by limiting model complexity through various strategies. Classical approaches such as ℓ_2 [129] and ℓ_1 [287] regularization constrain large or sparse weights, respectively, while Dropout [213] randomly deactivates neurons to prevent feature co-adaptation. Among loss-based methods, Label

Smoothing (LS) [216] redistributes probability mass away from the ground-truth class, improving both accuracy and calibration [161]. Variants like Online Label Smoothing (OLS) [263] and Zipf Label Smoothing (Zipf-LS) [141] adapt LS by considering the model's evolving predictions, yet they still fail to address the fundamental issue that arises when the ground-truth logit is not maximal (Section 3.1, Table VII.1). Other loss-based regularizers, such as Confidence Penalty [173] and Logit Penalty [51], target different aspects of the output distribution. Confidence Penalty discourages overconfident predictions, whereas Logit Penalty minimizes the global ℓ_2-norm of logits to improve feature separability [127]. However, Logit Penalty can reduce intra-class variation, impairing transfer learning (Section 4).

Our MaxSup approach MaxSup diverges from these methods by selectively penalizing only the top-1 logit (z_{max}) rather than the ground-truth logit (z_{gt}). Unlike LS-based techniques, which can exacerbate errors by excessively shrinking z_{gt} for misclassified samples, MaxSup uniformly applies regularization to all predictions, regardless of correctness. Consequently, it effectively sidesteps the *error-enhancement* issue, preserves richer intra-class diversity (Table VII.2), and sustains robust transfer performance across various datasets and architectures (Table VII.3).

2.2. Studies on Label Smoothing

Label Smoothing (LS) has also been extensively examined in the context of knowledge distillation. Yuan et al. [257] showed that LS can act as a proxy for distillation, while Shen et al. [203] explored its role within teacher–student frameworks. Chandrasegaran et al. [29] further demonstrated that a low-temperature, LS-trained teacher can improve distillation performance. Meanwhile, Kornblith et al. [127] found that LS tightens class clusters in feature space, reducing transfer performance. From a Neural Collapse (NC) perspective [276, 90], LS drives the model toward rigid feature clusters, a phenomenon measured by Xu and Liu [245] via a variability metric.

Comparison with existing LS techniques Our primary objective is to mitigate the error-enhancement effect. Instead of refining a smoothed label, as in OLS or Zipf-LS, **MaxSup** directly penalizes the highest logit z_{max}. This simple yet effective modification ensures uniform regularization even when z_{gt} is not the top logit, thereby maintaining greater intra-class diversity and avoiding the performance degradation common to LS-based approaches (Section 3.2). Additionally, MaxSup integrates seamlessly with standard training pipelines, requiring no extra computational overhead beyond simply replacing LS.

3. Max Suppression Regularization (MaxSup)

We first partition the training objective into two components: the standard Cross-Entropy (CE) loss and a regularization term introduced by Label Smoothing (LS). By expressing LS in terms of logits (Theorem 3.3), we isolate two key factors: a *regularization term* that controls overconfidence and an *error-enhancement term* that enlarges the gap between the ground-truth logit z_{gt} and any higher logits (Corollary 3.4, Equation (VII.5)), ultimately degrading performance. To address these shortcomings, we propose **Max Suppression Regularization (MaxSup)**, which applies the penalty to the largest logit z_{max} rather than to z_{gt} (Equation (VII.8), Section 3.2). This shift delivers consistent regularization for both correct and incorrect predictions, preserves intra-class variation, and bolsters inter-class separability. Consequently, MaxSup mitigates the representation collapse found in LS, attains superior ImageNet-1K accuracy (Table VII.1), and improves transferability (Table VII.2, Table VII.3). The following sections elaborate on MaxSup's formulation and its integration into the overall training pipeline.

3.1. Revisiting Label Smoothing

Label Smoothing (LS) is a regularization technique designed to reduce overconfidence by softening the target distribution. Rather than assigning probability 1 to the ground-truth class and 0 to all others, LS redistributes a fraction α of the probability uniformly across all classes:

Definition 3.1. For a classification task with K classes, LS converts a one-hot label $\mathbf{y} \in \mathbb{R}^K$ into a soft label $\mathbf{s} \in \mathbb{R}^K$:

$$s_k = (1-\alpha)y_k + \frac{\alpha}{K}, \tag{VII.1}$$

where $y_k = \mathbb{1}_{\{k=gt\}}$ denotes the ground-truth class. The smoothing factor $\alpha \in [0,1]$ reduces the confidence assigned to the ground-truth class and distributes $\frac{\alpha}{K}$ to other classes uniformly, thereby mitigating overfitting, enhancing robustness, and promoting better generalization.

To clarify the effect of LS on model training, we first decompose the Cross-Entropy (CE) loss into a standard CE term and an additional LS-induced regularization term:

Lemma 3.2. ***Decomposition of Cross-Entropy Loss with Soft Labels.***

$$H(\mathbf{s},\mathbf{q}) \;=\; H(\mathbf{y},\mathbf{q}) \;+\; L_{LS}, \tag{VII.2}$$

where

$$L_{LS} \;=\; \alpha\biggl(H\Bigl(\tfrac{\mathbf{1}}{K},\mathbf{q}\Bigr) \;-\; H(\mathbf{y},\mathbf{q})\biggr). \tag{VII.3}$$

Here, $\mathbf{q}$ is the predicted probability vector, $H(\cdot)$ denotes the Cross-Entropy, and $\frac{\mathbf{1}}{K}$ is the uniform distribution introduced by LS. This shows that LS adds a regularization term, L_{LS}, which smooths the output distribution and helps to reduce overfitting. (See Appendix 1 for a formal proof.)

Building on Lemma 3.2, we next explicitly express L_{LS} at the logit level for further analysis.

Theorem 3.3. ***Logit-Level Formulation of Label Smoothing Loss.***

$$L_{LS} \;=\; \alpha\Bigl(z_{gt} \;-\; \frac{1}{K}\sum_{k=1}^{K} z_k\Bigr), \tag{VII.4}$$

where z_{gt} is the logit corresponding to the ground-truth class, and $\frac{1}{K}\sum_{k=1}^{K} z_k$ is the average logit. Thus, LS penalizes the gap between z_{gt} and the average logit, encouraging a more balanced output distribution and reducing overconfidence. (See Appendix 2 for the proof.)

The behavior of L_{LS} differs depending on whether z_{gt} is already the maximum logit. Specifically, depending on whether the prediction is correct ($z_{gt} = z_{max}$) or incorrect ($z_{gt} \neq z_{max}$), we can decompose L_{LS} into two parts:

Corollary 3.4. ***Decomposition of Label Smoothing Loss.***

$$L_{LS} = \underbrace{\frac{\alpha}{K} \sum_{z_m < z_{gt}} \Big(z_{gt} - z_m \Big)}_{Regularization} + \underbrace{\frac{\alpha}{K} \sum_{z_n > z_{gt}} \Big(z_{gt} - z_n \Big)}_{Error\text{-}Enhancement}, \tag{VII.5}$$

where M and N are the numbers of logits below and above z_{gt}, respectively ($M + N = K - 1$). Note that the error-enhancement term vanishes when $z_{gt} = z_{max}$.

(i) ***Regularization**: Penalizes the gap between z_{gt} and any smaller logits, thereby moderating overconfidence.*

(ii) ***Error-Enhancement**: Penalizes the gap between z_{gt} and larger logits, inadvertently increasing overconfidence in incorrect predictions.*

Although LS aims to combat overfitting by reducing prediction confidence, its error-enhancement component can be detrimental for misclassified samples, as it widens the gap between the ground-truth logit z_{gt} and the incorrect top logit. Concretely:

(i) **Correct Predictions** ($z_{gt} = z_{max}$): The error-enhancement term is zero, and the regularization term effectively reduces overconfidence by shrinking the gap between z_{gt} and any smaller logits.

(ii) **Incorrect Predictions** ($z_{gt} \neq z_{max}$): LS introduces two potential issues:

- **Error-Enhancement**: Increases the gap between z_{gt} and larger logits, reinforcing overconfidence in incorrect predictions.
- **Inconsistent Regularization**: The regularization term lowers z_{gt} yet does not penalize z_{max}, which further impairs learning.

These issues with LS on misclassified samples have also been observed in prior work [241]. By precisely identifying these two components (regularization vs. error-enhancement), we can design a more targeted solution.

Ablation Study on LS Components To gauge the influence of each LS component, we conduct an ablation study on ImageNet-1K using a DeiT-Small model [221] without Mixup or CutMix. As shown in Table VII.1, LS's gains arise solely from the regularization term, whereas the error-enhancement term degrades performance. In contrast, our proposed **Max Suppression Regularization (MaxSup)** omits the error-enhancement effect and achieves higher accuracy by retaining the beneficial regularization. From Table VII.1, it is evident that LS's overall accuracy boost is exclusively attributed to the regularization component, whereas error-enhancement consistently degrades performance (73.63% or 73.69%). Removing the error-enhancement term while keeping only the regularization improves accuracy slightly (75.98% vs. 75.91%). Finally, by avoiding error-enhancement entirely and preserving the helpful regularization, **MaxSup** achieves 76.12% accuracy—surpassing LS. This result underscores that MaxSup directly addresses LS's primary shortcoming by consistently applying the intended regularization even when the model's top-1 prediction is incorrect.

Table VII.1.: Ablation on LS components using DeiT-Small on ImageNet-1K (without CutMix or Mixup). "Regularization" denotes penalizing logits smaller than z_{gt}; "Error-Enhancement" penalizes logits larger than z_{gt}. MaxSup removes error-enhancement while retaining regularization.

Method	**Formulation**	**Accuracy**
Baseline	–	74.21
+ Label Smoothing	$\frac{\alpha}{K}\sum_{z_m<z_{gt}}(z_{gt}-z_m)$ $+\frac{\alpha}{K}\sum_{z_n>z_{gt}}(z_{gt}-z_n)$	75.91
+ Regularization	$\frac{\alpha}{M}\sum_{z_m<z_{gt}}(z_{gt}-z_m)$	75.98
+ Error-Enhancement	$\frac{\alpha}{N}\sum_{z_n>z_{gt}}(z_{gt}-z_n)$	73.63
+ Error-Enhancement	$\alpha\,(z_{gt}-z_{max})$	73.69
+ MaxSup	$\alpha\Big(z_{max}-\frac{1}{K}\sum_{k=1}^{K}z_k\Big)$	76.12

3.2. Max Suppression Regularization

Label Smoothing (LS) suffers from two main limitations: *inconsistent regularization* and *error amplification*. As discussed in Section 3.1 and illustrated in Table VII.1, LS penalizes the ground-truth logit z_{gt} even for misclassified examples, thereby unnecessarily widening the gap between z_{gt} and the incorrect top-1 logit. To address these critical shortcomings, we propose **Max Suppression Regularization (MaxSup)**, which explicitly penalizes the largest logit z_{max} rather than z_{gt}. This crucial shift ensures uniform regularization across both correct and misclassified samples, effectively eliminating the error-amplification issue seen in LS (Table VII.1), and preserving the integrity of the ground-truth logit for more stable and robust learning performance.

Definition 3.5. Max Suppression Regularization

We define the Cross-Entropy loss with MaxSup as follows:

$$\underbrace{H(\mathbf{s},\mathbf{q})}_{\text{CE with Soft Labels}} = \underbrace{H(\mathbf{y},\mathbf{q})}_{\text{CE with Hard Labels}} + \underbrace{L_{MaxSup}}_{\text{Max Suppression Loss}}, \tag{VII.6}$$

where

$$L_{MaxSup} = \alpha\Big(H\big(\tfrac{\mathbf{1}}{K},\mathbf{q}\big) - H(\mathbf{y}',\mathbf{q})\Big), \tag{VII.7}$$

and

$$y'_k = \mathbb{1}_{\big\{k=\arg\max(\mathbf{q})\big\}},$$

so that $y'_k = 1$ identifies the model's top-1 prediction and $y'_k = 0$ otherwise. Here, $H\big(\frac{\mathbf{1}}{K},\mathbf{q}\big)$ encourages a uniform output distribution to mitigate overconfidence, while $H(\mathbf{y}',\mathbf{q})$ penalizes the current top-1 logit. By shifting the penalty from z_{gt} (the ground-truth logit) to z_{max} (the highest logit), MaxSup avoids unduly suppressing z_{gt} when the model misclassifies, thus overcoming Label Smoothing's principal shortcoming.

Logit-Level Formulation of MaxSup Building on the logit-level perspective introduced for LS in Section 3.1, we can express L_{MaxSup} as:

$$L_{MaxSup} = \alpha\left(z_{max} - \frac{1}{K}\sum_{k=1}^{K} z_k\right), \tag{VII.8}$$

where $z_{max} = \max_k\{z_k\}$ is the largest (top-1) logit, and $\frac{1}{K}\sum_{k=1}^{K} z_k$ is the mean logit. Unlike LS, which penalizes the ground-truth logit z_{gt} and may worsen errors in misclassified samples, MaxSup shifts the highest logit uniformly, thus providing consistent regularization for both correct and incorrect predictions. As shown in Table VII.1, this approach eliminates LS's error-amplification issue while preserving the intended overconfidence suppression.

Comparison with Label Smoothing MaxSup fundamentally differs from LS in handling correct and incorrect predictions. When $z_{gt} = z_{max}$, both LS and MaxSup similarly reduce overconfidence. However, when $z_{gt} \neq z_{max}$, LS continues to shrink z_{gt}, widening the gap with the incorrect logit, whereas MaxSup penalizes z_{max}, preserving z_{gt} from undue suppression. As illustrated in Figure VII.3, this allows the model to recover from mistakes more effectively and avoid reinforcing incorrect predictions.

Gradient Analysis To understand MaxSup's optimization dynamics, we compute its gradients with respect to each logit z_k. Specifically,

$$\frac{\partial L_{MaxSup}}{\partial z_k} = \begin{cases} \alpha\left(1 - \frac{1}{K}\right), & \text{if } k = \arg\max(\mathbf{q}), \\ -\frac{\alpha}{K}, & \text{otherwise.} \end{cases} \tag{VII.9}$$

Thus, the top-1 logit z_{max} is reduced by $\alpha\left(1 - \frac{1}{K}\right)$, while all other logits increase slightly by $\frac{\alpha}{K}$. In misclassified cases, the ground-truth logit z_{gt} is therefore spared from penalization, thereby avoiding the error-amplification issue seen in LS. For completeness, Appendix A provides a full derivation of these gradients, and Figure VII.2 compares the resulting logit distributions under different regularizers.

Behavior Across Different Samples MaxSup applies a dynamic penalty that depends on the model's current predictions. For high-confidence, correctly classified examples, it behaves similarly to LS by reducing overconfidence, thus effectively mitigating overfitting. In contrast, for misclassified or uncertain samples, MaxSup specifically suppresses the incorrect top-1 logit, further safeguarding the ground-truth logit z_{gt}. This selective strategy preserves an accurate representation of the true class while actively discouraging the propagation of errors. As shown in Section 5.1 and Table VII.5, this promotes more robust decision boundaries and ultimately leads to stronger generalization performance.

Theoretical Insights and Practical Benefits MaxSup provides both theoretical and practical advantages compared to LS. Whereas LS applies a uniform penalty to the ground-truth logit regardless of correctness, MaxSup focuses on penalizing only the most confident logit z_{max}. This dynamic adjustment prevents error accumulation in misclassifications, thereby ensuring more stable convergence. As a result, MaxSup achieves stronger generalization, exhibits greater robustness to label noise, and performs well on

Table VII.2.: Metrics for feature representation quality using ResNet-50 trained on ImageNet-1K. We report *intra-class variation* ($\bar{d}_{\text{within}}$) and *inter-class separability* (R^2), both of which benefit from higher values. Although all methods reduce $\bar{d}_{\text{within}}$ relative to the baseline, MaxSup preserves the most within-class diversity.

Method	$\bar{d}_{\text{within}} \uparrow$		$R^2 \uparrow$	
	Train	**Val**	**Train**	**Val**
Baseline	0.3114	0.3313	0.4025	0.4451
LS	0.2632	0.2543	0.4690	0.4611
OLS	0.2707	0.2820	0.5943	0.5708
Zipf's	0.2611	0.2932	0.5522	0.4790
MaxSup	**0.2926**	<u>0.2998</u>	0.5188	0.4972
Logit Penalty	<u>0.2840</u>	**0.3144**	0.6448	0.6024

challenging datasets. Moreover, as shown in Section 4, MaxSup preserves higher intra-class diversity, which substantially improves transfer learning performance (Table VII.3) and yields more interpretable activation maps (Figure VII.3).

4. Analysis of MaxSup's Learning Benefits

MaxSup simultaneously promotes **inter-class separability** and **intra-class variation**, both essential for robust classification and effective feature transfer. In this section, we explore how MaxSup achieves these objectives and contrast its effectiveness with alternative regularization methods.

4.1. Intra-Class Variation and Transferability

As noted in Section 3.1, **Label Smoothing (LS)** primarily restricts overconfidence when the ground-truth class is correctly predicted, inadvertently causing *error enhancement* for misclassified samples. This selective penalty can overly compress intra-class diversity. In contrast, **MaxSup** uniformly penalizes the top-1 logit in both correct and incorrect cases, eliminating LS's error-enhancement component and thus preserving more fine-grained distinctions within each class. Table VII.2 compares *intra-class variation* $\bar{d}_{\text{within}}$ and *inter-class separability* R^2 [127] for a ResNet-50 model trained on ImageNet-1K. Although all regularization strategies reduce $\bar{d}_{\text{within}}$ relative to the baseline, MaxSup shows the smallest reduction, implying stronger retention of within-class variability—often correlated with improved generalization and transferability. The benefits of this richer intra-class structure appear clearly in Table VII.3, where linear transfer performance on CIFAR-10 is reported. Although LS and Logit Penalty improve ImageNet accuracy, they diminish transfer accuracy by over-suppressing informative features. In contrast, MaxSup preserves transfer performance near that of the baseline, suggesting it retains crucial, discriminative features that generalize effectively to downstream tasks.

Table VII.3.: Validation performance on CIFAR-10 with a linear probe using l_2-regularized multinomial logistic regression. Although Label Smoothing and Logit Penalty improve ImageNet accuracy, they substantially degrade transfer accuracy compared to MaxSup.

Method	**Linear Transfer Acc.**
Baseline	0.8143
Label Smoothing	0.7458
Logit Penalty [51]	0.7242
MaxSup	**0.8102**

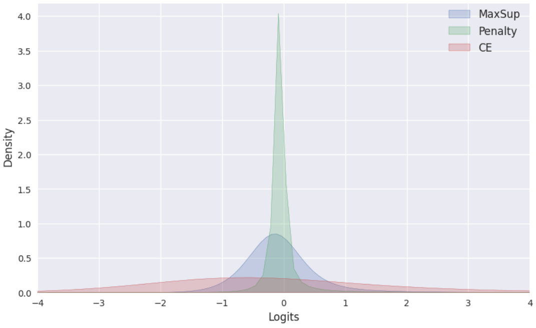


Figure VII.2.: Logit density plots under three different regularization strategies: MaxSup, Logit Penalty, and standard Cross Entropy. Logit Penalty induces a narrower logit distribution, reflecting excessive shrinkage that reduces intra-class variation. By contrast, MaxSup preserves a broader range of logits and thus richer representations.

4.2. Impact of Logit Regularization

Different regularization methods impose distinct constraints on the logit space, thereby shaping the model's representational capacity [127]. Among these approaches, **Logit Penalty** and **MaxSup** both act directly on logits but differ fundamentally in how they apply regularization. Logit Penalty operates by minimizing the ℓ_2-norm of the entire logit vector, causing a global reduction in logit magnitudes that often induces sparsity. This uniform shrinkage can limit intra-class variation, thereby weakening the model's ability to transfer features to downstream tasks. In contrast, MaxSup targets only the largest (top-1) logit, nudging it closer to the average logit. By selectively penalizing only the most confident prediction, MaxSup avoids universal shrinkage and preserves richer intra-class diversity, a property crucial for effective transferability. Figure VII.2 illustrates the distribution of logits under various regularizers. Logit Penalty yields a narrower logit range, reflecting excessive sparsity and aligning with its lower transfer performance (Table VII.3). By comparison, MaxSup maintains broader logit distributions, thereby retaining the fine-grained feature distinctions needed to excel on downstream tasks.

5. Experiments

5.1. Evaluation on ImageNet Classification

In this section, we assess the effectiveness of MaxSup on ImageNet-1K, comparing its performance against standard Label Smoothing and related variants.

Experiment Setup

Model Training Configurations We conduct extensive experiments with both CNN and Transformer models, including the ResNet family [96], MobileNetV2 [190], and DeiT-Small [221], all thoroughly evaluated on the large-scale ImageNet dataset [128] for comprehensive performance analysis.

For **ResNet Series** models, we train for 200 epochs using stochastic gradient descent (SGD) with momentum 0.9, a weight decay of 1×10^{-4}, and a batch size of 2048. The initial learning rate is set to 0.85 and scheduled via cosine annealing.[1] We also evaluate ResNet-based CNNs on CIFAR-100. Here, we use an initial learning rate of 0.1, reducing it by a factor of 5 at the 60th, 120th, and 160th epochs. We train for 200 epochs with a batch size of 128, weight decay of 5×10^{-4}, and Nesterov momentum set to 0.9. For **DeiT-Small**, we employ the official implementation and train from scratch without knowledge distillation. Although the original DeiT paper emphasizes distillation, we exclude it to provide a clearer, unbiased assessment of MaxSup's contributions. We also omit CutMix and Mixup to retain the same optimization objective.

Hyperparameters for Compared Methods We compare Max Suppression Regularization against multiple Label Smoothing variants, including Zipf Label Smoothing [141] and Online Label Smoothing [263]. When official implementations are available, we use them directly; otherwise, we follow the respective papers' descriptions closely to ensure fair comparisons. All training hyperparameters are kept identical to those of the baseline models, except for algorithm-specific settings and necessary adjustments. Additionally, we employ a linearly increasing α scheduler, which generally benefits training and stability; see Appendix 6 for details. This scheduler is applied to both MaxSup and standard Label Smoothing by default to maintain consistency.

Experiment Results

ConvNet Comparison Table VII.4 summarizes the performance of MaxSup alongside various smoothing and self-distillation methods on both ImageNet and CIFAR-100. Across all tested convolutional architectures, MaxSup achieves the highest accuracy among label smoothing–based regularizers. By contrast, OLS [263] and Zipf-LS [141] yield less consistent gains, suggesting their reported empirical benefits may depend heavily on specific training schedules. In our reproductions of OLS and Zipf-LS, we follow the authors' original codebases and method-specific hyperparameters but do not adopt their complete training recipes. For example, the OLS paper uses a step learning-rate scheduler over 250 epochs with an initial rate of 0.1, while Zipf-LS trains for 100 epochs under a separate set of hyperparameters. Our results underscore the robustness of MaxSup across different training setups, in contrast to the more scheme-dependent improvements noted for OLS and Zipf-LS.

DeiT Comparison Table VII.5 compares various regularization techniques for DeiT-Small on ImageNet. MaxSup achieves an accuracy of 76.49%, surpassing Label Smoothing by 0.41 percentage points. Label Smoothing variants such as Zipf's and OLS yield only marginally higher or comparable performance relative to standard LS, suggesting that

[1] Additional training hyperparameters follow the FFCV training scripts in `https://github.com/libffcv/ffcv`. See Appendix 5 for further details on training setups.

Table VII.4.: Comparison of classic convolutional neural networks on ImageNet and CIFAR-100. Results are reported as "mean ± std" (percentage). **Bold** entries highlight the best performance; underlined entries mark the second best. (Methods with * denote code adaptations from official repositories; see text for details.)

Method	**ImageNet**				**CIFAR-100**			
	ResNet-18	**ResNet-50**	**ResNet-101**	**MobileNetV2**	**ResNet-18**	**ResNet-50**	**ResNet-101**	**MobileNetV2**
Baseline	69.09±0.12	76.41±0.10	75.96±0.18	71.40±0.12	76.16±0.18	78.69±0.16	79.11±0.21	68.06±0.06
Label Smoothing	69.54±0.15	76.91±0.11	77.37±0.15	71.61±0.09	77.05±0.17	78.88±0.13	79.19±0.25	69.65±0.08
Zipf-LS*	69.31±0.12	76.73±0.17	76.91±0.11	71.16±0.15	76.21±0.12	78.75±0.21	79.15±0.18	69.39±0.08
OLS*	69.45±0.15	77.23±0.21	77.71±0.17	71.63±0.11	77.33±0.15	78.79±0.12	79.25±0.15	68.91±0.11
MaxSup	**69.96±0.13**	**77.69±0.07**	**78.18±0.12**	**72.08±0.17**	**77.82±0.15**	**79.15±0.13**	**79.41±0.19**	**69.88±0.07**
Logit Penalty	68.48±0.10	76.73±0.10	77.20±0.15	71.13±0.10	76.41±0.15	78.90±0.16	78.89±0.21	69.46±0.08

Table VII.5.: Accuracy (%) comparison on DeiT-Small [221] using different Label Smoothing variants. Results are reported as "mean ± std"; parentheses indicate absolute improvement over the baseline.

Method	**Accuracy (Mean)**	**Std**
Baseline	74.39	0.19
Label Smoothing	76.08 (+1.69)	0.16
Zipf-LS	75.89 (+1.50)	0.26
OLS	76.16 (+1.77)	0.18
MaxSup	**76.49** (+2.10)	**0.12**

these approaches may be less effective for vision transformer architectures—potentially due to their reliance on extensive data augmentation schemes. In contrast, MaxSup consistently outperforms both standard LS and its variants, indicating its stronger ability to enhance feature representations without additional data manipulations. These findings underscore MaxSup's robustness across distinct model architectures, especially in settings where other regularization methods show limited effectiveness.

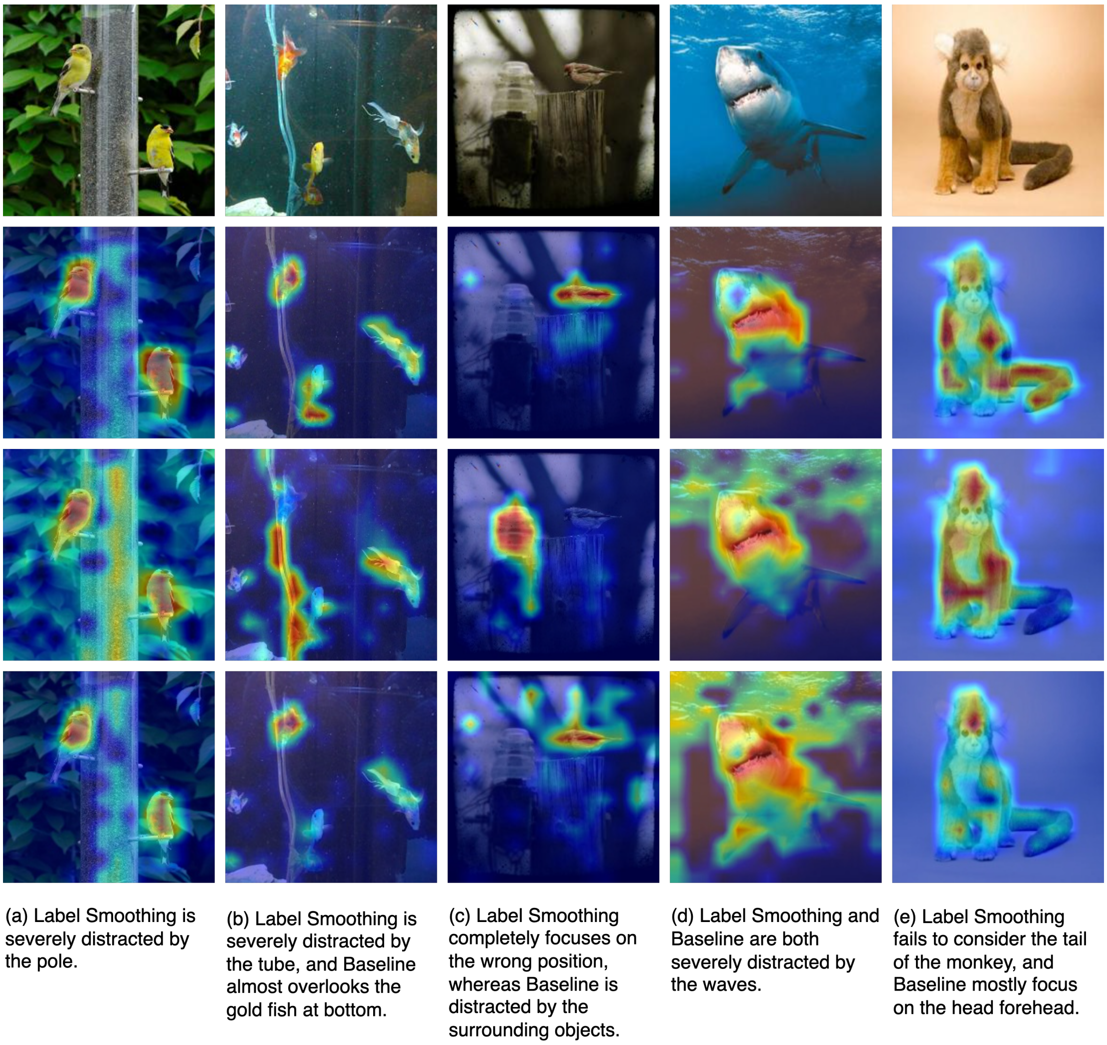
Table VII.6.: Semantic segmentation results on the ADE20K validation set. Models are pretrained on ImageNet-1K and then fine-tuned with UperNet [244] for enhanced performance. We report mean Intersection over Union (mIoU) under multi-scale (MS) testing for comprehensive evaluation.

Backbone	**Method**	**mIoU (MS)**
DeiT-Small [221]	Baseline	42.1
	Label Smoothing	42.4 (+0.3)
	MaxSup	**42.8** (+0.7)

5.2. Evaluation on Semantic Segmentation

To further assess the transferability of MaxSup to downstream tasks, we evaluate its performance on **semantic segmentation** using the MMSegmentation framework.[2] Specifically, we employ the UperNet architecture [244] with a DeiT-Small backbone, trained on ADE20K. We compare backbones trained with Label Smoothing and MaxSup (on ImageNet-1K) against a baseline, following the same setup as in Section 5.1. During fine-tuning, all

[2] https://github.com/open-mmlab/mmsegmentation



(a) Label Smoothing is severely distracted by the pole.

(b) Label Smoothing is severely distracted by the tube, and Baseline almost overlooks the gold fish at bottom.

(c) Label Smoothing completely focuses on the wrong position, whereas Baseline is distracted by the surrounding objects.

(d) Label Smoothing and Baseline are both severely distracted by the waves.

(e) Label Smoothing fails to consider the tail of the monkey, and Baseline mostly focus on the head forehead.

Figure VII.3.: Class activation maps generated by Grad-CAM [196] for DeiT-Small models trained with MaxSup (2nd row), Label Smoothing (3rd row), and a standard Baseline (4th row). The first row shows the original images. Compared with Label Smoothing, MaxSup more effectively suppresses distractions from non-target objects and preserves key features of the target class, thereby reducing instances in which the model partially or completely focuses on irrelevant regions.

models use a standard cross-entropy loss. As shown in Table VII.6, MaxSup achieves a mean Intersection over Union (mIoU) of 42.8%, surpassing the 42.4% obtained with Label Smoothing. These findings further highlight the improved feature representations afforded by MaxSup in downstream tasks such as semantic segmentation.

5.3. Visualization via Class Activation Maps

To assess how MaxSup influences model decision-making relative to Label Smoothing (LS), we employ Gradient-weighted Class Activation Mapping (Grad-CAM) [196]. Grad-CAM produces class-discriminative localization maps that highlight the regions most relevant to each classification decision.

We conduct comprehensive experiments on the DeiT-Small model under three distinct training setups: MaxSup (second row), Label Smoothing (third row), and standard Cross-Entropy (CE) as a baseline (fourth row). As illustrated in Figure VII.3, **MaxSup-trained models** exhibit a distinct advantage in effectively mitigating distractions caused by non-target salient objects in the background (e.g., a pole in the 'Bird' image, a tube in the ‘Goldfish’ image, and a cap in the ‘House Finch’ image). By contrast, LS-trained models often lose focus or erroneously attend to background objects, further reflecting the detrimental influence of LS’s Error-Enhancement term and its impact on feature learning (Please see Figures VII.3).

Moreover, **MaxSup** preserves a wider range of relevant object features, as evidenced in the ‘Shark’ and ‘Monkey’ examples, where LS-trained models fail to capture key details (fins and tails). These observations align with the analysis in Appendix 7, underscoring that MaxSup better retains rich intra-class information. Consequently, MaxSup-trained models yield more accurate and reliable classifications by leveraging these detailed feature representations.

6. Conclusion

In this chapter, we carefully examined the root causes of Label Smoothing’s (LS) shortcomings and introduced **Max Suppression Regularization (MaxSup)** as a targeted remedy. Our analysis shows that LS can unintentionally promote overconfidence in misclassified samples by applying insufficient regularization to erroneous top-1 logits. In contrast, MaxSup effectively addresses this by consistently penalizing the most confident logit, regardless of prediction correctness. Through extensive experiments and in-depth analyses, we demonstrate that MaxSup not only improves accuracy but also preserves richer intra-class variation and significantly enhances inter-class separability. Consequently, models trained with MaxSup capture finer-grained information about individual samples, ultimately leading to stronger transfer learning capabilities. Class activation maps further reveal that MaxSup directs model attention more accurately toward salient parts of target objects, effectively mitigating distractions from background elements.

Limitations & Future Work

Although our findings validate MaxSup’s effectiveness, several directions merit further investigation. Prior work [161] shows that teachers trained with LS can degrade performance in Knowledge Distillation [105], and Guo et al. [90] suggests LS accelerates convergence via conditioning number analysis. Future research could explore MaxSup’s impact on Knowledge Distillation workflows and its influence on training convergence. Additionally, recent studies [214, 81] indicate that ℓ_2 regularization biases final layer features and weights toward lower-rank solutions than those typically associated with neural collapse. Investigating how MaxSup interacts with these low-rank biases and whether it leads to similarly optimal or novel solutions is another intriguing avenue for future work.

Chapter VIII.

Balancing Diversity and Risk in Sampling-Based Decoding for Large Language Models

This chapter advances the thesis's investigation into neural probabilistic scoring by shifting focus to the output generation stage of large language models. While the preceding chapter examined how training objectives affect confidence calibration and feature structure in classification, this chapter tackles the challenge of balancing diversity and coherence in generative models.

Decoding strategies for language generation often face a trade-off: increasing diversity can lead to incoherence, while enforcing coherence tends to reduce creativity. Moreover, tuning sampling parameters is computationally expensive and sensitive, making reliable evaluation difficult. To address these issues, we introduce a novel evaluation framework based on a context-aware prefix tree that enables robust assessment of sampling methods' adaptability to the true data distribution, providing insights that are less sensitive to hyperparameter choices.

This framework facilitates principled comparison of different sampling techniques and reveals their practical strengths and limitations. By bridging probabilistic scoring methods from discriminative classifiers to generative language models, this chapter contributes critical insights and guidelines for reliable and diverse text generation.

1. Introduction

Large Language Models (LLMs) [2, 223, 118, 217] have demonstrated exceptional performance across a variety of applications, and the reliability of decoding strategies has become a critical concern. Previous works have revealed that likelihood-maximization such as beam search [70, 108, 234, 155] produces degenerate text which contains repetitive loops and incoherent context, particularly in open-ended tasks. Therefore, sampling-based decoding strategies, e.g., Top-p [108] and Top-k sampling [180, 70], have been widely adopted. The balance between diversity and quality of the generated text could be adjusted by tuning the temperature and truncation position to some extend, but requires non-trivial trial and error.

Recent studies [14, 285, 104, 154] proposed adaptive tail truncation mechanisms based on different criteria or assumptions, which maintain an allowed set of tokens with a flexible size according to the given prefix. To validate the effectiveness of a sampling method, they are often compared through extrinsic evaluation based on open-ended text generation

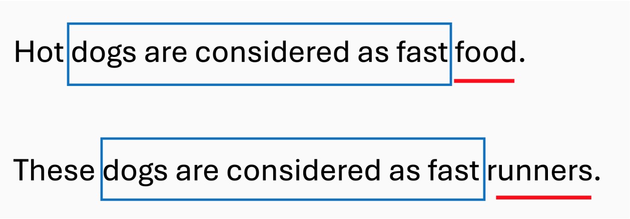

Hot dogs are considered as fast food.
These dogs are considered as fast runners.

Figure VIII.1.: N-gram models tend to overestimate the data support size given a prefix (marked by a red line) due to limited window size (marked with a blue window).

applications. For example, story generation [70] and document continuation [156]. Various metrics [234, 154, 175, 79] have been adopted to consider different aspects of the generated text.

We reveal two underlying issues in the current evaluation, which hinder the assessment of a method's significance in real-world applications:

- **The improvement of one method over another might be simply due to a better tuned parameter for the targeted task**: the performance of sampling methods is sensitive to their parameters, and parameter sweep is often operated on a extremely sparse grid due to the high computation cost. This is especially problematic considering the non-linear dependency between performance and parameters.
- **Users are agnostic to the optimal parameters in real-world applications**: Practically speaking, users often pick parameters based on their own need for the compromise between diversity and quality, after a few tryouts. There exists no universal optimal paramters in different scenarios and users are agnostic to the optimal parameters for their own tasks.

The above issues exactly indicate the need for an evaluation that allows for estimating the theoretical capacity of a truncation sampling method (how well it adapts to the variation in data supports given different prefixes), independent of hyperparamter tuning. Moreover, the second issue additionally highlights the need to identify the sweet spots of existing sampling methods, which could serve as a user guideline for practitioners.

In light of the above analysis, we propose a systematic way to assess the inherent adaptability of a sampling method. First, we rearrange Wikipedia-English [1] data into a word-level prefix tree, known as a Trie [74, 83]. It is noteworthy that a n-gram Trie [121] tends to overestimate the data support size given a prefix [15], as shown in Figure VIII.1. In a similar spirit to [60], we construct the prefix tree with only sentence-starting n-grams to preserve full sentence context, called Context-Preserving Trie (CP-Trie).

Given the CP-Trie, we are able to estimate the theoretical capacity of a sampling method, by examining the amount of tokens within and out of the data support with varying truncation parameter values. As shown in Figure VIII.2, the truncation positions, which exactly cover the full data supports, vary drastically given different prefixes and Top-k sampling could be regarded as a baseline method with zero adaptability. Therefore, an adaptive truncation method is supposed to better follow such a variation, so that improved diversity can be achieved without harming the quality.

In summary, our contributions are as follows:

[1] https://dumps.wikimedia.org/

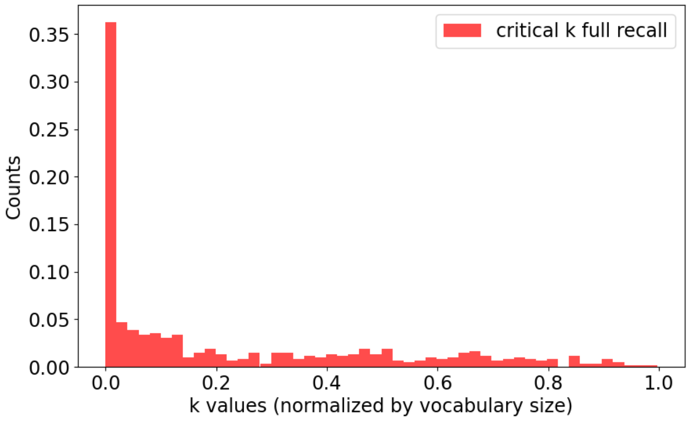


Figure VIII.2.: Histogram of the estimated optimal truncation values for gpt2-xl, which achieve exactly full recall of data support given different prefixes.

- We establish an intrinsic evaluation benchmark based on the collected CP-Trie, which allows for estimating the theoretical capacity of different sampling methods via thoroughly designed diversity and stability metrics.
- We conduct a comprehensive comparison of existing sampling approaches, which serves as a guideline for choosing a method and its parameter in real-world applications.
- We reveal that sampling-based decoding methods are underestimated in the existing study [204] due to the difficulty in parameter selection, highlighting the merit of our evaluation protocol.

2. Related Work

In this section, we summarize recent sampling decoding strategies, along with common benchmarks and metrics for open-ended text generation.

2.1. Sampling-based Decoding Methods

Vanilla sampling suffers from the risk of obtaining incoherent tokens; thus, truncation of the tail distribution has been heavily discussed, e.g., Top-k [180, 70] and Top-p sampling [108]. However, a fixed k or p is problematic when considering the high dynamic range of next reasonable tokens, as pointed out in more recent studies on adaptive sampling methods: Mirostat [14] is proposed based on Zipf statistics and the assumption of a steady perplexity during generation. Hewitt, Manning, and Liang [104] introduce η-sampling which dismisses the tokens with low probabilities in the tail of the predicted distribution based on absolute and relative thresholds. Locally Typical Sampling [154] assumes that the generated text should retain a similar entropy rate to that of human-generated text. Adaptive Decoding [285] proposes to keep the entropy of the truncated distribution close to the original entropy. Although these approaches have been demonstrated to be effective, their performance is highly dependent on the curated truncation parameters and the limited exemplar text.

2.2. Evaluation of Sampling-based Decoding

Common benchmarks include story generation with WritingPrompts dataset [70], document continuation with WikiText-103 dataset [156] and abstractive summarization on the CNN/DAILYMAIL dataset [162]. These benchmarks suffer from the problem of limited exemplar text, which fails to capture the diverse nature of human language.
Statistical metrics are mostly based on n-gram statistics and focus on a single aspect, such as Repetition [234], Diversity [154], Semantic coherence [79], Zipf's coefficient [108] (Unigram rank-frequency) and Self-BLEU [286].
Exemplar-based metrics dominate the evaluation of sampling-based decoding methods. As observed by Fan, Lewis, and Dauphin [70] and Holtzman et al. [108], lower perplexity of the generated text does not necessarily indicate better quality. And Holtzman et al. [108] suggested that the perplexity of the generated text should be close to that of the human text. MAUVE [175] takes the trade-off between precision and recall into account, by comparing the learnt distribution from a text generation model to the distribution of human-written text using divergence frontiers. Shi et al. [204] provides a comprehensive evaluation on a large collection of tasks, mostly relying on exemplar-based metrics. However, we reveal that such evaluation is affected by the biases in the curated parameters and limited exemplar text, and our evaluation method is shown to alleviate such an issue.

3. Revisiting Truncation Sampling

We begin by revisiting the formulation of truncation sampling, followed by identifying the unresolved challenges in evaluating truncation sampling methods.

3.1. Problem Formulation

Definition 3.1.

$$P_{trunc}(x_t|\boldsymbol{x}_{<t}) = \begin{cases} P_\theta(x_t|\boldsymbol{x}_{<t})/Z_{\boldsymbol{x}_{<t}} & x \in \mathcal{A}_{\boldsymbol{x}_{<t}} \\ 0 & \text{o.w.}, \end{cases} \tag{VIII.1}$$

where $\mathcal{A}_{\boldsymbol{x}_{<t}} \in \mathcal{V}$ denotes the allowed set of candidate next tokens at the t^{th} position, given a sequence of tokens $\boldsymbol{x}_{<t} = \{x_0, ..., x_{t-1}\}$ as prefix. $Z_{\boldsymbol{x}_{<t}} = \sum_{x \in \mathcal{A}_{\boldsymbol{x}_{<t}}} P_\theta(x_t|\boldsymbol{x}_{<t})$ is the renormalization term.

Given the Context-Preserving Trie of a reference dataset, we can compute the estimate of the optimal allowed set as follows :

Definition 3.2. Let $\mathcal{A}_{\boldsymbol{x}_{<t,\theta}}$ be the allowed set after truncation given the prefix $\boldsymbol{x}_{<t}$. The **approximated optimal allowed set** $\mathcal{A}^*_{\boldsymbol{x}_{<t}}$ corresponds to the allowed set with the minimum size, while covering the full data support for the t^{th} token $\mathcal{D}_{\boldsymbol{x}_{<t}}$ based on the Trie. It is the solution to the following objective function:

$$\begin{aligned} \mathcal{A}^*_{\boldsymbol{x}_{<t}} &= \min_\theta |\mathcal{A}_{\boldsymbol{x}_{<t,\theta}}| \\ \text{s.t.} \quad \mathcal{D}_{\boldsymbol{x}_{<t}} &\subseteq \mathcal{A}_{\boldsymbol{x}_{<t,\theta}}. \end{aligned} \tag{VIII.2}$$

Note that the above definition is designed to exclude the risk of obtaining OOD tokens before the cutoff [73], because such type of risk is unsolvable by truncation and is rather determined by the capacity of the trained LLMs. However, such risk is less severe compared to that introduced by inappropriate truncation, since LLMs exhibit a significant capability in predicting the next token [223, 2, 118, 217] and most OOD samples reside in the tail distribution.

3.2. Remaining Issues

We reveal three major issues in the evaluation of truncation sampling. We first summarize the problem of directly using probability as quality metric, then show that the choice of truncation parameter has a significant impact on the evaluation.
Unreliable Probability The probabilities of both the predicted and empirical distribution are not reliable for reflecting the quality of a text.

- Higher likelihood does not necessarily imply higher quality of the generated text [70, 108, 163, 233].
- Word frequencies are average statistics across various topics, and the optimal probabilities or ranking of each next token is ill-posed.
- Empirical distribution suffers from the sparsity issue [200, 136, 121] of the N-gram models.

Parameter Sensitivity We highlight the complexity and biases in parameter selection: Top-k and Top-p have constant upper bounds, i.e., the vocabulary size $|\mathcal{V}|$ and 1, respectively. In contrast, the upper bounds of η-sampling and adaptive sampling are dependent on LLM's predicted distribution, because they truncate the tail distribution based on the likelihood of tokens and the slope of Min-Max scaled entropy, respectively. The importance of identifying the effective ranges of such parameters is also reflected in the authors' choice of numeral digit for their parameters. For example, ΔConf is set to 0.0005 in Zhu et al. [285] and ϵ is chosen from 0.0001, 0.0009 and etc in Hewitt, Manning, and Liang [104]. In comparison, the adopted p values for Top-p sampling are merely two digits after zero, such as 0.95. This shows the significance of identifying the sweet spots of different sampling methods.

4. Method

In this section, we derive our metrics for evaluating different sampling-based decoding strategies. The metrics are carefully designed to address the issues discussed in Section 3.2.

4.1. Probability-Independent Metrics

To circumvent the **unreliable probability** issue, we merely check whether the predicted next token is in or out of the data support. Specifically, we define **Recall** and **Risk** to quantify diversity and quality of a sampling method on a single node of CP-Trie:

Definition 4.1.

$$\text{Recall}_{\theta,t} = \text{Minimum}\left(\frac{|\mathcal{A}_{\boldsymbol{x}_{<t},\theta}|}{|\mathcal{A}^*_{\boldsymbol{x}_{<t}}|}, 1\right) \tag{VIII.3}$$

$$\text{Risk}_{\theta,t} = \text{Maximum}\left(\frac{|\mathcal{A}_{\boldsymbol{x}_{<t},\theta}|}{|\mathcal{A}^*_{\boldsymbol{x}_{<t}}|} - 1, 0\right) \tag{VIII.4}$$

$\mathcal{A}_{\boldsymbol{x}_{<t},\theta}$ is dependent on the parameter selection for truncation, e.g., k value in Top-k sampling. When the allowed set is smaller than the approximated optimal allowed set after truncation, Recall is smaller than one and Risk is regarded as zero. With further increased size of the allowed set, Recall reaches one but Risk emerges. Since the sizes of reasonable sets vary drastically for different prefixes, it is not possible to always retain the approximated optimal allowed set with a predefined parameter. In this case, we reveal that the adaptability w.r.t. the varying size of data support of a sampling method indeed determines its effectiveness in real-world application.

More importantly, our evaluation does not rely on the empirical probability, which is biased and inaccurate due to limited dataset size or context window size. However, the tokens which appear in the dataset could be confidently regarded as reasonable, regardless of their actual probabilities. In addition, considering that temperature could change the flatness of distribution arbitrarily, we adopt ratio of token counts instead of probability mass to make the evaluation independent of temperature tuning and exemplar text. For a detailed discussion with supporting examples, please refer to Appendix 2.

4.2. Tuning-Independent Evaluation

To eliminate the huge impact of **Parameter Sensitivity** issue on fair evaluation, we adopt **Average Recall (AR)** at an average Risk and **Risk Standard Error (RSE)** at an average Risk to quantify **diversity** and **stability** of a sampling method across N nodes of CP-Trie, respectively:

Definition 4.2.

$$\begin{aligned}
&\mathrm{AR}_{\mathrm{Risk}-0.1} = \frac{1}{N}\sum_{i=1}^{N}\mathrm{Recall}_{\theta,t}^{(i)} \\
&\mathrm{RSE}_{\mathrm{Risk}-0.1} = \tfrac{1}{N}\sqrt{\textstyle\sum_{i=1}^{N}(\mathrm{Risk}_{\theta,t}^{(i)} - \tfrac{1}{N}\sum_{i=1}^{N}\mathrm{Risk}_{\theta,t}^{(i)})^2} \\
&\text{s.t.} \quad \frac{1}{N}\sum_{i=1}^{N}\mathrm{Risk}_{\theta,t}^{(i)} = 0.1,
\end{aligned} \tag{VIII.5}$$

where the superscript (i) denotes the i^{th} node in the evaluation set of nodes on the prefix tree. Analogously, a family of critical values such as $\mathrm{AR}_{\mathrm{Risk}-0.5}$ can be easily defined.

Since θ is now determined by the given average Risk, the diversity metric reflects the genuine capacity of a sampling method regardless of parameter tuning. This allows for a fair comparison of different sampling methods, especially considering their drastically different effective ranges, as mentioned in Section 1 and Section 3.2.

5. Experiment

In this section, we conduct evaluation of existing sampling-based decoding approaches on our collected EnWiki CP-Trie dataset. We aim to estimate the inherent adaptability of sampling-based methods and the results could be used as references for the application of LLMs in open-ended tasks.

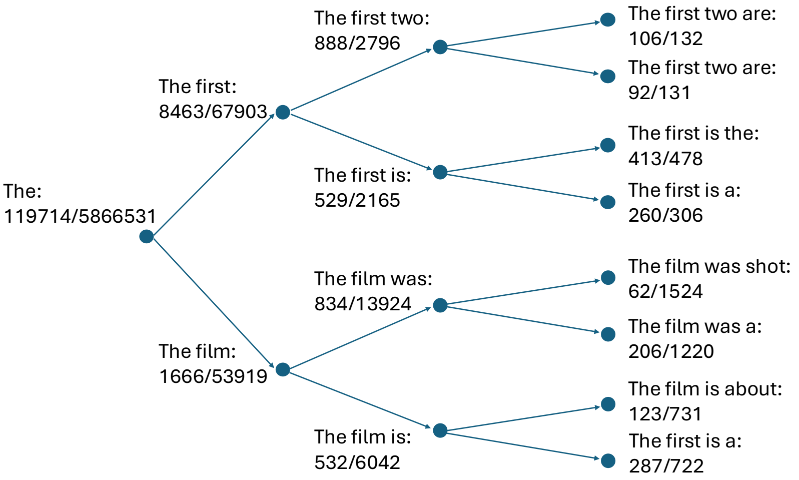


Figure VIII.3.: Illustration of the EnWiki CP-Trie. For brevity, only two child nodes are shown at each depth. The number at the left side of the slash symbol refers to the branching factor at the current node, and the number at the right side refers to the total number of leaves of the sub-tree with the current node as the root node.

5.1. Data Collection

We construct our Trie data based on the English subset of Wikipedia dataset, named EnWiki CP-Trie. As shown in Figure VIII.3, all possible words that appear after a given prefix in the dataset are treated as child nodes, with their preceding word regarded as the parent node. Starting from "Begin of Sequence" and collecting the child nodes recursively, we are able to transform the full dataset into a single prefix tree. We elaborate the main design choices in the following:

Basic Unit. It is possible to split the datasets into articles, paragraphs, sentences or n-grams. Constructing a tree based on articles or paragraphs may require more data than the training data of LLMs to guarantee an adequate number of branches (because LLMs lean to interpolate), whereas the construction based on n-grams suffers from poor contextual information and is heavily biased towards common tuplets of n tokens regardless of the context. Therefore, we adopt sentence as the basic unit, which guarantees a coherent context at sentence-level and requires much fewer data than training. It is noteworthy that a n-gram Trie [121] tends to overestimate the data support size given a prefix [15], due to the loss of information outside the contextual window, as shown in Figure VIII.1.

Filtering. To avoid invalid words or rare proper names which are unreasonable for the model to predict, we exclude the sentences containing such words by checking their presence in the WORD LIST dataset, which is available on the website [2]. It contains 354986 words in total and explicitly excludes proper names and compound words. Section titles are also excluded, which are often incomplete sentences with poor contextual information.

Statistics. Wikipedia-English dataset contains 6, 458, 670 articles, which result in EnWiki CP-Trie with 31, 557, 359 leaves, see Figure VIII.4.

Storage. The prefix tree is implemented as a nested dictionary and saved in JSON format. Since each lookup at any depth has constant complexity, the retrieval is highly efficient. Moreover, the dictionary is easily extendable if extra data are needed for a more accurate estimation of the full data support.

[2] word-list dataset homepage

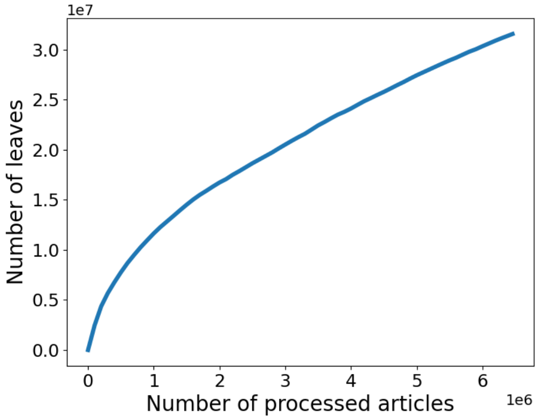


Figure VIII.4.: The total number of leaves on the CP-Trie against the total number of processed articles.

5.2. Evaluation Setup

Baselines. Our evaluation includes Top-k sampling [180, 70], Top-p sampling [108], η-sampling [104], Adaptive sampling [285] and Mirostat [14] into comparison.

Evaluation Data. To guarantee a tight lower bound of the ideal data support given different prefixes, we first sort the sub-nodes according to their total number of leaves at each depth, then we select the top 10 sub-trees with different sentence starting tokens for evaluation. Moreover, we keep the top 2 child nodes at each depth till depth 6, since the empirical data support becomes less adequate at large depth. This results in an evaluation set of 593 prefixes with varying lengths in total.

Evaluation Metrics. We measure the improvement in **diversity** via the increase of **Average Recall(AR)** at an average Risk, and the improvement of the **stability** at each decoding step in the auto-regressive process via the decrease of **Risk Standard Error (RSE)** at an average Risk. We adopt **AR** and **RSE** at average Risks of 1, 5 and 15 for comparison, representing low, medium, and high-risk regions, respectively.

LLMs. To ensure that the conclusion generalizes to different models, we adopt Llama [223, 66] family, Mistral [118, 119] family and GPT-2-XL [181] for comparison.

Tokenization. Since different LLMs are trained with different encoding methods, the evaluation has to be independent of the encoding methods. We solve this issue by constructing the CP-Trie with either a word or punctuation. For example, if the predicted next token corresponds to “sec", which is a part of the in-distribution word "section", then we regard this as a correct prediction. The second part “tion" is regarded as a hidden child node and is skipped in the evaluation.

Parameter Search. We apply grid search to determine the corresponding parameters of different sampling methods for each average Risk. To address the highly non-linear dependency between the sampling methods and their truncation parameters, we employ an efficient coarse-to-fine grid search strategy: the number of grids is initially set to 2000. If a parameter results in an average Risk within ±0.1 of the target value, it is considered a feasible solution. Otherwise, an additional grid search is performed within a smaller interval until a feasible solution is found, based on the initial search results. The grids are determined using Llama3-70B and are applied consistently across all models. As shown in Table F.2, almost all the deviations in the average Risks are much smaller than 0.1, demonstrating the robustness of our strategy.

Implementation. Our implementation mainly relies on Pytorch [170], HuggingFace [238] and OpenAI API [3] library. We implement a truncation sampling method ourselves if the official implementation is unavailable. For all methods, the minimum size of the allowed set is set to 1 to prevent breaking the sampling process.

5.3. Comparison at Different Average Risks

In this section, we conduct a comprehensive study of different truncation sampling methods at different average Risks. As discussed in Section 4.2, this allows for a fair comparison which is independent of parameter tuning. Moreover, we provide the corresponding parameters for each truncation sampling method at different average Risks, which could serve as a user reference for these methods.

As can be seen in Table VIII.1, different truncation sampling methods are compared at the average Risk of 1, 5, and 15 respectively. As discussed in Section 4.1, our defined risk and recall metrics explicitly exclude the source of risk induced by a LLM's capacity by design, thus similar parameter values correspond to the same risk level for most sampling methods across various model types and sizes. This exactly showcases the advantage of our evaluation being tuning-independent and sustainable to the rapid update of LLMs. Among the evaluated methods, Eta-sampling [104] is sensitive to the changes of model type and size, which might hinder its practical significance especially at a low risk level.

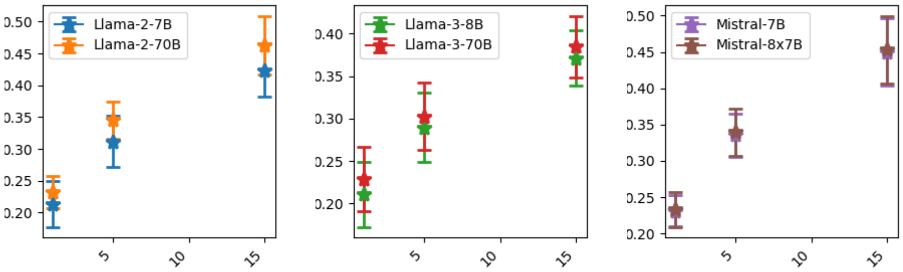
Regarding diversity, i.e., the average recall at the same average Risk, Adaptive sampling [285] and Mirostat [14] are the best and second performers, which consistently outperform the Top-k baseline by a considerable margin. Top-p mostly exhibits inferior recall comparing to the Top-k baseline, so does Eta-sampling at the average Risk of 1. As for the stability represented by standard error of Risks, Top-k sampling reaches the best scores in most cases. In comparison, Adaptive sampling and Mirostat deliver comparable standard error of risks to Top-k sampling, whereas Top-p sampling and Eta-sampling are again inferior. Considering both diversity and stability, Adaptive sampling and Mirostat are the top 2 adaptive methods to be recommended, whereas Top-p sampling shall be the last two methods to be considered.

[3] https://pypi.org/project/openai/

Model	Method	Avg. Risk 1			Avg. Risk 5			Avg. Risk 15		
		Parameter	RSE ↓	AR ↑	Parameter	RSE ↓	AR ↑	Parameter	RSE ↓	AR ↑
GPT2-XL	Adaptive	9.5e-4	0.006	**0.252**	1.1e-4	0.679	**0.339**	2.5e-05	2.241	**0.413**
	Mirostat	4.425	**0.005**	0.236	5.9475	0.717	0.326	6.76	2.501	0.401
	Top-k	15	0.006	0.220	64	**0.613**	0.290	184	**1.781**	0.340
	Eta	0.318	0.013	0.198	0.011	1.484	0.301	0.001	4.261	0.404
	Top-p	0.5705	0.015	0.170	0.746	2.129	0.240	0.8555	6.210	0.338
Llama-2-7b	Adaptive	1.1e-3	0.154	**0.257**	1.4e-4	0.856	**0.364**	3.1e-5	2.966	0.470
	Mirostat	4.253	0.133	0.236	5.82	0.650	0.349	6.628	2.286	**0.474**
	Top-k	14	**0.126**	0.226	61	**0.587**	0.296	177	**1.722**	0.369
	Eta	0.512	0.563	0.192	0.023	2.599	0.297	0.002	6.531	0.407
	Top-p	0.54	0.529	0.156	0.7665	2.331	0.254	0.9	6.208	0.400
Llama-2-70b	Adaptive	0.0011	0.142	**0.269**	1.2e-4	0.796	0.374	2.3e-5	2.697	0.485
	Mirostat	4.16	0.135	0.238	5.7875	0.684	0.353	6.67	2.125	0.478
	Top-k	14	**0.128**	0.232	60	**0.583**	0.307	174	**1.712**	0.375
	Eta	0.092	0.304	0.236	0.003	1.590	**0.378**	2.1e-4	4.243	**0.510**
	Top-p	0.6535	0.475	0.189	0.8465	2.136	0.316	0.9395	5.522	0.468
Llama-3-8B	Adaptive	1.1e-3	0.167	**0.260**	1.7e-4	0.787	**0.343**	3.7e-5	2.685	**0.418**
	Mirostat	4.24	0.139	0.230	5.8175	0.804	0.318	6.693	2.630	0.393
	Top-k	14	**0.128**	0.228	59	**0.576**	0.290	172	**1.701**	0.346
	Eta	0.673	0.445	0.181	0.029	2.112	0.271	0.002	6.009	0.373
	Top-p	0.5395	0.451	0.154	0.736	2.061	0.224	0.855	5.770	0.326
Llama-3-70B	Adaptive	1.1e-3	0.137	**0.263**	1.4e-4	0.787	**0.353**	3.16e-5	2.778	**0.424**
	Mirostat	4.21	0.138	0.230	5.91	0.708	0.332	6.84	2.193	0.417
	Top-k	14	**0.127**	0.230	60	**0.581**	0.295	173	**1.695**	0.352
	Eta	0.37	0.137	**0.263**	0.014	2.231	0.295	0.001	6.265	0.398
	Top-p	0.5695	0.502	0.158	0.758	2.386	0.237	0.8705	6.685	0.332
Mixtral-7B	Adaptive	0.00105	0.152	**0.260**	1.2e-4	0.809	0.364	2.2e-5	2.757	0.466
	Mirostat	4.1825	0.141	0.236	5.8125	0.721	0.345	6.71	2.213	0.468
	Top-k	14	**0.126**	0.224	62	**0.596**	0.297	181	**1.759**	0.364
	Eta	0.075	0.307	0.243	0.003	1.542	**0.368**	1.96e-4	4.712	**0.505**
	Top-p	0.6565	0.539	0.194	0.8375	2.476	0.303	0.9315	6.315	0.447
Mixtral-8x7B	Adaptive	0.00105	0.148	**0.265**	1.1e-4	0.798	0.372	2.1e-5	2.802	0.476
	Mirostat	4.2775	0.143	0.238	5.845	0.710	0.346	6.6875	2.213	0.461
	Top-k	15	**0.134**	0.229	63	**0.598**	0.301	183	**1.757**	0.366
	Eta	0.087	0.335	0.241	0.003	1.822	**0.375**	2.15e-4	4.922	**0.506**
	Top-p	0.6505	0.535	0.192	0.8375	2.423	0.303	0.9325	6.139	0.456

Table VIII.1.: Risk Standard Error (RSE, indicating stability) and Average Recall (AR, indicating diversity) of different truncation sampling methods at different average Risks using different models. The corresponding parameter of each method at an average risk level is also provided. The best and worst scores are marked in bold and underlined, respectively. For more detailed results, please refer to Appendix 1.

We also show in Figure VIII.5 that larger models of the same family have higher average recall at the same risk level comparing to the smaller ones. This conforms to the fact that larger models better captures the human text distribution. Please note that our metrics does not allow a direct comparison between different model families, mainly due to their different vocabulary sizes and tokenizers, e.g., Llama-3 has a 128,256 vocabulary size, while Llama-2 has only 32, 000 vocabulary size. Moreover, our metrics also explicitly exclude the source of risk within the optimal allowed set, which is heavily dependent on a LLM's capacity.



(a) Llama-2 family. (b) Llama-3 family. (c) Mistral family

Figure VIII.5.: Comparing the average Recalls at given average Risks using different model sizes.

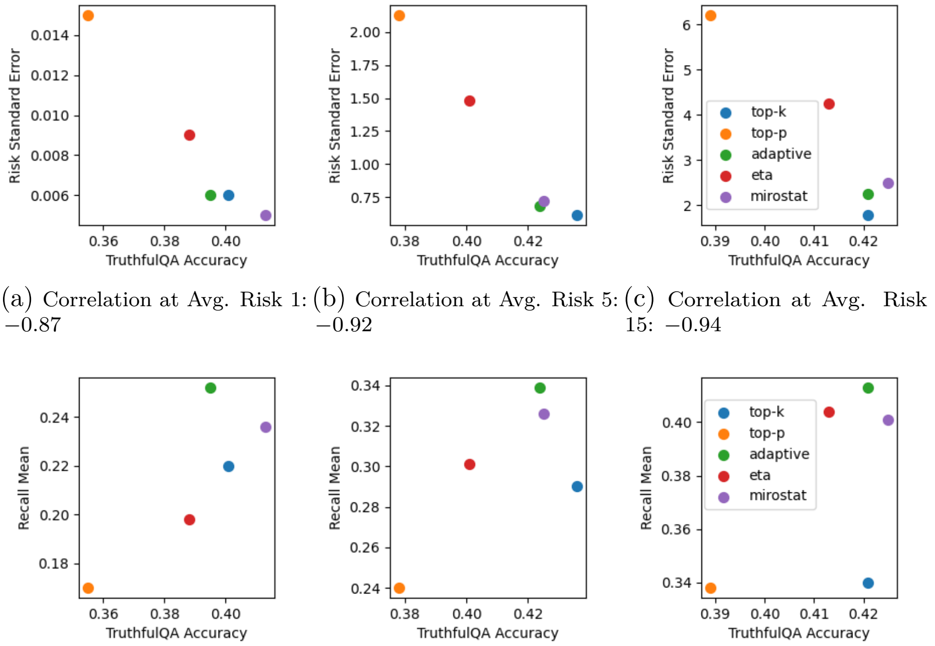
Methods	Mean(std) Accuracy ↑		
	Avg. Risk 1	Avg. Risk 5	Avg. Risk 15
Greedy	0.338		
Naïve	0.421(0.004)		
Top-k	0.401(0.010)	**0.436**(0.008)	0.421(0.010)
Top-p	0.355(0.013)	0.378(0.011)	0.389(0.012)
Adaptive	0.395(0.012)	0.424(0.011)	0.421(0.009)
Eta	0.388(0.005)	0.401(0.013)	0.413(0.026)
Mirostat	0.413(0.010)	0.425(0.013)	**0.425**(0.009)

Table VIII.2.: Evaluation on the TruthfulQA benchmark under the open-ended generation setup. Naive sampling refers to sampling without truncation. The best and worst scores are marked in bold and underlined, respectively. For more details, please refer to Appendix 1.

5.4. Validation on TruthfulQA Benchmark

Although our evaluation protocol is grounded by the thorough design process with reasonable simplifications, we would like to verify its effectiveness in the real-world scenario using the TruthfulQA Benchmark [142]. The evaluation results using gpt2-xl are shown in Section 5.3. For all the methods other than greedy decoding, we run 3 times at each average risk level and report the mean and standard deviation (parenthetical value).

It can be observed that greedy decoding falls far behind sampling-based decoding strategies, which conforms to the issue of likelihood-oriented decoding discussed in Section 1, as well as the findings in recent studies [46, 232, 233, 204]. All the truncation sampling methods at the low risk level achieves lower accuracy comparing to Naive sampling, due to the over-truncation of the decoding paths. At the average risk level of 5, all the truncation sampling methods slightly improve their own accuracy. Top-k sampling, Adaptive sampling and Mirostat also reach comparable or slightly higher accuracy in comparison to Naive sampling. However, further increased average risk level (means improved average recall and thus diversity) does not benefit the performance on TruthfulQA, which is plausible. Moreover, there exists a even stronger correlation between Risk SE (Standard Error of Risks) and TruthfulQA accuracy, validating the importance of stability when evaluating an adaptive decoding method. The strong correlation between TruthfulQA accuracy and



(a) Correlation at Avg. Risk 1: −0.87

(b) Correlation at Avg. Risk 5: −0.92

(c) Correlation at Avg. Risk 15: −0.94

(d) Correlation at Avg. Risk 1: 0.83

(e) Correlation at Avg. Risk 5: 0.83

(f) Correlation at Avg. Risk 15: 0.50

Figure VIII.6.: The scatter plots of TruthfulQA accuracy against risk standard error (first row) and recall mean (second row) at different average Risks.

our proposed average recall as well as standard error of risks at different average Risks validate the soundness and effectiveness of our evaluation method.

6. Revisiting Existing Evaluation

In this section, we revisit the recent study [204] by comparing sampling-based decoding methods at the same average Risks. We adopt the official implementation of Shi et al. [204]. Following their setups, we adopt Llama-2-7B on MBPP [10], HumanEval [10] and GSM8K [46] to evaluate coding and math problem solving performance. Mean and standard deviation for three runs are reported in Table VIII.3, Table VIII.4 and Table VIII.5, respectively.

For all the three tasks, Mirostat does not perform well in general, probably because it is based on the Zipf-law of natural language and thus not suitable for code and math tasks. Notably, our greedy decoding baseline achieves significantly lower result than reported by Shi et al. [204] on HumanEval. Our results should be plausible, because the instruction tuned Llama-2-7B only achieves 7.9 according to Meta-Llama Github[4] .

While their study concludes that deterministic methods outperform sampling methods across most tasks, our evaluation reveals that sampling methods are indeed underestimated. In contrast to the conclusion in Shi et al. [204], all the sampling-based decoding methods could achieve better performance than greedy decoding on HumanEval in Table VIII.4. In addition, Top-p and eta sampling also beat greedy decoding at a low average Risk on GSM8K in Table VIII.5. This observation underscores the challenges in parameter selection for sampling-based decoding, which is effectively addressed by our method.

[4] https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md

Methods	Avg. Risk 1	Avg. Risk 5	Avg. Risk 15
Top-k	19.70 (0.50)	21.00 (2.30)	20.50 (0.30)
Top-p	21.50 (1.30)	**21.10** (0.40)	**21.70** (0.70)
Mirostat	9.50 (0.30)	8.80 (2.00)	8.80 (0.40)
Eta	**22.10** (0.70)	19.10 (0.40)	19.70 (0.40)
Greedy	24.00		

Table VIII.3.: Pass@1 accuracy on MBPP. It is consistent to the observation by Shi et al. [204] that sampling methods are inferior to greedy decoding.

Methods	Avg. Risk 1	Avg. Risk 5	Avg. Risk 15
Top-k	**5.68** (2.00)	5.08 (1.52)	6.50 (0.76)
Top-p	3.46 (1.05)	5.89 (0.76)	**6.52** (2.43)
Mirostat	3.25 (0.76)	4.27 (1.00)	4.68 (0.58)
Eta	2.64 (2.01)	**6.91** (1.04)	6.10 (1.32)
Greedy	2.44		

Table VIII.4.: Pass@1 accuracy on HumanEval. Sampling methods perform better with higher average Recalls and Risks.

Methods	Avg. Risk 1	Avg. Risk 5	Avg. Risk 15
Top-k	7.56 (5.39)	**11.90** (0.80)	**11.73** (0.57)
Top-p	**14.13** (0.47)	8.72 (6.18)	11.67 (0.11)
Mirostat	5.46 (0.47)	5.74 (0.64)	3.46 (2.10)
Eta	13.72 (0.46)	8.42 (5.54)	11.22 (0.75)
Greedy	13.19		

Table VIII.5.: Accuracy on GSM8K. Top-p and eta sampling outperforms greedy decoding at an average Risk of 1.

7. Conclusion

In this chapter, we propose an evaluation protocol to assess the trade-off between diversity and quality of truncation sampling methods for open-ended text generation. Our evaluation enjoys the merit of being independent of parameter tuning for the curated tasks. The evaluation results also serve as a user reference for different downstream tasks.

Chapter IX.

Conclusion and Outlook

1. Conclusion

This thesis explored the limitations of traditional neural probabilistic scoring techniques in both the attention and output layers of deep learning models, proposing novel methods that enhance expressiveness, reliability, and structural alignment in vision and language tasks. Softmax-based scoring, while foundational, tends to produce unimodal distributions, constraining its ability to model complex, multi-peaked attention patterns. This limitation is particularly pronounced when increasing the temperature to promote diversity, as higher temperatures can introduce noise and instability into the model's predictions. To address this, Chapter III introduced Multimax as a more flexible alternative, enabling richer, multi-modal attention distributions while maintaining stability in both attention and prediction layers.

To overcome the structural shortcomings of permutation-invariant attention, the thesis presented Hyperformer in Chapter V and SP-ViT in Chapter IV, which inject learned structural priors into self-attention. Hyperformer extends attention to hypergraphs and uses graph distance–based positional encoding to capture connectivity priors—edges and hyperedges—enabling the model to reason over higher-order relationships. SP-ViT incorporates a learnable 2D spatial prior, granting the model inductive bias for spatial patterns and improving performance on vision tasks. Together, these approaches yield a more expressive and robust attention mechanism capable of modeling complex dependencies.

In output generation, techniques like temperature scaling, label smoothing, and sampling-based decoding adjust the softmax distribution to enhance stability and generalization. Despite their utility, each has drawbacks: temperature scaling can amplify noise; label smoothing may induce overconfidence in wrong predictions and collapse intra-class variation; and fixed sampling schemes can either overly restrict diversity or introduce incoherence. To address these issues, MaxSup was introduced in Chapter VII to correct label smoothing's two core flaws—overconfidence and feature collapse—thereby improving calibration and preserving intra-class diversity. Finally, Chapter VIII offers a systematic guideline for balancing diversity and risk during decoding of Large Language Models, providing practical rules for open-ended text generation where creative variation and coherence must be tightly managed.

Overall, the contributions of this thesis advance neural probabilistic scoring by:

- Multimax: extending attention capacity beyond unimodal softmax.

- Hyperformer, BlockGCN and SP-ViT: embedding structural and spatial priors into attention.
- MaxSup: refining output regularization to avoid overconfidence and feature collapse.
- LLM-Sampling: establishing a risk-aware, diversity-controlled decoding framework.

By unifying adaptive scoring in attention (MultiMax, Hyperformer, BlockGCN, SP-ViT) with principled output modulation (MaxSup, LLM-Sampling), this work establishes a cohesive paradigm of neural probabilistic modeling—treating probability distributions as first-class design elements across both perception and generation.

While the methods proposed in this thesis demonstrate consistent improvements across a range of vision and language tasks, the scale of the models and datasets studied has been constrained by available computational resources. Consequently, the full potential of these approaches—particularly in large-scale or more heterogeneous settings—remains to be fully explored. Moreover, the tasks evaluated thus far cover only a subset of the broader application space these methods are designed to address. These limitations highlight the need for future work to investigate scalability, generalizability, and broader applicability. The following section outlines several promising directions for extending and building upon this work.

2. Outlook

Building on our suite of neural probabilistic scoring approaches for both the attention and output layers, several promising directions emerge for future research.

One key avenue is scaling up both models and datasets to better understand how our methods perform in large-scale settings. While resource constraints have limited such exploration within the academic context, this line of investigation could be highly valuable for industrial applications. It remains to be seen whether the gains observed in our current experiments will persist, diminish, or even amplify as scale increases. However, we are optimistic given that our methods have been deliberately designed to be as generic and adaptable as possible, for several reasons:

- Hypergraph self-attention and structural encodings in Chapters V and VI extend standard self-attention and positional encoding mechanisms by incorporating additional structural information in a learnable form. This design inherently provides greater modeling capacity compared to their vanilla counterparts. While the reliance on prior information becomes less critical as dataset size and model capacity increase, our methods continue to offer efficiency gains in the learning process—without compromising expressiveness—thanks to their general and trainable nature. In contrast, hard-coded priors such as local convolutional windows are effective in low-data regimes but often prove overly restrictive for large-scale training.
- As improved generalizations to SoftMax and Label Smoothing, the advantages of MultiMax in Chapter III and MaxSup in Chapter VII in enhancing learning have been demonstrated both theoretically and empirically, making them broadly applicable across neural network architectures. These methods are among the most promising approaches not only for accelerating convergence but also for significantly extending the representational capacity of models. Importantly, they are particularly well-suited for foundation models, which follow a pretraining–fine-tuning paradigm

and are expected to generalize well to downstream tasks. In such settings, the benefits of improved representation learning offered by MultiMax and MaxSup are especially evident—going beyond mere improvements in classification accuracy or reductions in language modeling perplexity reported in our paper. We anticipate future research will explore the application of MultiMax to large language models (LLMs), potentially yielding improvements across multiple dimensions. For instance, LLMs are often prone to incorporating irrelevant contextual information into their latent representations, a known limitation of the standard attention mechanism [235]. MultiMax, with its enhanced sparsity compared to SoftMax, is expected to mitigate this issue by enabling more focused and interpretable attention patterns.

- The llm-sampling work in Chapter VIII also benefits from scaling up both datasets and models. With sufficiently large datasets, we can construct a Trie structure that closely approximates the ideal data support. This has two key advantages: it enhances the accuracy of both our evaluation and hyperparameter recommendations, and it enables broader applications—such as the development of more sophisticated adaptive sampling methods beyond truncation [73]. Furthermore, increasing model capacity improves the alignment between model predictions and the empirical distribution, making truncation-based adaptive sampling methods more robust. As a result, evaluation outcomes and hyperparameter recommendation become even more reliable.

Another promising direction is to apply our proposed approaches beyond the specific tasks explored in this thesis. As noted earlier, our methods are intentionally designed to be as generic as possible, and are therefore not limited to the studied applications. We believe they have the potential to benefit a wide range of tasks and anticipate that future research will adopt and extend our methods to address diverse problems. More specifically, we highlight the following considerations:

- MultiMax is not limited to attention layers. As a drop-in replacement for SoftMax, it can be applied to any task where SoftMax is used—provided that the optimization process can learn the appropriate parameters. For example, in our study, replacing SoftMax with MultiMax in the output layer also led to performance gains. A particularly promising application is reinforcement learning, where effectively balancing exploration and exploitation [215] is essential. For instance, recent advances in test-time scaling of LLMs [168, 89] heavily rely on post-training via reinforcement learning, especially policy optimization algorithms such as PPO and GRPO [194, 199]. Replacing the Softmax in the policy network's output layer with MultiMax is expected to enhance this balance, potentially leading to more efficient exploration strategies and better overall performance.
- The Hypergraph Self-Attention and Structural Encodings in Hyperformer and BlockGCN are not limited to skeleton-based action recognition. Owing to their generic and learnable design, both methods effectively capture high-order structural information, making them applicable to a wide range of graph learning tasks. For instance, Hyperformer's Hypergraph Self-Attention has demonstrated effectiveness in 3D pose estimation [34], video understanding [184], and multi-behavior recommendation [160]. Similarly, BlockGCN's topological encoding has been successfully adapted for multivariate time-series anomaly detection [148].

- The systematic study on the impact of parameter tuning in LLM-Sampling extends beyond merely evaluating and providing user guidance for sampling-based decoding methods. It can also be applied to other parameter-sensitive challenges and to the development of new decoding strategies for large language models (LLMs). For example, Arias et al. [7] have expanded our evaluation framework to encompass deterministic decoding methods, while Nguyen et al. [164] have proposed a novel adaptive sampling approach inspired by our benchmark.

In summary, this thesis lays a unified foundation for rethinking probabilistic scoring in neural networks, emphasizing flexibility, structural awareness, and practical scalability. By treating scoring functions not as fixed components but as carefully designed, learnable function families—grounded in theoretical analysis—we have opened new avenues for enhancing both attention mechanisms and output generation across a wide range of tasks. As models and datasets continue to grow in complexity and scale, we expect our contributions to serve not only as practical tools but also as conceptual building blocks for the next generation of adaptive and interpretable AI systems. We look forward to seeing these ideas further explored and expanded in future research.

Bibliography

[1] Samira Abnar and Willem Zuidema. “Quantifying Attention Flow in Transformers”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2020, pp. 4190–4197.

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. “Gpt-4 technical report”. In: *arXiv preprint arXiv:2303.08774* (2023).

[3] Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. “Persistence images: A stable vector representation of persistent homology”. In: *Journal of Machine Learning Research* 18 (2017).

[4] Abien Fred Agarap. “Deep learning using rectified linear units (relu)”. In: *arXiv preprint arXiv:1803.08375* (2018).

[5] Mehmet E Aktas, Esra Akbas, and Ahmed El Fatmaoui. “Persistence homology of networks: methods and applications”. In: *Applied Network Science* 4.1 (2019), pp. 1–28.

[6] Duarte Alves, Nuno Guerreiro, João Alves, José Pombal, Ricardo Rei, José de Souza, Pierre Colombo, and André FT Martins. “Steering Large Language Models for Machine Translation with Finetuning and In-Context Learning”. In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. 2023, pp. 11127–11148.

[7] Esteban Garces Arias, Meimingwei Li, Christian Heumann, and Matthias Aßenmacher. “Decoding Decoded: Understanding Hyperparameter Effects in Open-Ended Text Generation”. In: *Proceedings of the 31st International Conference on Computational Linguistics*. 2025, pp. 9992–10020.

[8] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. “Vivit: A video vision transformer”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 6836–6846.

[9] Nieves Atienza, Rocío González-Díaz, and Manuel Soriano-Trigueros. “On the stability of persistent entropy and new summary functions for topological data analysis”. In: *Pattern Recognition* 107 (2020), p. 107509.

[10] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. “Program synthesis with large language models”. In: *arXiv preprint arXiv:2108.07732* (2021).

[11] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. “Layer normalization”. In: *arXiv preprint arXiv:1607.06450* (2016).

[12] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate". In: *arXiv preprint arXiv:1409.0473* (2014).

[13] Song Bai, Feihu Zhang, and Philip HS Torr. "Hypergraph convolution and hypergraph attention". In: *Pattern Recognition* 110 (2021), p. 107637.

[14] Sourya Basu, Govardana Sachitanandam Ramachandran, Nitish Shirish Keskar, and Lav R Varshney. "Mirostat: A neural text decoding algorithm that directly controls perplexity". In: *International Conference on Learning Representations (ICLR)* (2021).

[15] Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. "A neural probabilistic language model". In: *Advances in Neural Information Processing Systems (NeurIPS))* 13 (2000).

[16] Eric Berry, Yen-Chi Chen, Jessi Cisewski-Kehe, and Brittany Terese Fasy. "Functional summaries of persistence diagrams". In: *Journal of Applied and Computational Topology* 4.2 (2020), pp. 211–262.

[17] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Springer, 2006.

[18] Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. "Findings of the 2014 Workshop on Statistical Machine Translation". In: *Proceedings of the Ninth Workshop on Statistical Machine Translation*. Ed. by Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, and Lucia Specia. Baltimore, Maryland, USA: Association for Computational Linguistics, June 2014, pp. 12–58. DOI: 10.3115/v1/W14-3302. URL: https://aclanthology.org/W14-3302/.

[19] Stephen Boyd, Lin Xiao, and Almir Mutapcic. "Subgradient methods". In: *lecture notes of EE392o, Stanford University, Autumn Quarter* (2003).

[20] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems*. Vol. 33. 2020, pp. 1877–1901.

[21] Peter Bubenik et al. "Statistical topological data analysis using persistence landscapes." In: *Journal of Machine Learning Research* 16.1 (2015), pp. 77–102.

[22] James M Buchanan. "The relevance of Pareto optimality". In: *Journal of conflict resolution* (1962).

[23] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. "Realtime Multi-person 2D Pose Estimation Using Part Affinity Fields". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 1302–1310.

[24] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. “End-to-end object detection with transformers”. In: *European conference on computer vision*. Springer. 2020, pp. 213–229.

[25] Mathieu Carriere, Marco Cuturi, and Steve Oudot. “Sliced Wasserstein kernel for persistence diagrams”. In: *International Conference on Machine Learning*. PMLR. 2017, pp. 664–673.

[26] Mathieu Carrière, Frédéric Chazal, Yuichi Ike, Théo Lacombe, Martin Royer, and Yuhei Umeda. “Perslay: A neural network layer for persistence diagrams and new graph topological signatures”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 2786–2796.

[27] Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Jan Niehues, Sebastian Stüker, Katsuitho Sudoh, Koichiro Yoshino, and Christian Federmann. “Overview of the iwslt 2017 evaluation campaign”. In: *Proceedings of the 14th International Workshop on Spoken Language Translation*. 2017.

[28] Kit C Chan, Umar Islambekov, Alexey Luchinsky, and Rebecca Sanders. “A computationally efficient framework for vector representation of persistence diagrams”. In: *The Journal of Machine Learning Research* 23.1 (2022), pp. 12281–12313.

[29] Keshigeyan Chandrasegaran, Ngoc-Trung Tran, Yunqing Zhao, and Ngai-Man Cheung. “Revisiting label smoothing and knowledge distillation compatibility: What was missing?” In: *International Conference on Machine Learning*. PMLR. 2022, pp. 2890–2916.

[30] Frédéric Chazal and Bertrand Michel. “An introduction to topological data analysis: fundamental and practical aspects for data scientists”. In: *Frontiers in Artificial Intelligence* 4 (2021).

[31] Hila Chefer, Shir Gur, and Lior Wolf. “Transformer Interpretability Beyond Attention Visualization”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2021, pp. 782–791.

[32] Chun-Fu Chen, Quanfu Fan, and Rameswar Panda. “CrossViT: Cross-Attention Multi-Scale Vision Transformer for Image Classification”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 357–366.

[33] Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. “Measuring and relieving the over-smoothing problem for graph neural networks from the topological view”. In: *Proceedings of the AAAI conference on artificial intelligence*. 2020.

[34] Hanyuan Chen, Jun-Yan He, Wangmeng Xiang, Zhi-Qi Cheng, Wei Liu, Hanbing Liu, Bin Luo, Yifeng Geng, and Xuansong Xie. “HDFormer: high-order directed transformer for 3D human pose estimation”. In: *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*. 2023, pp. 581–589.

[35] Phil Chen, Mikhal Itkina, Ransalu Senanayake, and Mykel J Kochenderfer. “Evidential softmax for sparse multimodal distributions in deep generative models”. In: *Advances in Neural Information Processing Systems (NeurIPS)* (2021).

[36] Shiming Chen, Guosen Xie, Yang Liu, Qinmu Peng, Baigui Sun, Hao Li, Xinge You, and Ling Shao. “Hsva: Hierarchical semantic-visual adaptation for zero-shot learning”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 16622–16634.

[37] Yen-Chi Chen, Daren Wang, Alessandro Rinaldo, and Larry Wasserman. “Statistical analysis of persistence intensity functions”. In: *arXiv preprint arXiv:1510.02502* (2015).

[38] Yuxin Chen, Ziqi Zhang, Chunfeng Yuan, Bing Li, Ying Deng, and Weiming Hu. “Channel-Wise Topology Refinement Graph Convolution for Skeleton-Based Action Recognition”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 13359–13368.

[39] Zhan Chen, Sicheng Li, Bing Yang, Qinghan Li, and Hong Liu. “Multi-Scale Spatial Temporal Graph Convolutional Network for Skeleton-Based Action Recognition”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2021, pp. 1113–1122.

[40] Ke Cheng, Yifan Zhang, Congqi Cao, Lei Shi, Jian Cheng, and Hanqing Lu. “Decoupling GCN with DropGraph Module for Skeleton-Based Action Recognition”. In: *European Conference on Computer Vision*. 2020, pp. 536–553.

[41] Ke Cheng, Yifan Zhang, Xiangyu He, Weihan Chen, Jian Cheng, and Hanqing Lu. “Skeleton-based action recognition with shift graph convolutional network”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 183–192.

[42] Hyung-gun Chi, Myoung Hoon Ha, Seunggeun Chi, Sang Wan Lee, Qixing Huang, and Karthik Ramani. “InfoGCN: Representation Learning for Human Skeleton-Based Action Recognition”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 20186–20196.

[43] I Eli Chien, Huozhi Zhou, and Pan Li. “Active learning over hypergraphs with pointwise and pairwise queries”. In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR. 2019, pp. 2466–2475.

[44] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, and Chunhua Shen. “Conditional Positional Encodings for Vision Transformers”. In: *The Eleventh International Conference on Learning Representations*.

[45] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. “Fast and accurate deep network learning by exponential linear units (elus)”. In: *arXiv preprint arXiv:1511.07289* (2015).

[46] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. “Training verifiers to solve math word problems”. In: *URL https://arxiv.org/abs/2110.14168* (2021).

[47] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. “Stability of persistence diagrams”. In: *Proceedings of the Twenty-first Annual Symposium on Computational Geometry*. 2005, pp. 263–271.

[48] Stéphane d’Ascoli, Hugo Touvron, Matthew Leavitt, Ari Morcos, Giulio Biroli, and Levent Sagun. “ConViT: Improving Vision Transformers with Soft Convolutional Inductive Biases”. In: *ICML 2021: 38th International Conference on Machine Learning*. 2021.

[49] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. “Coatnet: Marrying convolution and attention for all data sizes”. In: *Advances in neural information processing systems* 34 (2021), pp. 3965–3977.

[50] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. “Transformer-xl: Attentive language models beyond a fixed-length context”. In: *arXiv preprint arXiv:1901.02860* (2019).

[51] Yann Dauphin and Ekin Dogus Cubuk. “Deconstructing the regularization of batchnorm”. In: *International Conference on Learning Representations.* 2021.

[52] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. “Convolutional neural networks on graphs with fast localized spectral filtering”. In: *Advances in Neural Information Processing systems* 29 (2016).

[53] Andac Demir, Baris Coskunuzer, Yulia Gel, Ignacio Segovia-Dominguez, Yuzhou Chen, and Bulent Kiziltan. “ToDD: Topological compound fingerprinting in computer-aided drug discovery”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 27978–27993.

[54] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. “ImageNet: A large-scale hierarchical image database”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition.* 2009, pp. 248–255.

[55] Yuntian Deng, Yoon Kim, Justin Chiu, Demi Guo, and Alexander Rush. “Latent alignment and variational attention”. In: *Advances in eural information processing systems (NeurIPS)* (2018).

[56] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers).* Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186.

[57] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers).* 2019, pp. 4171–4186.

[58] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina N. Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers).* 2018, pp. 4171–4186.

[59] Josep Díaz, Jordi Petit, and Maria Serna. “A survey of graph layout problems”. In: *ACM Computing Surveys (CSUR)* 34.3 (2002), pp. 313–356.

[60] Hantian Ding, Zijian Wang, Giovanni Paolini, Varun Kumar, Anoop Deoras, Dan Roth, and Stefano Soatto. “Fewer truncations improve language modeling”. In: *International Conference on Machine Learning (ICML)* (2024).

[61] Kaize Ding, Jianling Wang, Jundong Li, Dingcheng Li, and Huan Liu. “Be More with Less: Hypergraph Attention Networks for Inductive Text Classification”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP).* Online: Association for Computational Linguistics, Nov. 2020, pp. 4927–4936.

[62] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. "Cswin transformer: A general vision transformer backbone with cross-shaped windows". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 12124–12134.

[63] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *ICLR 2021: The Ninth International Conference on Learning Representations*. 2021.

[64] Yong Du, Wei Wang, and Liang Wang. "Hierarchical recurrent neural network for skeleton based action recognition". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 1110–1118.

[65] Haodong Duan, Yue Zhao, Kai Chen, Dahua Lin, and Bo Dai. "Revisiting skeleton-based action recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 2969–2978.

[66] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. "The llama 3 herd of models". In: *arXiv preprint arXiv:2407.21783* (2024).

[67] Herbert Edelsbrunner and John L Harer. *Computational topology: an introduction*. American Mathematical Society, 2022.

[68] Herbert Edelsbrunner and Dmitriy Morozov. *Persistent homology: theory and practice*. eScholarship, University of California, 2013.

[69] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning". In: *Neural networks* (2018).

[70] Angela Fan, Mike Lewis, and Yann Dauphin. "Hierarchical neural story generation". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. 2018.

[71] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. "Hypergraph neural networks". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2019, pp. 3558–3565.

[72] Yutong Feng, Jianwen Jiang, Mingqian Tang, Rong Jin, and Yue Gao. "Rethinking supervised pre-training for better downstream transferring". In: *arXiv preprint arXiv:2110.06014* (2021).

[73] Matthew Finlayson, John Hewitt, Alexander Koller, Swabha Swayamdipta, and Ashish Sabharwal. "Closing the curious case of neural text degeneration". In: *International Conference on Learning Representations (ICLR)* (2024).

[74] Edward Fredkin. "Trie memory". In: *Communications of the ACM* 3.9 (1960), pp. 490–499.

[75] Markus Freitag and Yaser Al-Onaizan. "Beam Search Strategies for Neural Machine Translation". In: *Proceedings of the First Workshop on Neural Machine Translation*. Ed. by Thang Luong, Alexandra Birch, Graham Neubig, and Andrew Finch. Vancouver: Association for Computational Linguistics, Aug. 2017, pp. 56–60. DOI: 10.18653/v1/W17-3207. URL: https://aclanthology.org/W17-3207/.

[76] Octavian Ganea, Sylvain Gelly, Gary Bécigneul, and Aliaksei Severyn. “Breaking the softmax bottleneck via learnable monotonic pointwise non-linearities”. In: *International Conference on Machine Learning (ICML)* (2019).

[77] Bolin Gao and Lacra Pavel. “On the properties of the softmax function with application in game theory and reinforcement learning”. In: *arXiv preprint arXiv:1704.00805* (2017).

[78] Lingling Gao, Yanli Ji, Yang Yang, and HengTao Shen. “Global-Local Cross-View Fisher Discrimination for View-Invariant Action Recognition”. In: *Proceedings of the 30th ACM International Conference on Multimedia.* 2022, pp. 5255–5264.

[79] Tianyu Gao, Xingcheng Yao, and Danqi Chen. “Simcse: Simple contrastive learning of sentence embeddings”. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing.* 2021.

[80] Yingbo Gao, Weiyue Wang, Christian Herold, Zijian Yang, and Hermann Ney. “Towards a better understanding of label smoothing in neural machine translation”. In: *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing.* 2020, pp. 212–223.

[81] Connall Garrod and Jonathan P Keating. “The Persistence of Neural Collapse Despite Low-Rank Bias: An Analytic Perspective Through Unconstrained Features”. In: *arXiv preprint arXiv:2410.23169* (2024).

[82] Jonas Gehring, Michael Auli, David Grangier, and Yann Dauphin. “A Convolutional Encoder Model for Neural Machine Translation”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).* 2017, pp. 123–135.

[83] Chavoosh Ghasemi, Hamed Yousefi, Kang G Shin, and Beichuan Zhang. “On the granularity of trie-based data structures for name lookups and updates”. In: *IEEE/ACM Transactions on Networking* 27.2 (2019), pp. 777–789.

[84] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. “Neural Message Passing for Quantum Chemistry”. In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70.* 2017, pp. 1263–1272.

[85] Chengyue Gong, Dilin Wang, Meng Li, Vikas Chandra, and Qiang Liu. “Improve vision transformers training by suppressing over-smoothing”. In: *arXiv preprint arXiv:2104.12753* (2021).

[86] Chengyue Gong, Dilin Wang, Meng Li, Vikas Chandra, and Qiang Liu. “Vision transformers with patch diversification”. In: *arXiv preprint arXiv:2104.12753* (2021).

[87] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning.* MIT Press, 2016.

[88] Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. “LeViT: a Vision Transformer in ConvNet’s Clothing for Faster Inference.” In: *arXiv preprint arXiv:2104.01136* (2021).

[89] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning". In: *arXiv preprint arXiv:2501.12948* (2025).

[90] Li Guo, Keith Ross, Zifan Zhao, George Andriopoulos, Shuyang Ling, Yufeng Xu, and Zixuan Dong. "Cross entropy versus label smoothing: A neural collapse perspective". In: *arXiv preprint arXiv:2402.03979* (2024).

[91] Qiushan Guo, Xinjiang Wang, Yichao Wu, Zhipeng Yu, Ding Liang, Xiaolin Hu, and Ping Luo. "Online knowledge distillation via collaborative learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 11020–11029.

[92] Ankit Gupta, Guy Dar, Shaya Goodman, David Ciprut, and Jonathan Berant. "Memory-efficient Transformers via Top-k Attention". In: *Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing*. 2021, pp. 39–52.

[93] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing XU, and Yunhe Wang. "Transformer in Transformer". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 15908–15919.

[94] Arman Hasanzadeh, Ehsan Hajiramezanali, Shahin Boluki, Mingyuan Zhou, Nick Duffield, Krishna Narayanan, and Xiaoning Qian. "Bayesian graph neural networks with adaptive connection sampling". In: *International Conference on Machine Learning (ICML)* (2020).

[95] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. "Masked autoencoders are scalable vision learners". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 16000–16009.

[96] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778.

[97] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 1026–1034.

[98] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. "DEBERTA: DECODING-ENHANCED BERT WITH DISENTANGLED ATTENTION". In: *International Conference on Learning Representations*. 2021.

[99] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. "DEBERTA: DECODING-ENHANCED BERT WITH DISENTANGLED ATTENTION". In: *ICLR 2021: The Ninth International Conference on Learning Representations*. 2021.

[100] Ruining He, Anirudh Ravula, Bhargav Kanagal, and Joshua Ainslie. "Realformer: Transformer likes residual attention". In: *arXiv preprint arXiv:2012.11747* (2020).

[101] Matthias Hein, Simon Setzer, Leonardo Jost, and Syama Sundar Rangapuram. "The total variation on hypergraphs-learning on hypergraphs revisited". In: *Advances in Neural Information Processing Systems* 26 (2013).

[102] Dan Hendrycks and Kevin Gimpel. “Gaussian error linear units (gelus)”. In: *arXiv preprint arXiv:1606.08415* (2016).

[103] Felix Hensel, Michael Moor, and Bastian Rieck. “A Survey of Topological Machine Learning Methods.” In: *Frontiers in artificial intelligence* 4 (2021), p. 681108.

[104] John Hewitt, Christopher D Manning, and Percy Liang. “Truncation sampling as language model desmoothing”. In: *Findings of the Association for Computational Linguistics: EMNLP* (2022).

[105] Geoffrey Hinton. “Distilling the Knowledge in a Neural Network”. In: *arXiv preprint arXiv:1503.02531* (2015).

[106] Christoph Hofer, Florian Graf, Bastian Rieck, Marc Niethammer, and Roland Kwitt. “Graph filtration learning”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 4314–4323.

[107] Christoph D Hofer, Roland Kwitt, and Marc Niethammer. “Learning representations of persistence barcodes.” In: *Journal of Machine Learning Research* 20.126 (2019), pp. 1–45.

[108] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. “The curious case of neural text degeneration”. In: *The curious case of neural text degeneration* (2020).

[109] Max Horn, Edward De Brouwer, Michael Moor, Yves Moreau, Bastian Rieck, and Karsten Borgwardt. “Topological Graph Neural Networks”. In: *International Conference on Learning Representations*. 2021.

[110] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. “Deep networks with stochastic depth”. In: *European Conference on Computer Vision (ECCV)*. 2016.

[111] Linjiang Huang, Yan Huang, Wanli Ouyang, and Liang Wang. “Part-level graph convolutional network for skeleton-based action recognition”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2020, pp. 11045–11052.

[112] Xiaohu Huang, Hao Zhou, Jian Wang, Haocheng Feng, Junyu Han, Errui Ding, Jingdong Wang, Xinggang Wang, Wenyu Liu, and Bin Feng. “Graph Contrastive Learning for Skeleton-based Action Recognition”. In: *The Eleventh International Conference on Learning Representations*. 2023.

[113] Niall Hurley and Scott Rickard. “Comparing measures of sparsity”. In: *IEEE Transactions on Information Theory* (2009).

[114] Umar Islambekov and Hasani Pathirana. “Vector Summaries of Persistence Diagrams for Permutation-based Hypothesis Testing”. In: *arXiv preprint arXiv:2306.06257* (2023).

[115] Masha Itkina, Boris Ivanovic, Ransalu Senanayake, Mykel J Kochenderfer, and Marco Pavone. “Evidential sparsification of multimodal latent spaces in conditional variational autoencoders”. In: *Advances in Neural Information Processing Systems (NeurIPS)* (2020).

[116] Eric Jang, Shixiang Gu, and Ben Poole. “Categorical reparameterization with gumbel-softmax”. In: *arXiv preprint arXiv:1611.01144* (2016).

[117] Robin Jia and Percy Liang. “Adversarial examples for evaluating reading comprehension systems”. In: *arXiv preprint arXiv:1707.07328* (2017).

[118] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. "Mistral 7B". In: *arXiv preprint arXiv:2310.06825* (2023).

[119] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. "Mixtral of experts". In: *arXiv preprint arXiv:2401.04088* (2024).

[120] Zihang Jiang, Qibin Hou, Li Yuan, Daquan Zhou, Yujun Shi, Xiaojie Jin, Anran Wang, and Jiashi Feng. "All Tokens Matter: Token Labeling for Training Better Vision Transformers." In: *arXiv preprint arXiv:2104.10858* (2021).

[121] Dan Jurafsky. *Speech & language processing.* Pearson Education India, 2000.

[122] Qiuhong Ke, Mohammed Bennamoun, Senjian An, Ferdous Sohel, and Farid Boussaid. "A New Representation of Skeleton Sequences for 3D Action Recognition". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 2017, pp. 4570–4579.

[123] Eun-Sol Kim, Woo Young Kang, Kyoung-Woon On, Yu-Jung Heo, and Byoung-Tak Zhang. "Hypergraph attention networks for multimodal learning". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* 2020, pp. 14581–14590.

[124] Yoon Kim, Carl Denton, Luong Hoang, and Alexander M Rush. "Structured attention networks". In: *International Conference on Learning Representations (ICLR)* (2017).

[125] Thomas N. Kipf and Max Welling. "Semi-Supervised Classification with Graph Convolutional Networks". In: *International Conference on Learning Representations.* 2016.

[126] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. "Segment anything". In: *Proceedings of the IEEE/CVF international conference on computer vision.* 2023, pp. 4015–4026.

[127] Simon Kornblith, Ting Chen, Honglak Lee, and Mohammad Norouzi. "Why do better loss functions lead to less transferable features?" In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 28648–28662.

[128] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25 (2012).

[129] Anders Krogh and John Hertz. "A simple weight decay can improve generalization". In: *Advances in neural information processing systems* 4 (1991).

[130] Genki Kusano, Yasuaki Hiraoka, and Kenji Fukumizu. "Persistence weighted Gaussian kernel for topological data analysis". In: *International Conference on Machine Learning.* PMLR. 2016, pp. 2004–2013.

[131] Anirban Laha, Saneem Ahmed Chemmengath, Priyanka Agrawal, Mitesh Khapra, Karthik Sankaranarayanan, and Harish G Ramaswamy. “On controllable sparse alternatives to softmax”. In: *Advances in Neural Information Processing Systems (NeurIPS)* (2018).

[132] Yann LeCun. “The MNIST database of handwritten digits”. In: *http://yann. lecun. com/exdb/mnist/* (1998).

[133] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *nature* 521.7553 (2015), pp. 436–444.

[134] Jungho Lee, Minhyeok Lee, Dogyoon Lee, and Sangyoun Lee. “Hierarchically decomposed graph convolutional networks for skeleton-based action recognition”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 10444–10453.

[135] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. “BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault. Online: Association for Computational Linguistics, July 2020, pp. 7871–7880. DOI: `10.18653/v1/2020.acl-main.703`. URL: `https://aclanthology.org/2020.acl-main.703/`.

[136] Bofang Li, Zhe Zhao, Tao Liu, Puwei Wang, and Xiaoyong Du. “Weighted neural bag-of-n-grams model: New baselines for text classification”. In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. 2016, pp. 1591–1600.

[137] Chunyuan Li, Maks Ovsjanikov, and Frederic Chazal. “Persistence-based structural recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1995–2002.

[138] Hao Li, Jinfa Huang, Peng Jin, Guoli Song, Qi Wu, and Jie Chen. “Toward 3d spatial reasoning for human-like text-based visual question answering”. In: *arXiv preprint arXiv:2209.10326* (2022).

[139] Hao Li, Jinfa Huang, Peng Jin, Guoli Song, Qi Wu, and Jie Chen. “Weakly-Supervised 3D Spatial Reasoning for Text-based Visual Question Answering”. In: *IEEE Transactions on Image Processing* (2023).

[140] Pan Li and Olgica Milenkovic. “Inhomogeneous hypergraph clustering with applications”. In: *Advances in Neural Information Processing Systems* 30 (2017).

[141] Jiajun Liang, Linze Li, Zhaodong Bing, Borui Zhao, Yao Tang, Bo Lin, and Haoqiang Fan. “Efficient one pass self-distillation with zipf’s label smoothing”. In: *European conference on computer vision*. Springer. 2022, pp. 104–119.

[142] Stephanie Lin, Jacob Hilton, and Owain Evans. “Truthfulqa: Measuring how models mimic human falsehoods”. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*. 2021.

[143] Hanbing Liu, Jun-Yan He, Zhi-Qi Cheng, Wangmeng Xiang, Qize Yang, Wenhao Chai, Gaoang Wang, Xu Bao, Bin Luo, Yifeng Geng, et al. "Posynda: Multi-hypothesis pose synthesis domain adaptation for robust 3d human pose estimation". In: *Proceedings of the 31st ACM International Conference on Multimedia*. 2023, pp. 5542–5551.

[144] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C Kot. "Ntu RGB+D 120: A large-scale benchmark for 3d human activity understanding". In: *IEEE transactions on pattern analysis and machine intelligence* 42.10 (2019), pp. 2684–2701.

[145] Mengyuan Liu, Hong Liu, and Chen Chen. "Enhanced skeleton visualization for view invariant human action recognition". In: *Pattern Recognition* 68.68 (2017), pp. 346–362.

[146] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. "RoBERTa: A Robustly Optimized BERT Pretraining Approach". In: *arXiv preprint arXiv:1907.11692* (2019).

[147] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. "Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 10012–10022.

[148] Zhe Liu, Xiang Huang, Jingyun Zhang, Zhifeng Hao, Li Sun, and Hao Peng. "Multivariate time-series anomaly detection based on enhancing graph attention networks with topological analysis". In: *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 2024, pp. 1555–1564.

[149] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. "Disentangling and unifying graph convolutions for skeleton-based action recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 143–152.

[150] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. "Rectifier nonlinearities improve neural network acoustic models". In: *International Conference on Machine Learning (ICML)*. 2013.

[151] Matt Mahoney. *Large text compression benchmark*. 2011.

[152] Andre Martins and Ramon Astudillo. "From Softmax to Sparsemax A Sparse Model of Attention and MultiLabel Classification". In: *International Conference on Machine Learning (ICML)* (2016).

[153] Sameen Maruf, André FT Martins, and Gholamreza Haffari. "Selective attention for context-aware neural machine translation". In: *arXiv preprint arXiv:1903.08788* (2019).

[154] Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. "Locally typical sampling". In: *Transactions of the Association for Computational Linguistics* 11 (2023), pp. 102–121.

[155] Clara Meister, Gian Wiher, Tiago Pimentel, and Ryan Cotterell. "On the probability-quality paradox in language generation". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*. 2022.

[156] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. "Pointer sentinel mixture models". In: *International Conference on Learning Representations (ICLR)* (2017).

[157] Yuriy Mileyko, Sayan Mukherjee, and John Harer. "Probability measures on the space of persistence diagrams". In: *Inverse Problems* 27.12 (2011), p. 124007.

[158] Chul Moon and Nicole A Lazar. "Hypothesis testing for shapes using vectorized persistence diagrams". In: *Journal of the Royal Statistical Society Series C: Applied Statistics* 72.3 (2023), pp. 628–648.

[159] Michael Moor, Max Horn, Bastian Rieck, and Karsten Borgwardt. "Topological autoencoders". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 7045–7054.

[160] Tendai Mukande, Esraa Ali, Annalina Caputo, Ruihai Dong, and Noel E O'Connor. "A flash attention transformer for multi-behaviour recommendation". In: *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 2023, pp. 4210–4214.

[161] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. "When does label smoothing help?" In: *Advances in neural information processing systems* 32 (2019).

[162] Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. "Abstractive text summarization using sequence-to-sequence rnns and beyond". In: *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*. 2016.

[163] Yatin Nandwani, Vineet Kumar, Dinesh Raghu, Sachindra Joshi, and Luis A Lastras. "Pointwise mutual information based metric and decoding strategy for faithful generation in document grounded dialogs". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 2023.

[164] Minh Nguyen, Andrew Baker, Clement Neo, Allen Roush, Andreas Kirsch, and Ravid Shwartz-Ziv. "Turning Up the Heat: Min-p Sampling for Creative and Coherent LLM Outputs". In: *arXiv preprint arXiv:2407.01082* (2024).

[165] Vlad Niculae and Mathieu Blondel. "A regularized framework for sparse and structured neural attention". In: *Advances in Neural Information Processing Systems (NeurIPS)* (2017).

[166] Zachary Novack, Julian McAuley, Zachary Chase Lipton, and Saurabh Garg. "Chils: Zero-shot image classification with hierarchical label sets". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 26342–26362.

[167] Kenta Oono and Taiji Suzuki. "Graph neural networks exponentially lose expressive power for node classification". In: *arXiv preprint arXiv:1905.10947* (2019).

[168] OpenAI. *OpenAI o1 System Card*. https://arxiv.org/abs/2412.16720. Accessed: 2025-05-12. 2024.

[169] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. "Continual lifelong learning with neural networks: A review". In: *Neural Networks* 113 (2019), pp. 54–71.

[170] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. "Automatic differentiation in pytorch". In: (2017).

[171] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems*. Vol. 32. 2019, pp. 8026–8037.

[172] Wei Peng, Xiaopeng Hong, Haoyu Chen, and Guoying Zhao. "Learning graph convolutional network for skeleton-based human action recognition by neural searching". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 03. 2020, pp. 2669–2676.

[173] Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. "Regularizing neural networks by penalizing confident output distributions". In: *arXiv preprint arXiv:1701.06548* (2017).

[174] Ben Peters, Vlad Niculae, and André FT Martins. "Sparse sequence-to-sequence models". In: *arXiv preprint arXiv:1905.05702* (2019).

[175] Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. "Mauve: Measuring the gap between neural text and human text using divergence frontiers". In: *Advances in Neural Information Processing Systems (NeurIPS)* 34 (2021), pp. 4816–4828.

[176] Chiara Plizzari, Marco Cannici, and Matteo Matteucci. "Spatial temporal transformer network for skeleton-based action recognition". In: *International Conference on Pattern Recognition Workshops and Challenges*. Springer. 2021, pp. 694–701.

[177] Matt Post. "A call for clarity in reporting BLEU scores". In: *arXiv preprint arXiv:1804.08771* (2018).

[178] Katharina Prasse, Steffen Jung, Yuxuan Zhou, and Margret Keuper. "Local Spherical Harmonics Improve Skeleton-Based Hand Action Recognition". In: *arXiv preprint arXiv:2308.10557* (2023).

[179] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. "Learning transferable visual models from natural language supervision". In: *International conference on machine learning*. PmLR. 2021, pp. 8748–8763.

[180] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. "Improving language understanding by generative pre-training". In: (2018).

[181] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8 (2019), p. 9.

[182] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. "Exploring the limits of transfer learning with a unified text-to-text transformer". In: *Journal of machine learning research* 21.140 (2020), pp. 1–67.

[183] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. "Stand-alone self-attention in vision models". In: *Advances in Neural Information Processing Systems* 32 (2019).

[184] Dominick Reilly and Srijan Das. “Just add?! pose induced video transformers for understanding activities of daily living”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 18340–18350.

[185] Jan Reininghaus, Stefan Huber, Ulrich Bauer, and Roland Kwitt. “A stable multi-scale kernel for topological machine learning”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 4741–4748.

[186] Bastian Rieck, Christian Bock, and Karsten Borgwardt. “A persistent weisfeiler-lehman procedure for graph classification”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 5448–5458.

[187] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. “Dropedge: Towards deep graph convolutional networks on node classification”. In: *arXiv preprint arXiv:1907.10903* (2019).

[188] Gavin A Rummery and Mahesan Niranjan. *On-line Q-learning using connectionist systems*. University of Cambridge, Department of Engineering Cambridge, UK, 1994.

[189] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. “Imagenet large scale visual recognition challenge”. In: *International journal of computer vision* 115 (2015), pp. 211–252.

[190] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. “Mobilenetv2: Inverted residuals and linear bottlenecks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4510–4520.

[191] Mert Bulent Sariyildiz, Yannis Kalantidis, Karteek Alahari, and Diane Larlus. “No reason for no supervision: Improved generalization in supervised models”. In: *arXiv preprint arXiv:2206.15369* (2022).

[192] Robin Schirrmeister, Yuxuan Zhou, Tonio Ball, and Dan Zhang. “Understanding anomaly detection with deep invertible networks through hierarchies of distributions and features”. In: *Advances in Neural Information Processing Systems (NeurIPS)* (2020).

[193] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. “Modeling relational data with graph convolutional networks”. In: *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15*. Springer. 2018, pp. 593–607.

[194] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. “Proximal policy optimization algorithms”. In: *arXiv preprint arXiv:1707.06347* (2017).

[195] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. “Grad-cam: Visual explanations from deep networks via gradient-based localization”. In: *International Conference on Machine Learning (ICML)* (2017).

[196] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization". In: *International Journal of Computer Vision* 128.2 (Oct. 2019), pp. 336–359. ISSN: 1573-1405. URL: http://dx.doi.org/10.1007/s11263-019-01228-7.

[197] Lee M Seversky, Shelby Davis, and Matthew Berger. "On time-series topological data analysis: New data and opportunities". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition Workshops*. 2016, pp. 59–67.

[198] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. "Ntu rgb+ d: A large scale dataset for 3d human activity analysis". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 1010–1019.

[199] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. "Deepseekmath: Pushing the limits of mathematical reasoning in open language models". In: *arXiv preprint arXiv:2402.03300* (2024).

[200] Ehsan Shareghi, Daniela Gerz, Ivan Vulic, et al. "Show some love to your n-grams: A bit of progress and stronger n-gram language modeling baselines". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2019.

[201] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. "Self-Attention with Relative Position Representations". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. Vol. 2. 2018, pp. 464–468.

[202] Noam Shazeer, Zhenzhong Lan, Youlong Cheng, Nan Ding, and Le Hou. "Talking-heads attention". In: *arXiv preprint arXiv:2003.02436* (2020).

[203] Zhiqiang Shen, Zechun Liu, Dejia Xu, Zitian Chen, Kwang-Ting Cheng, and Marios Savvides. "Is label smoothing truly incompatible with knowledge distillation: An empirical study". In: *arXiv preprint arXiv:2104.00676* (2021).

[204] Chufan Shi, Haoran Yang, Deng Cai, Zhisong Zhang, Yifan Wang, Yujiu Yang, and Wai Lam. "A thorough examination of decoding methods in the era of llms". In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 2024.

[205] Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. "Large language models can be easily distracted by irrelevant context". In: (2023).

[206] Han Shi, Jiahui Gao, Hang Xu, Xiaodan Liang, Zhenguo Li, Lingpeng Kong, Stephen Lee, and James T Kwok. "Revisiting over-smoothing in BERT from the perspective of graph". In: *arXiv preprint arXiv:2202.08625* (2022).

[207] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. "Decoupled spatial-temporal attention network for skeleton-based action recognition". In: *arXiv preprint arXiv:2007.03263* (2020).

[208] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. "Two-Stream Adaptive Graph Convolutional Networks for Skeleton-Based Action Recognition". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 12026–12035.

[209] Carlos N Silla and Alex A Freitas. “A survey of hierarchical classification across different application domains”. In: *Data mining and knowledge discovery* 22 (2011), pp. 31–72.

[210] Yi-Fan Song, Zhang Zhang, Caifeng Shan, and Liang Wang. “Constructing Stronger and Faster Baselines for Skeleton-based Action Recognition”. In: *arXiv preprint arXiv:2106.15125* (2021).

[211] Yi-Fan Song, Zhang Zhang, Caifeng Shan, and Liang Wang. “Stronger, faster and more explainable: A graph convolutional baseline for skeleton-based action recognition”. In: *Proceedings of the 28th ACM International Conference on Multimedia*. 2020, pp. 1625–1633.

[212] Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. “An end-to-end spatio-temporal attention model for human action recognition from skeleton data”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 31. 2017.

[213] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research*. Vol. 15. 2014, pp. 1929–1958.

[214] Peter Súkeník, Marco Mondelli, and Christoph Lampert. “Neural Collapse versus Low-rank Bias: Is Deep Neural Collapse Really Optimal?” In: *arXiv preprint arXiv:2405.14468* (2024).

[215] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[216] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. “Rethinking the inception architecture for computer vision”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2818–2826.

[217] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. “Gemini: a family of highly capable multimodal models”. In: *arXiv preprint arXiv:2312.11805* (2023).

[218] Kalpit Thakkar and PJ Narayanan. “Part-based graph convolutional network for action recognition”. In: *arXiv preprint arXiv:1809.04983* (2018).

[219] Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. “MLP-Mixer: An all-MLP Architecture for Vision”. In: *arXiv preprint arXiv:2105.01601* (2021).

[220] Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, and Hervé Jégou. “ResMLP: Feedforward networks for image classification with data-efficient training.” In: *arXiv preprint arXiv:2105.03404* (2021).

[221] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. “Training data-efficient image transformers & distillation through attention”. In: *International conference on machine learning*. PMLR. 2021, pp. 10347–10357.

[222] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. “Going deeper with Image Transformers”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 32–42.

[223] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. “Llama 2: Open foundation and fine-tuned chat models”. In: *arXiv preprint arXiv:2307.09288* (2023).

[224] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017).

[225] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. “Graph attention networks”. In: *arXiv preprint arXiv:1710.10903* (2017).

[226] Hang Wang, Youtian Du, Yabin Zhang, Shuai Li, and Lei Zhang. “One-Stage Visual Relationship Referring With Transformers and Adaptive Message Passing”. In: *IEEE Transactions on Image Processing* (2022).

[227] Jiang Wang, Xiaohan Nie, Yin Xia, Ying Wu, and Song-Chun Zhu. “Cross-view action modeling, learning and recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 2649–2656.

[228] Lei Wang and Piotr Koniusz. “3Mformer: Multi-order Multi-mode Transformer for Skeletal Action Recognition”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 5620–5631.

[229] Peihao Wang, Wenqing Zheng, Tianlong Chen, and Zhangyang Wang. “Anti-oversmoothing in deep vision transformers via the fourier domain analysis: From theory to practice”. In: *arXiv preprint arXiv:2203.05962* (2022).

[230] Pichao Wang, Xue Wang, Fan Wang, Ming Lin, Shuning Chang, Hao Li, and Rong Jin. “Kvt: k-nn attention for boosting vision transformers”. In: *European Conference on Computer Vision (ECCV)*. 2022.

[231] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. “Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions”. In: *arXiv preprint arXiv:2102.12122* (2021).

[232] X Wang, J Wei, D Schuurmans, Q Le, E Chi, S Narang, A Chowdhery, and D Zhou. “Self-consistency improves chain of thought reasoning in language models”. In: *International Conference on Learning Representations (ICLR)* (2023).

[233] Xuezhi Wang and Denny Zhou. “Chain-of-thought reasoning without prompting”. In: *Advances in Neural Information Processing Systems (NeurIPS)* (2024).

[234] Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. “Neural text generation with unlikelihood training”. In: *International Conference on Learning Representations (ICLR)* (2020).

[235] Jason Weston and Sainbayar Sukhbaatar. “System 2 Attention (is something you might need too)”. In: *arXiv preprint arXiv:2311.11829* (2023).

[236] David A White and Donald A Sofge. “The role of exploration in learning control”. In: *Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches* (1992).

[237] Ronald J Williams. “Simple statistical gradient-following algorithms for connectionist reinforcement learning”. In: *Machine learning* (1992).

[238] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. “Huggingface’s transformers: State-of-the-art natural language processing”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations.* 2020.

[239] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. “CvT: Introducing Convolutions to Vision Transformers.” In: *arXiv preprint arXiv:2103.15808* (2021).

[240] Kan Wu, Houwen Peng, Minghao Chen, Jianlong Fu, and Hongyang Chao. “Rethinking and improving relative position encoding for vision transformer”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 2021, pp. 10033–10041.

[241] Guoxuan Xia, Olivier Laurent, Gianni Franchi, and Christos-Savvas Bouganis. “Understanding Why Label Smoothing Degrades Selective Classification and How to Fix It”. In: *arXiv preprint arXiv:2403.14715* (2024).

[242] Hailun Xia and Xinkai Gao. “Multi-Scale Mixed Dense Graph Convolution Network for Skeleton-Based Action Recognition”. In: *IEEE Access* 9 (2021), pp. 36475–36484.

[243] Wangmeng Xiang, Chao Li, Yuxuan Zhou, Biao Wang, and Lei Zhang. “Language supervised training for skeleton-based action recognition”. In: *arXiv preprint arXiv:2208.05318* (2022).

[244] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. “Unified perceptual parsing for scene understanding”. In: *Proceedings of the European conference on computer vision (ECCV).* 2018, pp. 418–434.

[245] Jing Xu and Haoxiong Liu. “Quantifying the variability collapse of neural networks”. In: *International Conference on Machine Learning.* PMLR. 2023, pp. 38535–38550.

[246] Kailin Xu, Fanfan Ye, Qiaoyong Zhong, and Di Xie. “Topology-aware Convolutional Neural Network for Efficient Skeleton-based Action Recognition”. In: *arXiv preprint arXiv:2112.04178* (2021).

[247] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. “How Powerful are Graph Neural Networks”. In: *International Conference on Learning Representations.* 2018.

[248] Naganand Yadati, Madhav Nimishakavi, Prateek Yadav, Vikram Nitin, Anand Louis, and Partha Talukdar. “Hypergcn: A new method for training graph convolutional networks on hypergraphs”. In: *Advances in neural information processing systems* 32 (2019).

[249] Sijie Yan, Yuanjun Xiong, and Dahua Lin. “Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition”. In: *AAAI.* 2018, pp. 7444–7452.

[250] Zuoyu Yan, Tengfei Ma, Liangcai Gao, Zhi Tang, Yusu Wang, and Chao Chen. “Neural approximation of graph topological features”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 33357–33370.

[251] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. "XLNet: Generalized Autoregressive Pretraining for Language Understanding". In: *Advances in Neural Information Processing Systems.* Vol. 32. 2019, pp. 5753–5763.

[252] Fanfan Ye, Shiliang Pu, Qiaoyong Zhong, Chao Li, Di Xie, and Huiming Tang. "Dynamic GCN: Context-enriched Topology Learning for Skeleton-based Action Recognition". In: *Proceedings of the 28th ACM International Conference on Multimedia.* 2020, pp. 55–63.

[253] Kai Yi, Xiaoqian Shen, Yunhao Gou, and Mohamed Elhoseiny. "Exploring hierarchical graph representation for large-scale zero-shot image classification". In: *European Conference on Computer Vision.* Springer. 2022, pp. 116–132.

[254] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. "Do transformers really perform badly for graph representation?" In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 28877–28888.

[255] Kun Yuan, Shaopeng Guo, Ziwei Liu, Aojun Zhou, Fengwei Yu, and Wei Wu. "Incorporating convolution designs into visual transformers". In: *Proceedings of the IEEE/CVF international conference on computer vision.* 2021, pp. 579–588.

[256] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. "Tokens-to-token vit: Training vision transformers from scratch on imagenet". In: *Proceedings of the IEEE/CVF international conference on computer vision.* 2021, pp. 558–567.

[257] Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. "Revisiting knowledge distillation via label smoothing regularization". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* 2020, pp. 3903–3911.

[258] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. "Graph transformer networks". In: *Advances in neural information processing systems* 32 (2019).

[259] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. "Deep sets". In: *Advances in Neural Information Processing Systems* 30 (2017).

[260] Matthew D Zeiler and Rob Fergus. "Visualizing and understanding convolutional networks". In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13.* Springer. 2014, pp. 818–833.

[261] Sebastian Zeng, Florian Graf, Christoph Hofer, and Roland Kwitt. "Topological attention for time series forecasting". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 24871–24882.

[262] Biao Zhang, Ivan Titov, and Rico Sennrich. "On Sparsifying Encoder Outputs in Sequence-to-Sequence Models". In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021.* 2021, pp. 2888–2900.

[263] Chang-Bin Zhang, Peng-Tao Jiang, Qibin Hou, Yunchao Wei, Qi Han, Zhen Li, and Ming-Ming Cheng. "Delving deep into label smoothing". In: *IEEE Transactions on Image Processing* 30 (2021), pp. 5984–5996.

[264] Chenzi Zhang, Shuguang Hu, Zhihao Gavin Tang, and TH Hubert Chan. "Re-revisiting learning on hypergraphs: confidence interval and subgradient method". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 4026–4034.

[265] Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. "View adaptive neural networks for high performance skeleton-based human action recognition". In: *IEEE transactions on pattern analysis and machine intelligence* 41.8 (2019), pp. 1963–1978.

[266] Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. "View adaptive recurrent neural networks for high performance human action recognition from skeleton data". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 2117–2126.

[267] Pengfei Zhang, Cuiling Lan, Wenjun Zeng, Junliang Xing, Jianru Xue, and Nanning Zheng. "Semantics-guided neural networks for efficient skeleton-based human action recognition". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 1112–1121.

[268] Zhengyou Zhang. "Microsoft kinect sensor and its effect". In: *IEEE multimedia* 19.2 (2012), pp. 4–10.

[269] Zizhao Zhang, Han Zhang, Long Zhao, Ting Chen, Sercan Ö Arik, and Tomas Pfister. "Nested hierarchical transformer: Towards accurate, data-efficient and interpretable visual understanding". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 3. 2022, pp. 3417–3425.

[270] Guangxiang Zhao, Junyang Lin, Zhiyuan Zhang, Xuancheng Ren, Qi Su, and Xu Sun. "Explicit sparse transformer: Concentrated attention through explicit selection". In: *arXiv preprint arXiv:1912.11637* (2019).

[271] Qi Zhao and Yusu Wang. "Learning metrics for persistence-based summaries and applications for graph classification". In: *Advances in Neural Information Processing Systems* 32 (2019).

[272] Cheng Zheng, Bo Zong, Wei Cheng, Dongjin Song, Jingchao Ni, Wenchao Yu, Haifeng Chen, and Wei Wang. "Robust graph representation learning via neural sparsification". In: *International Conference on Machine Learning (ICML)* (2020).

[273] Daquan Zhou, Bingyi Kang, Xiaojie Jin, Linjie Yang, Xiaochen Lian, Qibin Hou, and Jiashi Feng. "DeepViT: Towards Deeper Vision Transformer." In: *arXiv preprint arXiv:2103.11886* (2021).

[274] Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. "Learning with hypergraphs: Clustering, classification, and embedding". In: *Advances in neural information processing systems* 19 (2006).

[275] Huanyu Zhou, Qingjie Liu, and Yunhong Wang. "Learning discriminative representations for skeleton based action recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 10608–10617.

[276] Jinxin Zhou, Chong You, Xiao Li, Kangning Liu, Sheng Liu, Qing Qu, and Zhihui Zhu. "Are all losses created equal: A neural collapse perspective". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 31697–31710.

[277] Yuxuan Zhou, Zhi-Qi Cheng, Jun-Yan He, Bin Luo, Yifeng Geng, Xuansong Xie, and Margret Keuper. “Overcoming Topology Agnosticism: Enhancing Skeleton-Based Action Recognition through Redefined Skeletal Topology Awareness”. In: *arXiv preprint arXiv:2305.11468* (2023).

[278] Yuxuan Zhou, Zhi-Qi Cheng, Chao Li, Yanwen Fang, Yifeng Geng, Xuansong Xie, and Margret Keuper. “Hypergraph transformer for skeleton-based action recognition”. In: *arXiv preprint arXiv:2211.09590* (2022).

[279] Yuxuan Zhou, Mario Fritz, and Margret Keuper. “MultiMax: sparse and multi-modal attention learning”. In: *Forty-first International Conference on Machine Learning* (2024).

[280] Yuxuan Zhou, Margret Keuper, and Mario Fritz. “Balancing Diversity and Risk in LLM Sampling: How to Select Your Method and Parameter for Open-Ended Text Generation”. In: *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*. 2025.

[281] Yuxuan Zhou, Heng Li, Zhi-Qi Cheng, Xudong Yan, Mario Fritz, and Margret Keuper. “MaxSup: Overcoming Representation Collapse in Label Smoothing”. In: *Proceedings of the Thirty-Ninth Annual Conference on Neural Information Processing Systems (Oral)*. 2025.

[282] Yuxuan Zhou, Wangmeng Xiang, Chao Li, Biao Wang, Xihan Wei, Lei Zhang, Margret Keuper, and Xiansheng Hua. “SP-ViT: Learning 2D Spatial Priors for Vision Transformers”. In: *33rd British Machine Vision Conference*. BMVA Press. 2022.

[283] Yuxuan Zhou, Xudong Yan, Zhi-Qi Cheng, Yan Yan, Qi Dai, and Xian-Sheng Hua. “Blockgcn: Redefine topology awareness for skeleton-based action recognition”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 2049–2058.

[284] Fei Zhu, Zhen Cheng, Xu-Yao Zhang, and Cheng-Lin Liu. “Rethinking confidence calibration for failure prediction”. In: *European Conference on Computer Vision*. Springer. 2022, pp. 518–536.

[285] Wenhong Zhu, Hongkun Hao, Zhiwei He, Yiming Ai, and Rui Wang. “Improving Open-Ended Text Generation via Adaptive Decoding”. In: *International Conference on Machine Learning (ICML)* (2024).

[286] Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. “Texygen: A benchmarking platform for text generation models”. In: *The 41st international ACM SIGIR conference on research & development in information retrieval*. 2018, pp. 1097–1100.

[287] Hui Zou and Trevor Hastie. “Regularization and variable selection via the elastic net”. In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 67.2 (2005), pp. 301–320.

Appendix

Appendix A.

Improving the Trade-Off between Sparsity and Multi-Modality in Attention

In this section, we provide the supplementary materials for Chapter III.

1. Lemmas

In the following, we provide the lemmas which will be used for our later proofs.

Lemma 1.1. *The following inequalities hold:*

$$\begin{aligned}
x_i &> tx_i + (1-t)b, && \forall\, x_i < b \quad \textit{and} \quad \forall\, t > 1\\
x_i &< tx_i + (1-t)b, && \forall\, x_i > b \quad \textit{and} \quad \forall\, t > 1\\
x_i &< tx_i + (1-t)b, && \forall\, x_i < b \quad \textit{and} \quad \forall\, t < 1\\
x_i &> tx_i + (1-t)b, && \forall\, x_i > b \quad \textit{and} \quad \forall\, t < 1
\end{aligned}$$

(See Appendix 2 for the proof.)

Lemma 1.2. *The following inequality holds* $\forall\, \epsilon \leq \frac{1}{L}(\sum_{x_l<b}^{L} X_l - lnL)$ *and* $\forall\, t > 1$*:*

$$\sum_{x_l<b}^{L} e^{t(x_l-x_i)} \geq \sum_{x_l<b}^{L} e^{x_l-x_i}$$

(See Appendix 2 for the proof.)

2. Proofs

In the following, we provide the proofs for all the lemmas and propositions in our paper.

Proof of Lemma 3.4

$\dfrac{\partial \mathcal{S}(\boldsymbol{x})}{\partial \phi(\boldsymbol{x})_l} = -\dfrac{1}{s} \exp\,(\dfrac{s-\phi(\boldsymbol{x})_l}{s} - 1)$. $\forall\, s > 0 \Rightarrow -\dfrac{1}{s} < 0$. Since the exponential term is always positive, we have $\dfrac{\partial \mathcal{S}(\boldsymbol{x})}{\partial \phi(\boldsymbol{x})_l} < 0, \quad \forall\, \phi(\boldsymbol{x})_l$.

Proof of Proposition 3.5

Proof. Statement 1

from Eq. (III.1) and Definition 3.2

$$\frac{\partial \mathcal{M}_(\boldsymbol{x})}{\partial t} = \frac{(x_{max} - x_n)e^{\frac{x_n - x_{max}}{t}}}{t^2 \sum_{k=1}^{K} e^{\frac{x_k - x_{max}}{t}}} + \frac{(1 - e^{\frac{x_n - x_{max}}{t}}) \sum_{k=1}^{K} \frac{x_{max} - x_k}{t^2} e^{\frac{x_k - x_{max}}{t}}}{(\sum_{k=1}^{K} e^{\frac{x_k - x_{max}}{t}})^2}$$

since $x_n - x_{max} < 0$, we have $0 < e^{\frac{x_n - x_{max}}{t}} < 1$

$$\Rightarrow \quad \frac{\partial \mathcal{M}(\boldsymbol{x})}{\partial t} > 0 \quad \text{holds} \quad \forall\, t$$

□

Proof. Statement 2

from Eq. (III.1)

$$\frac{\partial \phi_(\boldsymbol{x})_l}{\partial t} = \frac{\sum_{k=1}^{K} (x_k - x_l) e^{\frac{x_k - x_l}{t}}}{t^2 (\sum_{k=1}^{K} e^{\frac{x_k - x_l}{t}})^2}$$

from Chebyshev's sum inequality

$$\sum_{k=1}^{K} (x_k - x_l) e^{\frac{x_k - x_l}{t}} > \frac{1}{K} \sum_{k=1}^{K} (x_k - x_l) \sum_{k=1}^{K} e^{\frac{x_k - x_l}{t}}$$

since $x_l < \epsilon \leq \frac{\|\boldsymbol{x}\|_1}{K}$, we have $\sum_{k=1}^{K} (x_k - x_l) \geq 0$

$$\Rightarrow \quad \frac{\partial \phi_(\boldsymbol{x})_l}{\partial t} > 0$$

from Lemma 3.4

$$\Rightarrow \quad \frac{\partial \mathcal{S}(\boldsymbol{x})}{\partial t} = \frac{\partial \mathcal{S}(\boldsymbol{x})}{\partial \phi_(\boldsymbol{x})_l} \frac{\partial \phi_(\boldsymbol{x})_l}{\partial t} < 0$$

□

Proof of Lemma 1.1

From basic laws of algebra, $x - tx - (1 - t)b = (1 - t)(x - b)$. For $t > 1$ and $x < b$, we have $(1 - t)(x - b) > 0 \Rightarrow x > tx + (1 - t)b$, and vice versa.

Proof of Lemma 1.2

since $e^{x_l} > 0$, from Hoelder's inequality, we have

$$\begin{aligned}
&\sum_{x_l<b}^{L} e^{x_l-x_i} = \sum_{x_l<b}^{L} \left|e^{x_l-x_i}\right|^1 \cdot 1 \\
&\le \sum_{x_l<b}^{L} \Big(\big(\Big|e^{x_l-x_i}\big)\Big|\Big)^t\big)^{\frac{1}{t}} \cdot \left(\sum_{l=1}^{L} 1^{\frac{t}{t-1}}\right)^{1-\frac{1}{t}}
\end{aligned}$$

raise both sides to the power of t and multiply by L^{1-t}

$$\Rightarrow L^{1-t}(\sum_{x_l<b}^{L} e^{(x_l-x_i)})^t \le \sum_{x_l<b}^{L} e^{t(x_l-x_i)}$$

the above inequality holds if

$$\sum_{x_l<b}^{L} e^{x_l-x_i} \le L^{1-t}(\sum_{x_l<b}^{L} e^{(x_l-x_i)})^t$$

take the natural log on both sides

$$\begin{aligned}
&ln \sum_{x_l<b}^{L} e^{x_l-x_i} \le (1-t)lnL + tln \sum_{x_l<b}^{L} e^{x_l-x_i} \\
&\Rightarrow lnL \le ln \sum_{x_l<b}^{L} e^{(x_l-x_i)}
\end{aligned}$$

since e^x is convex and $x_i < \epsilon$

$$\sum_{x_l<b}^{L} e^{(x_l-x_i)} \ge e^{\sum_{x_l<b}^{L}(x_l-x_i)} \ge e^{\sum_{x_l<b}^{L}(x_l-\epsilon)}$$

the condition is satisfied for $\epsilon \le \frac{1}{L}(\sum_{x_l<b}^{L} x_l - lnL)$

Proof of Proposition 4.2

When only term (1) is considered, Eq. (III.5) is reduced to:

$$\sigma(x) = \begin{cases} t_b x + (1-t_b)b & x < b \\ x & x \ge b \end{cases}, \tag{A.1}$$

and we obtain:

$$\phi_{MultiMax\text{-}l}(\boldsymbol{x})_i = \begin{cases} \dfrac{e^{t_b x_l + (1-t_b)b}}{\sum_{x_l<b}^{L} e^{t_b x_l + (1-t_b)b} + \sum_{x_n \ge b}^{N} e^{x_n}} & x_l < b \\ \dfrac{e^{x_n}}{\sum_{x_l<b}^{L} e^{t_b x_l + (1-t_b)b} + \sum_{x_n \ge b}^{N} e^{x_n}} & x_l \ge b \end{cases}, \tag{A.2}$$

where L and N denote the number of entries smaller than or greater than b and $L+N = K$.

Proof. Statement 1

from Eq. (A.2), $\forall\, x_i < \epsilon \leq b$, eliminate the numerator

$$\phi_{\textit{MultiMax-l}}(\boldsymbol{x})_i = \frac{1}{\sum\limits_{x_l<b}^{L} e^{t_b(x_l - x_i)} + \sum\limits_{x_n \geq b}^{N} e^{x_n - (t_b x_i + (1-t_b)b)}}$$

substitute $t_b x_i + (1-t_b)b$ with x_i at lower right and $\sum\limits_{x_l<b}^{L} e^{t_b(x_l-x_i)}$ at lower left, from Lemma 1.1 and Lemma 1.2

$$\leq \frac{1}{\sum\limits_{x_l<b}^{L} e^{x_l - x_i)} + \sum\limits_{x_n \geq b}^{N} e^{x_n - x_i}}$$

$$\Rightarrow \phi_{\textit{MultiMax-l}}(\boldsymbol{x})_i < \phi_{\textit{SoftMax}}(\boldsymbol{x})_i$$ □

Proof. Statement 2

Eliminate e^{x_i}, from Eq. (A.2), $\forall\, x_i > x_j > b$

$$m_{\textit{MultiMax-l}} = 1 - \frac{1 - e^{(x_j - x_i)}}{\sum\limits_{x_l<b}^{L} e^{t_b x_l + (1-t_b)b - x_i} + \sum\limits_{x_n \geq b}^{N} e^{(x_n - x_i)}}$$

substitute $(1-t_b)b - x_i$ with $-t_b x_i$, from Lemma 1.1

$$> 1 - \frac{1 - e^{(x_j - x_i)}}{\sum\limits_{x_l<b}^{L} e^{t_b(x_l - x_i)} + \sum\limits_{x_n \geq b}^{N} e^{(x_n - x_i)}}$$

substitute $\sum\limits_{x_l<b}^{L} e^{t_b(x_l-x_i)}$ with $\sum\limits_{x_l<b}^{L} e^{x_l - x_i}$, from Lemma 1.2 $\quad \forall\, \epsilon \leq \frac{1}{L}(\sum\limits_{x_l<b}^{L} x_l - lnL)$

$$> 1 - \frac{1 - e^{(x_j - x_i)}}{\sum\limits_{x_l<b}^{L} e^{x_l - x_i} + \sum\limits_{x_n \geq b}^{N} e^{(x_n - x_i)}} = \mathcal{M}_{\textit{SoftMax}}$$ □

Proof of Proposition 4.3

Combine Eq. (III.5) with SoftMax, we obtain:

$$\phi_{\scriptscriptstyle MultiMax}(\boldsymbol{x})_i = \begin{cases} \dfrac{e^{t_b x_i + (1-t_b)b}}{\sum\limits_{x_l<b}^{L} e^{\sigma(x_l)} + \sum\limits_{b \leq x_m \leq d}^{M} e^{x_m} + \sum\limits_{x_n>d}^{N} e^{\sigma(x_n)}} & x_i < b \\ \dfrac{e^{x_i}}{\sum\limits_{x_l<b}^{L} e^{\sigma(x_l)} + \sum\limits_{b \leq x_m \leq d}^{M} e^{x_m} + \sum\limits_{x_n>d}^{N} e^{\sigma(x_n)}} & b \leq x_i \leq d \\ \dfrac{e^{t_d x_i + (1-t_d)d}}{\sum\limits_{x_l<b}^{L} e^{\sigma(x_l)} + \sum\limits_{b \leq x_m \leq d}^{M} e^{x_m} + \sum\limits_{x_n>d}^{N} e^{\sigma(x_n)}} & x_i > d \end{cases}, \tag{A.3}$$

where L, M and N denote the number of entries belonging to different ranges and $L + M + N = K$.

Proof. Statement 1

from Eq. (A.3), $\forall\, x_i < \epsilon$, eliminate the numerator, then substitute $x_i + (1 - t_b)b$ with $t_b x_i$, from Lemma 1.1

$$< 1/(\sum_{x_l<b}^{L} e^{t_b(x_l - x_i)} + \sum_{b \leq x_m \leq d}^{M} e^{x_m - x_i} + \sum_{x_n>d}^{N} e^{t_d x_n + (1-t_d)d - t_b x_i - (1-t_b)b})$$

from Lemma 1.2, if $\epsilon \leq \frac{1}{M}(\sum\limits_{x_m<b}^{M} X_m - lnM)$

$$< 1/(\sum_{x_l<b}^{L} e^{x_l - x_i} + \sum_{b \leq x_m \leq d}^{M} e^{x_m - x_i} + \sum_{x_n>d}^{N} e^{t_d x_n + (1-t_d)d - t_b x_i - (1-t_b)b})$$

if $\sum\limits_{x_n>d}^{N} e^{t_d x_n + (1-t_d)d - t_b x_i - (1-t_b)b} > \sum\limits_{x_n>d}^{N} e^{x_n - x_i}$

$$\Rightarrow \phi_{MultiMax}(\boldsymbol{x})_i < \phi_{SoftMax}(\boldsymbol{x})_i$$

This is satisfied when $t_d x_n + (1 - t_d)d - t_b x_i - (1 - t_b)b > x_n - x_i$ holds $\forall\, x_n$, which can be reduced to

$$x_i < b - \frac{1 - t_d}{t_b - 1}(x_n - d)$$

where $x_n \geq d$, $t_d < 1$ and $t_b > 1$, and this is satisfied for

$$\Rightarrow \epsilon \leq b - \frac{1 - t_d}{t_b - 1}(x_n - d) \qquad \square$$

Proof. Statement 2

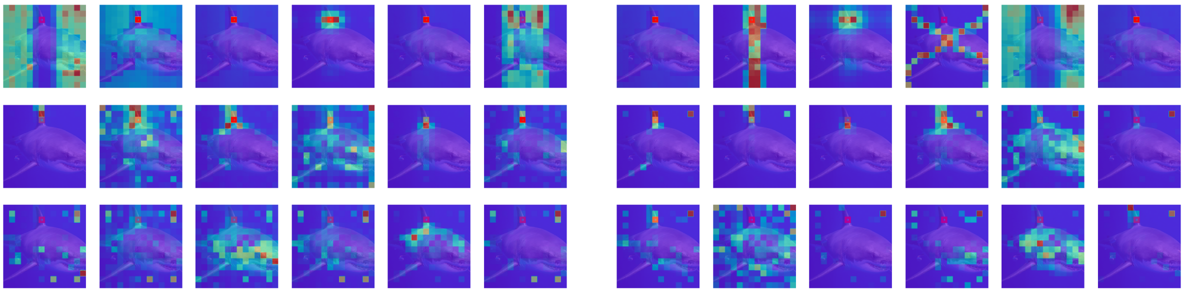


Figure A.1.: Attention scores of SoftMax (left) and MultiMax(right) at the input and hidden layers (1^{st}, 5^{th} and 10^{th}) w.r.t query 34. The query lies on the shark fin and is marked with red square. We see, from left to right, are attention scores of 6 heads for each method, where blue refers to low attention score and red indicates a high attention score. MultiMax attention is better localized while allowing for multiple modes.

from Eq. (A.3), $\forall\, x_i < \epsilon$, eliminate the numerator

$$m_{MultiMax} = 1 - (1 - e^{t_d(x_j - x_i)})/\Big(\sum_{x_l<b}^{L} e^{\sigma(x_l) - t_d x_i - (1-t_d)d}$$
$$+ \sum_{b \le x_m \le d}^{M} e^{x_m - t_d x_i - (1-t_d)d} + \sum_{x_n > d}^{N} e^{t_d(x_n - x_i)}\Big)$$

since $x_j - x_i < 1$ and $t_d < 1$, we have $e^{t_d(x_j - x_i)} > e^{x_j - x_i}$, also substitute $t_d x_i + (1 - t_d)d$ with t_x, from Lemma 1.1

$$> \frac{1 - e^{x_j - x_i}}{\sum_{x_l<b}^{L} e^{t_b x_l + (1-t_b)b - x_i} + \sum_{b \le x_m \le d}^{M} e^{x_m - x_i} + \sum_{x_n > d}^{N} e^{t_d(x_n - x_i)}}$$

$$\Rightarrow \mathcal{M}_{MultiMax}(\boldsymbol{x}) > \mathcal{M}_{MultiMax\text{-}l}(\boldsymbol{x}) \qquad \square$$

3. More visualizations

In the following, we provide additional visualizations for a more comprehensive qualitative comparison between SoftMax and our proposed MultiMax.

Single layer attention scores

As mentioned in Section 5.2, single layer attention scores Fig. A.1 are not informative for human beings, due to the complex interaction of information in deep transformer models.

Cumulative distribution of attention scores

We could calculate the cumulative distribution for each layer, i.e., the portion of attention scores smaller than a threshold as the thresholds increases. The result is shown in Fig. A.2. It can be seen that for most of the layers, MultiMax results in a sparser attention distribution, i.e., a large portion of attention scores are closer to zero comparing to SoftMax attention. Notably, the first two layers' attention distributions have a smaller degree of sparsity comparing to SoftMax. This shows that a smoother distribution is desired in

Layers	t_{b_1}	t_{d_1}	t_{b_2}	t_{d_2}	b_1	d_1	b_2	d_2
1	1.8347933	2.815388	0.9864913	0.68440557	1.185235	-1.208543	-2.1076407	1.9158255
2	1.9773115	1.9971638	0.985555	0.74650276	-0.8580209	0.02481092	-0.49835142	1.9772723
3	-1.1411996	1.4711196	1.9901285	0.8758977	0.18852632	2.8039892	2.9608543	1.0462786
4	0.6694808	1.206692	1.8682657	0.93786246	3.4023566	-1.5490056	2.500237	0.986331
5	0.8902384	1.5881691	1.8920481	0.72857785	2.5070796	-1.1942928	1.8854694	1.2248528
6	0.6015882	0.87738	2.818536	0.96271396	2.6490533	0.8454426	1.6205754	0.89434063
7	0.8023207	1.2427123	3.040797	0.84531546	2.6984618	1.2127148	1.2652112	1.2134424
8	0.64486825	0.79173684	2.5263662	0.968745	3.0230901	0.62191963	1.6307493	1.6259384
9	0.5796288	0.6852025	3.500835	0.99119073	2.675157	0.68776745	1.3239485	1.5808712
10	0.54873073	0.8240905	3.5563424	0.9692498	2.176066	0.39797062	0.9276044	1.5223614
11	0.38645744	0.6951747	4.0935583	0.9958999	1.6583583	0.29572898	0.77263904	2.9975116
12	0.16383016	0.25565386	3.2074118	0.99102634	1.6852132	-0.04795134	0.9796309	2.1836245

Table A.1.: MultiMax parameters of Deit-small trained on ImageNet.

Layers	t_{b_1}	t_{d_1}	t_{b_2}	t_{d_2}	b_1	d_1	b_2	d_2
1	0.6467285	0.7980957	0.98324585	0.9649048	0.7475586	-0.87939453	0.3395996	-0.14501953
2	0.69018555	0.8063965	0.98350525	0.9720764	0.25073242	0.15991211	0.2956543	-0.17687988
3	0.8557129	0.79797363	0.98939514	0.9855194	-0.12609863	0.06817627	0.14794922	-0.14428711
4	0.9662781	0.83569336	1.0231781	1.0240021	-0.07574463	0.8510742	-0.13220215	0.27368164
5	0.9260864	0.9187622	0.98670197	1.039093	-0.5239258	0.51416016	0.23999023	0.09521484
6	1.1514893	1.152832	0.98441315	1.0156403	0.1751709	0.05374146	-0.13269043	-0.08825684

Table A.2.: MultiMax parameters of the 6-layer Language Transformer trained on WikiText-103.

these two layers, as an optimized result of the training. This conforms to the observation in the previous studies that common low-level features in the shallow layers are shared across image patches [192]. A sparse attention has a high risk of information lost.

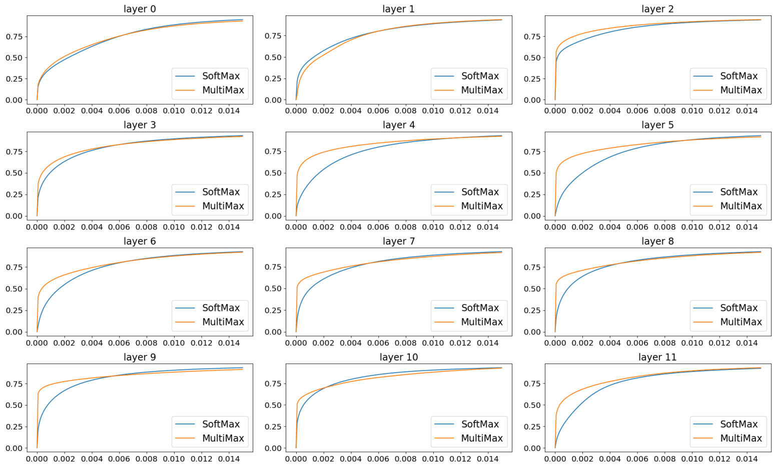


Figure A.2.: Cumulative distribution of the attention scores at each layer.

4. Connection between sparsification and over-smoothing

As shown by [1], information originating from different input tokens gets increasingly mixed in deeper layers, and the information flow can be estimated by taking the attention weights out and multiplying them sequentially. Such a matrix multiplication makes the identity of each token fades exponentially, which relates to the over-smoothing problem

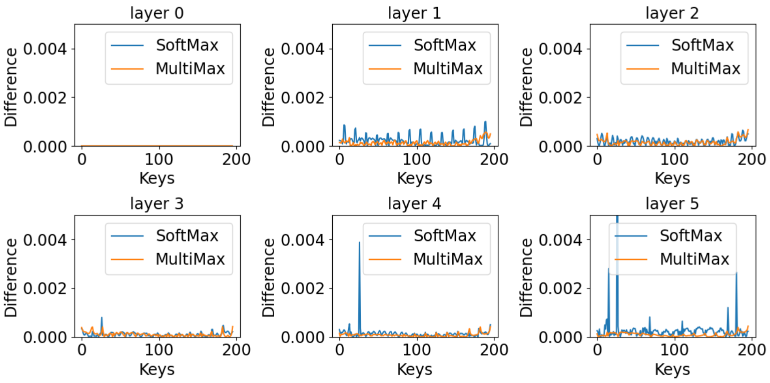


Figure A.3.: Comparing the discrepancy between rollout attention score and single layer attention score for SoftMax and MultiMax.

in GCNs [167]. Considering the information exchange across different attention heads, we take the the mean attention score over all heads out for multiplication, following the rollout technique [1]. In Fig. A.3, the discrepancy between the single layer and average accumulated SoftMax attention scores keeps increasing in the deeper layers. And the comparison shows a much less accumulated error for our MultiMax attention.

5. The learned parameters of MultiMax

In this section, we provide the learned parameters of MultiMax for reference. There are differences and similarities between the learned modulation functions of vision and language transformers, which could be observed after plotting the curves as shown in Fig. A.4.:

- Similarly, the need for sparsity increases as the layer goes deeper, but much less sparsity are needed in general for the language transformer compring to vision transformer, according to the learned parameters.
- As opposed to vision transformer, stronger multi-modality is needed at shallower layers of the language transformer.

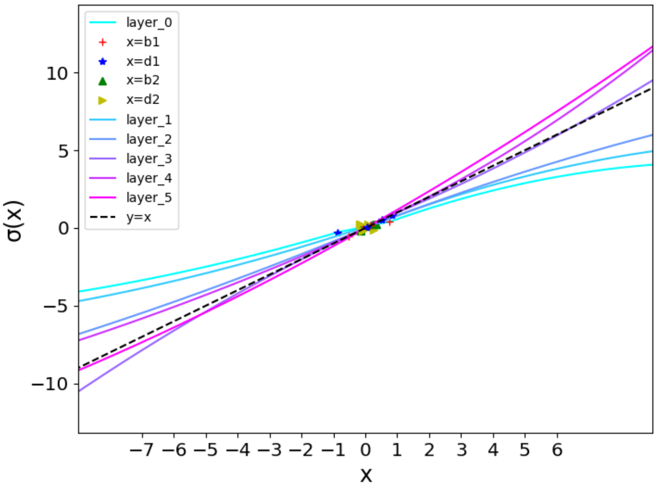


Figure A.4.: The learned modulator functions σ (Eq. (III.6)) at each layer of the 6-layer language transformer trained on WikiText-103, comparing to identity mapping of the SoftMax input $\boldsymbol{x}$ (dashed black line).

Appendix B.

Incorporating Geometric Prior into Attention for Image Modeling

In this section, we provide the supplementary materials for Chapter IV.

1. Experiment on CIFAR100

We have run additional experiments on CIFAR100 following the setup in DeiT [221] but w/o pretraining on ImageNet1K.

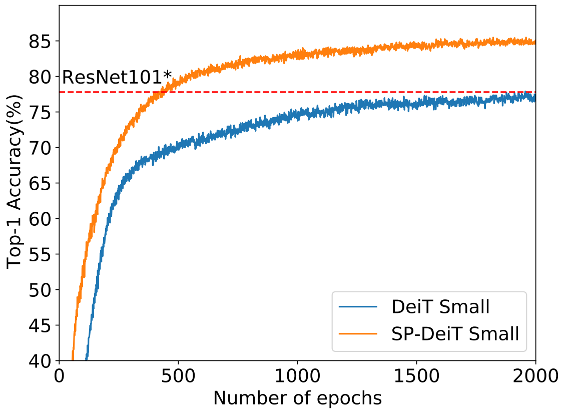
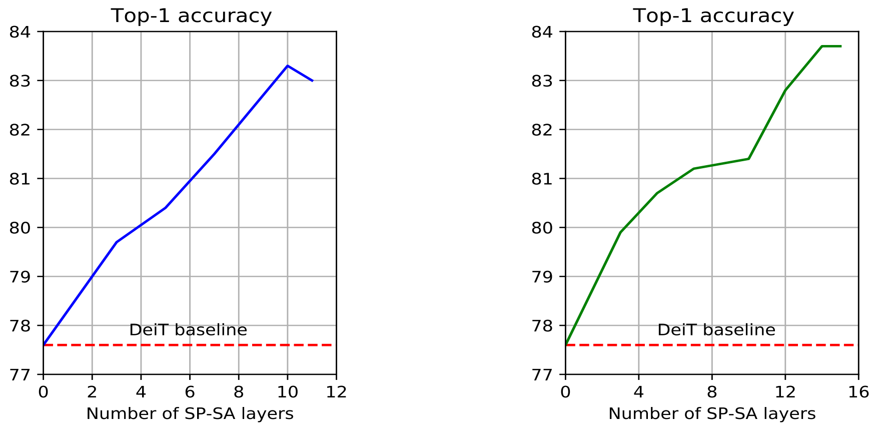


Figure B.1.: Training SP-ViT (DeiT [221] as baseline) on CIFAR100.

2. Numbers of Substituted SA Layers

We first investigate how the model performance is affected by the number of SP-SA layers. The layers are substituted from bottom to top and a classification token is inserted after the last SP-SA layer. It is shown in Fig. B.2a that substituting a number of SA layers with SP-SA results in improved accuracy comparing to DeiT baseline (0 layer). In general, the performance improves as more layers are substituted. For a model with 12 layers, the best performance is achieved when 10 layers are substituted. When substituting all but the last SA layer with SP-SA, the performance drops slightly. We hypothesize that when the classification token is only involved in the last layer, the class-specific features are not



(a) 12 layer SP-ViT

(b) 16 layer SP-ViT

Figure B.2.: Accuracy(%) of SP-ViT on ImageNet-100 with different numbers of SP-SA layers. Fig. B.2a and Fig. B.2b show consistent improvements of our SP-ViT over DeiT Baselines with a total number of 12 and 16 layers respectively.

Sub. layers	Cls token insertion layers	Top-1 (%)
0	0	77.6
0	10	81.7
10	10	**83.3**
0	Global Average Pooling	79.5
12	Global Average Pooling	**81.7**

Table B.1.: Eliminate the effect of inserting the class token at later layers on ImageNet-100.

adequately extracted. We further investigated a deeper model in Fig. B.2b, and found the similar trend. The best performance is achieved when the first to the penultimate layer are substituted. As discussed in the main text, we add the classification token directly after SP-SA layers because it has no valid 2D relative coordinate. To exclude the influence of inserting it at deeper layers instead of the first, we conduct a further comparison in Table B.1.

3. More Experiment Details

We show in Tab. B.2 the default hyperparameters for training our SP-ViT on ImageNet-1K based on DeiT and LV-ViT respectively. All hyperparameter settings follow the baselines' except that for DeiT-based SP-ViTs we adopt a smaller learning rate.

For our SP-ViT trained on ImageNet-1K, we further adopt the Conditional Positional Encoding (CPE) [44], which is found to be effective as shown in Table B.3.

4. Python Implementation

We also list our Pytorch implementation of SP-SA List. 1 SP-SA can be easily integrated into any existing vision transformer models by directly replacing a number of SA layers.

Base Config.	DeiT	LV-ViT
Supervision	Standard	Token labeling
SP-SA layers	10	10
Epoch	300	300
Optimizer	AdamW	AdamW
Batch size	1024	1024
LR	$2.5e-4 \cdot \frac{\text{batch size}}{512}$	$1e-3 \cdot \frac{\text{batch size}}{640}$
LR decay	cosine	cosine
Weight decay	0.05	0.05
Warmup epochs	5	5
Label smoothing ϵ	0.1	0.1
Stoch. Depth	0.1	0.1
Repeated Aug	✓	-
RandAug	9/0.5	9/0.5
Mixup prob.	0.8	-
Erasing prob.	0.25	0.25

Table B.2.: Default hyperparameters for our SP-ViTs on ImageNet-1K.

Model	CPE [44]	Top-1 (%)
SP-ViT-S	-	83.7
	✓	83.9
SP-ViT-M	-	84.7
	✓	84.9
SP-ViT-L	-	85.3
	✓	85.5

Table B.3.: Effect of Conditional Positional Encoding [44] on ImageNet-1K.

Calculating the relative coordinates to query patches is trival, so this part of code is not included for simplicity. Note that the insertion of classification token should be moved after SP-SA layers, as mentioned in the main text.

5. More Visualization

We provide more examples of learned Spatial Priors (SP) by our SP-ViT based on DeiT-Small and trained on ImageNet-1K in Fig. B.3 and Fig. B.4.

Listing 1 SP-SA `SP-SA.py`

```python
import torch
from torch import nn

class SP_SA(nn.Module):
    def __init__(self, dim, num_heads=8, qk_scale=None, attn_drop=0., proj_drop=0., rel_indices=None, **kwargs):
        super().__init__()
        self.num_heads = num_heads
        self.dim = dim
        head_dim = dim // num_heads
        self.scale = qk_scale or head_dim ** -0.5
        self.v = nn.Linear(dim, dim, bias=False)
        self.qk = nn.Linear(dim, dim * 2, bias=False)
        self.w1 = nn.Linear(2, dim, bias=True)
        self.w2 = nn.Parameter(torch.zeros(dim, 1))
        self.b2 = nn.Parameter(torch.ones(num_heads))

        self.attn_drop = nn.Dropout(attn_drop)
        self.proj = nn.Linear(dim, dim)
        self.proj_drop = nn.Dropout(proj_drop)
        self.act = nn.ReLU()
        self.rel_indices = rel_indices

    def forward(self, x):
        B, N, C = x.shape
        attn = self.get_attention(x)

        v = self.v(x).reshape(B, N, self.num_heads, C // self.num_heads).permute(0, 2, 1, 3)
        x = (attn @ v).transpose(1, 2).reshape(B, N, C)
        x = self.proj(x)
        x = self.proj_drop(x)
        return x

    def get_attention(self, x):
        B, N, C = x.shape

        # Calculating Patch Score
        qk = self.qk(x).reshape(B, N, 2, self.num_heads, C // self.num_heads).permute(2, 0, 3, 1, 4)
        q, k = qk[0], qk[1]
        patch_score = (q @ k.transpose(-2, -1)) * self.scale

        # Calculating Spatial Prior
        sp_hidden = self.w1(self.rel_indices).view(1, N, N, self.num_heads, self.dim // self.num_heads)
        sp = torch.einsum(’nm,hijnm->hijn’, (self.w2.view(self.num_heads, -1), self.act(sp_hidden))) + self.b2
        sp = sp.repeat(B, 1, 1, 1)

        enhanced_attention = (patch_score * sp.permute(0, 3, 1, 2)).softmax(dim=-1)
        attn = self.attn_drop(enhanced_attention)
        return attn
```

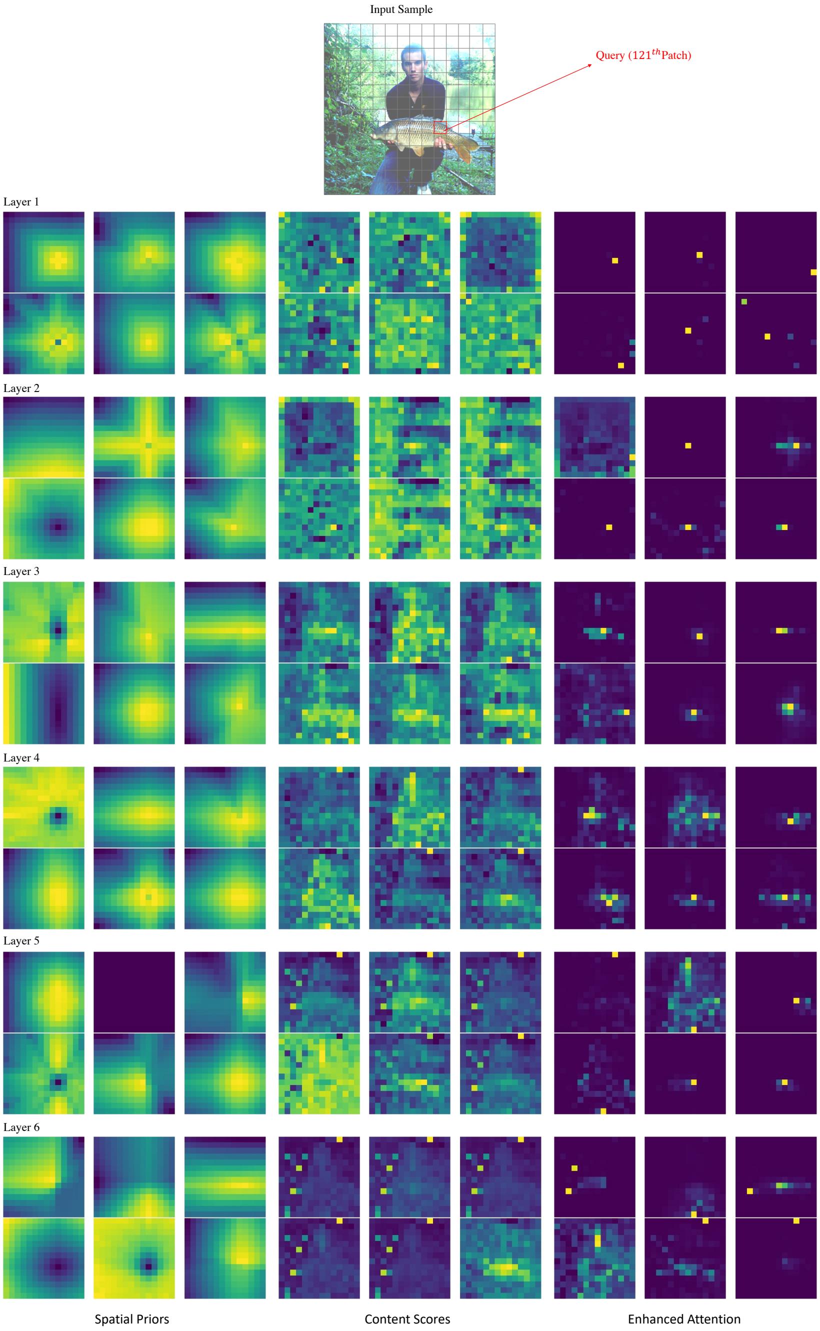


Figure B.3.: More Visualization of the learned 2D SPs, content scores and the enhanced attention of layer 1-6 for the 121th query patch.

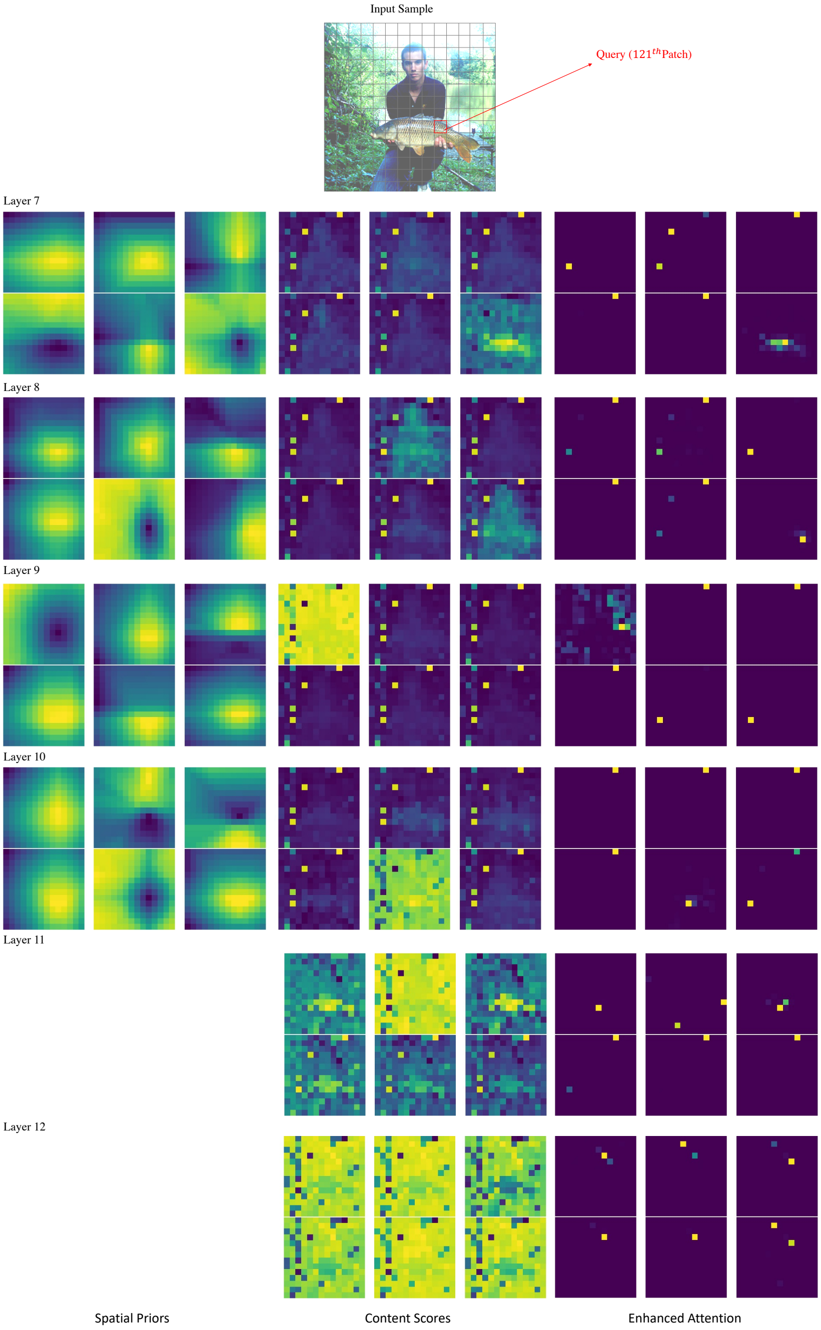


Figure B.4.: More Visualization of the learned 2D SPs, content scores and the enhanced attention of layer 7-12 for the 121th query patch. Note that layer 11 and 12 are vanilla SA layers, thus no spatial priors are existed.

Appendix C.

Incorporating Structural Prior into Attention for Skeleton-Based Action Recognition

In this section, we provide the supplementary materials for Chapter V.

1. More experiment details

We show in Table C.1 the default hyperparameters for training our Hyperformer on NTU RGB+D, NTU RGB+D 120 and Northwestern-UCLA datasets. We train the same 10-layer model with a total number of 216 channel dimensions for all the experiments in our paper.

Table C.1.: Default hyperparameters for our Hyperformer on NTU RGB+D, NTU RGB+D 120 and Northwestern-UCLA.

Config.	NTU RGB+D and NTU RGB+D 120	Northwestern-UCLA
random choose	False	True
random rotation	True	False
window size	64	52
weight decay	4e-4	0
base lr	2.5e-2	2.5e-2
lr decay rate	0.1	0.1
lr decay epoch	110, 120	110 120
warm up epoch	5	5
batch size	64	16
num. epochs	140	150
optimizer	Nesterov Accelerated Gradient	Nesterov Accelerated Gradient

2. More experiment results

In the following, we provide additional experiment results in detail to further support the effectiveness of our proposed Hyperformer.

Accuracy using single modalities

The performance of our Hyperformer trained on joint modality only is also remarkable. We provide the experiment results for each modality on different benchmarks in detail, see Tab. 2.

Modality	Model Size	NTU-RGB+D 120		NTU-RGB+D		NW-UCLA(%)
		X-Sub(%)	X-Set(%)	X-Sub(%)	X-View(%)	
Joint	2.6M	86.6	88.0	90.7	95.1	94.4
Bone		88.0	89.0	91.2	95.2	94.6
Motion		81.8	83.9	88.5	93.3	93.3
Bone Motion		82.2	83.5	88.5	92.6	92.7
Ensembled		89.9	91.3	92.9	96.5	96.7

Table C.2.: Classification accuracy of our Hyperformer using different modalities on the NTU RGB+D, NTU RGB+D 120 and Northwestern-UCLA dataset.

Effect of randomness

Table C.3.: Effect of randomness.

	Methods	NTU RGB+D 60		NTU RGB+D 120	
		X-Sub	X-View	X-Sub	X-Set
Ours	Hyperformer	90.7	95.1	86.6	88.0
		90.7	95.0	86.3	88.1
		90.6	95.2	86.6	88.2
	mean (std)	**90.67** (0.08)	95.10 (0.08)	**86.5** (0.14)	**88.10** (0.08)

To check the effect of randomness, we run our model on NTU-RGB+D 60&120 using joint modality three times and report the results in Appendix 2. It can be seen that the standard deviations are relatively small (below 0.2%) and our model delivers stable performance.

More comparison using joint modality only

We compare to more methods using joint modality only in Table 4 and Hyperformer performs the best on the most challenging NTU120 when fairly compared. HD-GCN ensembles 6 modalities and removes the weaker motion modalities to achieve the reported results. LST and InfoGCN rely on additional training losses and require intricate tuning of the associated hyperparameters. However, Hyperformer still outperforms them by a large margin, as shown in Table C.4.

The effect of additional losses

No performance improvement is observed after training our model with additional MMD losses [Chi et al. 2022]. Therefore, we further validate the effectiveness of MMD losses [Chi et al. 2022], but they are found to be useless, as shown in Table C.5.

Table C.4.: Performance of recent methods using joint modality only. We denote the methods that are not directly comparable with * (rely on additional supervision signal)

States	Methods	NTU RGB+D 60		NTU RGB+D 120	
		X-Sub	X-View	X-Sub	X-Set
SOTA	MST-GCN (Chen et al. 2021)	89.0	95.1	82.8	84.5
	InfoGCN* (Chi et al. 2022)	89.4*	95.2*	84.2*	86.3*
	LST* (Xiang et al. 2022)	90.2*	**95.6***	85.5*	87.0*
	HD-GCN (Lee et al. 2021)	-	-	85.7	87.3
Ours	Hyperformer	**90.7**	95.1	**86.6**	**88.0**

Table C.5.: The reported and reproduced results of InfoGCN on NTU RGB+D 60 X-Sub. * denotes the results with MMD losses.

Modality	Joint	Bone	Joint Vel.	Bone Vel.	4S
InfoGCN [Chi et al., 2022]	89.8*	90.6*	88.9*	88.6*	92.7*
InfoGCN ([Huang et al.] reproduced)	89.4*	90.6*	-	-	92.3*
InfoGCN (our reproduced)	89.5*	90.5*	88.6*	88.3*	92.3*
InfoGCN (our reproduced)	89.6	90.3	88.7	88.3	92.4

3. Python implementation

We list our Pytorch implementation of HyperSA layer in Listing 2. For simplicity, we omit the code for initialization.

Listing 2 `HyperSA.py`

```
import torch
from torch import nn
class HyperSA(nn.Module):
    def __init__(self, dim_in, dim, num_heads=9, qkv_bias=False, H=None, qk_scale=None, hops=None, num_point=25):
        '''
        :param H: Incidence Matrix
        :param hops: Shortest Path Distance (SPD)
        '''
        super().__init__()
        self.num_heads = num_heads
        self.dim = dim
        head_dim = dim // num_heads
        self.scale = qk_scale or head_dim ** -0.5
        self.num_point = num_point
        self.rpe_table = nn.Parameter(torch.zeros((hops.max()+1, dim)))
        self.u = nn.Parameter(torch.zeros(num_heads, head_dim))
        self.relational_bias = nn.Parameter(torch.stack([torch.eye(num_point]) for _ in range(num_heads)], dim=0), requires_grad=True)
        self.qkv = nn.Conv2d(dim_in, dim * 3, 1, bias=qkv_bias)
        self.proj = nn.Conv2d(dim, dim, 1)
        self.e_proj = nn.Conv2d(dim_in, dim, 1, bias=False)
        self.H = H

    def forward(self, x, joint_label, groups, pe):
        N, C, T, V = x.shape
        qkv = self.qkv(x).reshape(N, 3, self.num_heads, self.dim // self.num_heads, T, V).permute(1, 0, 4, 2, 5, 3)
        q, k, v = qkv[0], qkv[1], qkv[2]
        # Deriving hyperedge representation
        e = x@self.H/torch.sum(self.H, dim=0, keepdim=True)
        e = self.e_proj(e)
        e_aug = e@self.H.transpose(0, 1)
        e_aug = e_aug.reshape(N, self.num_heads, self.dim // self.num_heads, T, V).permute(0, 3, 1, 4, 2)
        pos_emb = self.rpe_table[self.hops]
        r = pos_emb.view(V, V, self.num_heads, self.dim // self.num_heads)
        a = q @ k.transpose(-2, -1)
        b = torch.einsum("bthnc, nmhc->bthnm", q, r)
        c = torch.einsum("bthnc, bthmc->bthnm", q, e_aug)
        d = torch.einsum("hc, bthmc->bthm", self.u, e_aug).unsqueeze(-2)
        attn = (a + b + c + d) * self.scale
        attn = attn.softmax(dim=-1)
        x = (attn + self.relational_bias) @ v
        x = x.transpose(3, 4).reshape(N, T, -1, V).transpose(1, 2)
        x = self.proj(x)
        return x
```

Appendix D.

Extending the Structural Prior with Topological Analysis beyond Connectivity

In this section, we provide the supplementary materials for Chapter VI.

1. Supplementary Material Structure

This supplementary material provides additional technical explanations and experimental validations to support and expand upon the main text of our work. The contents are organized as follows:

(i) Detailed elaboration of the dynamic topological encoding scheme, Appendix 2.

  (1) Definition and illustration of essential terms and concepts, Appendix 2.1.

  (2) Theoretical foundation and methodology of persistent homology analysis for graph-structured data, Appendix 2.2.

  (3) Comprehensive explanation of the adopted vectorization representation strategy, Appendix 2.3.

(ii) In-depth discussion of the hyperparameter settings and optimization of BlockGCN, Appendix 3.

(iii) Extended experimental validations and analysis, Appendix 4.

  (1) Evaluation and comparison of single modality performance, Appendix 4.1.

  (2) Investigation of the impact of different graph distance metrics on model performance, Appendix 4.2.

  (3) Visual exploration and interpretation of the learned feature representations, Appendix 4.4.

2. Technical Preliminaries

2.1. Fundamentals of Algebraic Topology

Topological data analysis (TDA) [197] leverages algebraic topology tools, such as persistent homology [68], to extract topological features, including connected components and cycles, from graph data that persist across multiple scales [5]. These topological descriptors have been shown to be effective representations for graph classification tasks [271, 186]. Furthermore, integrating these topological features with deep learning architectures has achieved significant success in enhancing the representational power of the models [271, 261, 159, 109, 53, 250]. In this section, we first introduce the core notations and concepts, followed by a general description of persistent homology analysis for graph data, and finally present a toy demonstration for intuitive understanding. For more detailed descriptions and formal illustrations of these techniques, we refer the reader to the corresponding literature in computational topology and topological data analysis [30, 67, 103].

Simplicial Complex: A simplicial complex is composed of simplices of different dimensions, such as vertices (0-simplices), edges (1-simplices), triangles (2-simplices), and tetrahedra (3-simplices). Given a k-simplex denoted as $\sigma = [v_0, ..., v_k]$, deleting one of its vertices v_i results in a $(k-1)$-simplex $[v_0, \ldots, \hat{v}_i, \ldots, v_k]$ ($\hat{v}_i$ denotes the deleted vertex), which is called the i-th *face* of σ. A simplicial complex $\mathcal{K}$ is defined as a set of simplices of varying dimensions that satisfies the following conditions:

(i) Any face τ of a simplex $\sigma \in \mathcal{K}$ is also in $\mathcal{K}$ (i.e., $\tau \in \mathcal{K}$).

(ii) If $\sigma_1, \sigma_2 \in \mathcal{K}$ and $\sigma_1 \cap \sigma_2 \neq \emptyset$, then $\sigma_1 \cap \sigma_2$ is a face of both σ_1 and σ_2.

A graph $\mathcal{G}$ is a simplicial complex $\mathcal{K}$ consisting only of vertices (0-simplices) and edges (1-simplices).

Boundary Map: Given a simplicial complex $\mathcal{K}$, consider the vector space $C_\kappa(\mathcal{K})$ generated with $\mathbb{Z}2$ (the field with two elements). The boundary map is denoted as $\partial\kappa : C_\kappa(\mathcal{K}) \to C_{\kappa-1}(\mathcal{K})$. For a k-simplex $\sigma = [v_0, \ldots, v_k) \in \mathcal{K}]$, the boundary map is defined as:

$$\partial_\kappa(\sigma) := \sum_{i=0}^{k} (v_0, \ldots, v_{i-1}, v_{i+1}, \ldots, v_k) \tag{D.1}$$

In other words, each vertex v_i of the simplex is omitted once. The boundary operator ∂ is a homomorphism between the simplicial chain groups, providing a precise way to define connectivity [109].

Homology: Homology theory employs commutative algebra tools to study topological features, such as connected components ($\kappa = 0$) and cycles ($\kappa = 1$) in a graph [67], using the boundary operator. The κ-th homology group $\mathbb{H}\kappa(\mathcal{K})$ of a simplicial complex $\mathcal{K}$ is defined as the quotient group:

$$\mathbb{H}\kappa(\mathcal{K}) := \mathbf{ker}\partial_\kappa / \mathbf{im}\partial_{\kappa+1} \tag{D.2}$$

The elements in $\mathbf{ker}(\partial_\kappa)$ and $\mathbf{im}(\partial_{\kappa+1})$ are called *κ-cycles* and *κ-boundaries*, respectively. The resulting homology groups $\mathbb{H}_\kappa(\mathcal{K})$ are topological invariants that remain unchanged under homeomorphisms and encode intrinsic information [103].

Betti Numbers: Betti numbers, defined as the ranks of the homology groups, serve as simpler invariants for classifying topological spaces. For $\mathbb{H}_\kappa(\mathcal{K})$, the 0-th Betti number $\beta_0 = \mathbf{rank}\mathbb{H}_0(\mathcal{K})$ represents the number of connected components, while the 1-st Betti number $\beta_1 = \mathbf{rank}\mathbb{H}_1(\mathcal{K})$ represents the number of cycles when $\kappa = 0$ and $\kappa = 1$, respectively. However, these counting-based topological summaries are too coarse to capture the complexity of graph structures. To address this limitation, a persistent version of homology-based topological invariant analysis is proposed, as described in the following section.

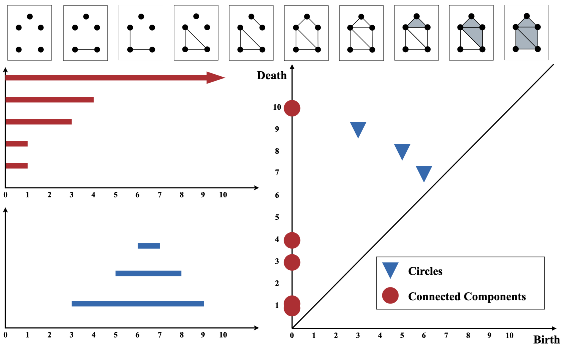


Figure D.1.: A graph filtration with $\epsilon = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9$ (from left to right): (a) the persistence barcodes of connected components (up) and cycles (down); (b) corresponding persistent diagram of connected components (red disk) and cycles (blue triangle).[Best viewed in zoom and color]

2.2. Persistent Homology Analysis for Graphs

In this subsection, we provide an overview of the persistent homology analysis for graphs, followed by an intuitive demonstration using a 5-node graph example. We then introduce the key notations and concepts for further reference.

Intuitive Demonstration: Consider an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a vertex set $\mathcal{V}$ and an edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. Given a threshold value ϵ, we can obtain a series of graphs by setting the edge weights $w_{ij}^{(\epsilon)}$ to 1 if $w_{ij}^{(\epsilon)} > \epsilon$, and 0 otherwise. Treating the graph $\mathcal{G}$ as a simplicial complex $\mathcal{K}$, we generate a sequence of simplicial complexes, termed as a *filtration*, $\{\mathcal{K}^i\}_{i=0}^m$, where $\emptyset = \mathcal{K}^0 \subseteq \mathcal{K}^1 \subseteq \ldots \subseteq \mathcal{K}^m = \mathcal{K}$, by increasing the threshold value ϵ. As the filtration parameter increases, more edges are removed from the graph. In extreme cases, when $\epsilon \to -\infty$, the graph becomes complete, and when $\epsilon \to \infty$, the graph reduces to a vertex set $\mathcal{V}$. For each sub-complex, we record the topological invariants, such as connected components and cycles, to describe the graph structure. During this filtration process, each topological object (i.e., homology) may appear at a specific ϵ_i and disappear at another value ϵ_j. The interval $\{\epsilon_i, \epsilon_j\}$ is called its *persistence*. *Persistent homology* analysis captures the global structure of graphs by recording these paired filtration values in the nested sequence. *Persistence barcodes* and *persistence diagrams* are used to represent the paired set $\{(b_i^{(0)}, d_i^{(0)})\}_{i=1}^n$, where $\mathcal{D}_i^{(0)} = (b_i^{(0)}, d_i^{(0)})$ and $b_i^{(0)}, d_i^{(0)} \in \{\epsilon_0, \epsilon_1, \ldots, \epsilon_k\}$ for connected components, and superscripts equal to 1 for cycles.

Figure D.1 presents an intuitive demonstration of a 5-node graph filtration with threshold values $\epsilon = 0, 1, \ldots, 9$. As ϵ increases from 0 to 9, edges gradually appear, forming different combinations of connected components and cycles. For example, when ϵ increases from 0 to 1, the number of connected components decreases from 5 to 4 as one edge emerges. When ϵ increases from 2 to 3, a cycle appears and persists until $\epsilon = 9$. Through this counting and recording process, the geometrical structure of a weighted graph is explored globally.

Persistent Homology: Given a filtration of $\mathcal{K}$ denoted as $\{\mathcal{K}_i\}_{i=0}^m$, we have a corresponding sequence of chain complexes $C_\kappa(\mathcal{K}^i)$. The concept of homology groups is extended from $\mathbb{H}_\kappa^i(\mathcal{K}) := \mathbf{ker}\partial_\kappa^i / \mathbf{im}\partial_{\kappa+1}^i$ (dependent on a single simplicial complex $\mathcal{K}^i$) to its persistent version (from $\mathcal{K}^i$ to $\mathcal{K}^j$) as:

$$\mathbb{H}_\kappa^{i,j}(\mathcal{K}) := \mathbf{ker}\partial_\kappa^i / (\mathbf{im}\partial_{\kappa+1}^j \cap \mathbf{ker}\partial_\kappa^i) \tag{D.3}$$

Table D.1.: Default Hyperparameters for BlockGCN on NTU RGB+D, NTU RGB+D 120, and Northwestern-UCLA.

Config.	NTU RGB+D 60 and 120	NW-UCLA
random choose	False	True
random rotation	True	False
window size	64	52
weight decay	4e-4	3e-4
base lr	0.05	0.05
lr decay rate	0.1	0.1
lr decay epoch	110, 120	90 100
warm up epoch	5	5
batch size	64	16
num. epochs	140	120
optimizer	Nesterov Accelerated Gradient	Nesterov Accelerated Gradient

The ranks of all the homology groups $\beta_\kappa^{i,j} = \mathbb{H}_\kappa^{i,j}(\mathcal{K})$ (namely the κ-th persistent Betti numbers) capture the number of homological features of dimensionality κ (e.g., connected components for $\kappa = 0$, cycles for $\kappa = 1$, etc.) that persist from i to (at least) j [106].

Persistence Barcodes of Filtration: For simplification, we use $\mathbb{R}^2$ of $\{\mathcal{D}_1^{(0)}, \mathcal{D}_2^{(0)}, \ldots, \mathcal{D}_p^{(0)}\}$, where $\mathcal{D}_i^{(0)} = \{(b_i^{(0)}, d_i^{(0)})\}$, to denote the barcodes extracted from $\mathcal{K}$. Formally, the filtration sequence of $\mathcal{K}$ can be defined using a vertex filter function $f : \mathbb{V} \to \mathbb{R}$ with the filtration values $\epsilon_1 < \epsilon_2 \cdots \epsilon_m$, where $\epsilon_i \in \{f(v) : \{v\} \in \mathcal{K}\}$. With function f, the filtration of $\mathcal{K}$ is:

$$\mathcal{K}^{f,0} = \emptyset, \quad \mathcal{K}^{f,i} = \{\sigma \in \mathcal{K} : \max_{v \in \sigma} f(v) \leq \epsilon_i\} \tag{D.4}$$

for $1 \leq i \leq m$. Then, for the filtration of $\mathcal{K}$ and homology dimension κ ($\kappa = 0, 1$ in this work), we obtain the persistence barcode representation $\{\mathcal{D}_i^{(0)}\}_{i=1}^m = \{(b_i^{(0)}, d_i^{(0)})\}_{i=1}^m$, which we denote as $\mathcal{B}$.

2.3. Vectorization Representation

The inconsistency of using persistence barcodes $\{(b_i^{(0)}, d_i^{(0)})\}_{i=1}^m$ in machine learning tasks has led to the development of various vectorization approaches, including statistical analysis [158, 16], kernel methods [137, 185, 130, 37, 25], distance metrics [157, 47], and $\mathbb{R}^d$ elements [21, 3, 28, 9, 114].

Recently, learning-based techniques have been proposed to facilitate the integration of such graph descriptions into modern deep learning architectures by introducing learnable weights for each barcode [106, 109]. Typical embedding functions include the *rational hat* function [106], point transformation-based techniques [26], and the *DeepSets* approach [259] adopted in [109].

For computational efficiency and ease of implementation, we employ the *rational hat* function, as described in [106], for vectorization extraction due to its differentiability and expressive power in representing graphs. Mathematically, the *barcode coordinate function* maps a barcode in $\mathcal{B}$ to a real value by aggregating the points in the persistence diagram via a weighted sum:

$$\Psi : \mathbb{B} \to \mathbb{R} \quad \mathcal{B} \to \sum_{(b,d) \in \mathcal{B}} s(b, d) \tag{D.5}$$

Table D.2.: Classification Accuracy (%) of BlockGCN using Different Modalities on NTU RGB+D, NTU RGB+D 120, and Northwestern-UCLA Dataset.

Modality	NTU-RGB+D		NTU-RGB+D 120		NW-UCLA
	X-Sub	X-View	X-Sub	X-Set	
Joint	90.9	95.4	86.9	88.2	95.5
Bone	91.3	95.3	88.1	89.3	93.3
Motion	88.7	93.3	82.7	84.6	92.9
Bone Motion	88.3	92.6	83.0	84.8	88.8
Ensembled	93.1	97.0	90.3	91.5	96.9

where $s : \mathbb{R}^2 \to \mathbb{R}$ is a differentiable function that vanishes on the diagonal of $\mathbb{R}^2$. The rational hat structure element from [107] is defined as:

$$p \in \mathcal{B} \quad p \to \frac{1}{1 + \|p - c\|_1} - \frac{1}{1 + ||r| - \|p - c\|_1|} \tag{D.6}$$

where $c \in \mathbb{R}^2$ and $r \in \mathbb{R}$ are learnable parameters. This function evaluates the "centrality" of each point $p \in \mathbb{B}$ with respect to a learned center c and a learned shift/radius r.

In our implementation, we adopt the modified version of the *rational hat* function provided in the *Pytorch-topological*[1] library, which is based on the original implementation by [106]. This vectorization approach allows us to transform the persistence barcodes into fixed-length feature vectors that can be readily integrated with deep learning models, such as the BlockGCN architecture used in our work. By learning the parameters of the *rational hat* function, we can adaptively capture the most informative topological features for the given graph classification task, enhancing the expressive power and discriminative capability of our model.

3. Hyperparameter Settings

In this section, we provide the default hyperparameter settings used for training our BlockGCN model on the NTU RGB+D, NTU RGB+D 120, and Northwestern-UCLA datasets. Throughout our experiments, we consistently train a 10-layer BlockGCN with a maximum channel dimension of 256. Table D.1 presents the default hyperparameters for our BlockGCN model on these datasets. These hyperparameter settings have been carefully tuned to achieve optimal performance on each dataset while maintaining a balance between model complexity and computational efficiency. By using consistent hyperparameter settings across all experiments, we ensure a fair comparison and evaluation of our BlockGCN model's performance on different datasets and modalities.

4. Extended Experimental Results

In this section, we present additional experimental results to provide a more comprehensive evaluation of our BlockGCN model's performance on various datasets and modalities.

[1] https://pypi.org/project/torch-topological/

4.1. Single Modality Performance

To gain further insights into the contribution of each modality to the overall performance of our BlockGCN model, we conduct experiments training the model on each single modality separately. Table D.2 provides detailed results of our BlockGCN's performance on each modality for the different benchmark datasets. These results demonstrate the effectiveness of our BlockGCN model in learning discriminative features from individual modalities, such as skeleton, RGB, depth, and infrared data. By examining the performance on each modality, we can identify the strengths and weaknesses of our model in capturing modality-specific information and guide future research efforts towards improving the fusion of multi-modal features. The single modality performance also serves as a baseline for evaluating the benefit of multi-modal fusion in our BlockGCN model. By comparing the results of single modality training with those of multi-modal fusion, we can quantify the synergistic effect of combining complementary information from different modalities to enhance the overall recognition accuracy.

4.2. Selection of Graph Distance for Static Topological Encoding

In the main text, we discuss the use of relative distances between joint pairs on the graph to symbolize graph topology. Theoretically, any proper graph distance can serve this purpose. In our work, we investigate two common graph distances for our Static Topological Encoding: the shortest path distance and the distance in the level structure [59]. Table D.3 compares these two distances. Interestingly, both distances lead to an equivalent improvement, suggesting that they fundamentally convey the same information, i.e., bone connectivity. To streamline our approach, we default to employing the shortest path distance.

The choice of graph distance for Static Topological Encoding is an important consideration, as it directly influences the model's ability to capture the intrinsic topology of the skeleton graph. By comparing the performance of different graph distances, we can identify the most informative and computationally efficient representation for encoding the graph topology. The equivalent improvement observed when using either the shortest path distance or the distance in the level structure indicates that both distances effectively capture the essential connectivity information of the skeleton graph. This finding simplifies the implementation of our Static Topological Encoding, as we can focus on using the shortest path distance without compromising the model's performance.

Table D.3.: Comparing different graph distances for our Static Topological Encoding.

Graph Distance		Acc(%)
shortest path distance	level difference	
-	-	86.7
✓	-	86.9
-	✓	86.9

4.3. Choice of Simplicial Complex

In addition to the graph distance, we also explore the choice of simplicial complex for persistent homology analysis used in our dynamic topological encoding. Table D.4 shows the comparison between two commonly used simplicial complexes: the Vietoris-Rips Complex and the Cubical Complex. The results indicate that using the Cubical Complex leads to a slight decrease of 0.2% in accuracy and significantly longer run time compared to the Vietoris-Rips Complex. Based on these findings, we adopt the Vietoris-Rips Complex for our dynamic topological encoding.

The choice of simplicial complex is crucial for efficient and effective persistent homology analysis. The Vietoris-Rips Complex, which is based on pairwise distances between points, provides a good balance between topological expressiveness and computational efficiency. On the other hand, the Cubical Complex, which is based on a cubical grid, may introduce additional computational overhead without providing significant benefits in terms of accuracy. By selecting the Vietoris-Rips Complex for our dynamic topological encoding, we ensure that our model can efficiently capture the evolving topological features of the skeleton graph over time, while maintaining high recognition accuracy.

Table D.4.: Comparing different simplicial complices.

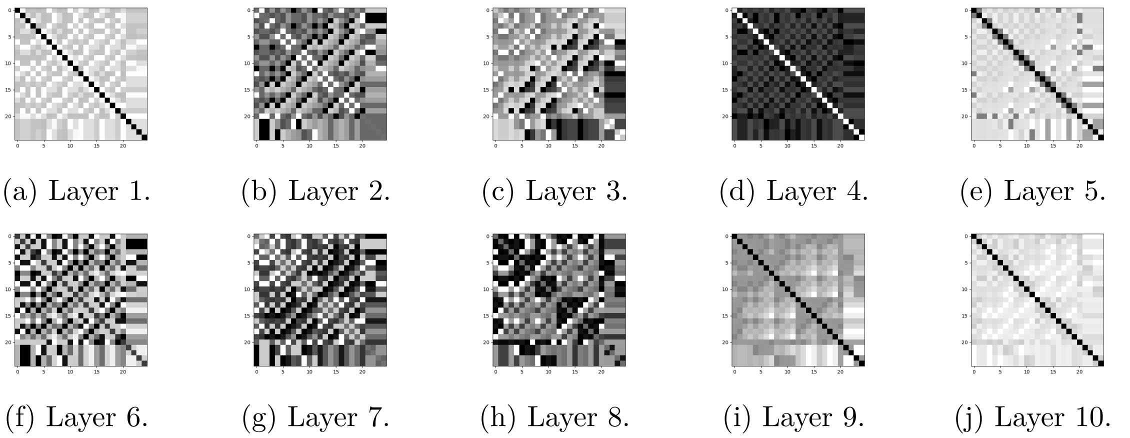
Vietoris–Rips Complex	Cubical Complex	Acc(%)
✓	-	86.9
-	✓	86.7

4.4. Visualization of Learned Representations

To gain further insights into the learned representations of our BlockGCN model, we provide additional visualizations of the Static Topological Encodings and the learned adjacency matrices.

Figure D.2 presents more examples of the learned Static Topological Encodings, showcasing the model's ability to capture the intrinsic topology of the skeleton graph. These visualizations illustrate how our model learns to encode the relative distances between joint pairs, effectively representing the connectivity information of the skeleton.

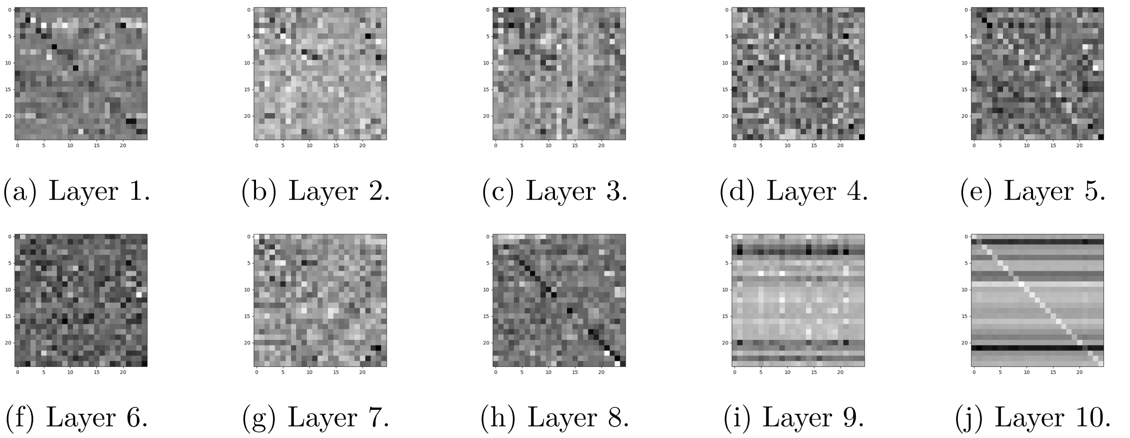
Figure D.3 visualizes the learned adjacency matrices of our BlockGCN model. These matrices represent the learned graph structure and the strength of connections between different joints. By examining these visualizations, we can gain insights into how our model adapts the graph structure to better capture the dependencies and relationships between joints for action recognition. The visualizations of the learned Static Topological Encodings and adjacency matrices provide a qualitative assessment of our BlockGCN model's learning process.



(a) Layer 1. (b) Layer 2. (c) Layer 3. (d) Layer 4. (e) Layer 5.

(f) Layer 6. (g) Layer 7. (h) Layer 8. (i) Layer 9. (j) Layer 10.

Figure D.2.: The learned Static Topological Encodings of our BlockGCN at each layer. It can be seen that the learned weights are diverse and adapted to different levels of semantics.



(a) Layer 1. (b) Layer 2. (c) Layer 3. (d) Layer 4. (e) Layer 5.

(f) Layer 6. (g) Layer 7. (h) Layer 8. (i) Layer 9. (j) Layer 10.

Figure D.3.: The learned adjacency matrices of the GCN baseline model at each layer (Darker colors stand for larger weights). It can be seen that the learned weights vary dramatically among different layers and deviate far from the bone connections, which are used for initialization.

Appendix E.

Overcoming the Error-Enhancement Defect in Label Smoothing for Image Classifiers

In this section, we provide the supplementary materials for Chapter VII.

1. Proof of Lemma 3.2

Proof. We aim to demonstrate the validity of Lemma 3.2, which states:

$$H(\mathbf{s}, \mathbf{q}) = H(\mathbf{y}, \mathbf{q}) + L_{LS} \tag{E.1}$$

where $L_{LS} = \alpha \left(H\left(\frac{\mathbf{1}}{K}, \mathbf{q}\right) - H(\mathbf{y}, \mathbf{q}) \right)$
Let us proceed with the proof:
We begin by expressing the cross-entropy $H(\mathbf{s}, \mathbf{q})$:

$$H(\mathbf{s}, \mathbf{q}) = -\sum_{k=1}^{K} s_k \log q_k \tag{E.2}$$

In the context of label smoothing, s_k is defined as:

$$s_k = (1-\alpha) y_k + \frac{\alpha}{K} \tag{E.3}$$

where α is the smoothing parameter, y_k is the original label, and K is the number of classes.

Substituting this expression for s_k into the cross-entropy formula:

$$H(\mathbf{s}, \mathbf{q}) = -\sum_{k=1}^{K} \left((1-\alpha) y_k + \frac{\alpha}{K} \right) \log q_k \tag{E.4}$$

Expanding the sum:

$$H(\mathbf{s}, \mathbf{q}) = -(1-\alpha) \sum_{k=1}^{K} y_k \log q_k - \frac{\alpha}{K} \sum_{k=1}^{K} \log q_k \tag{E.5}$$

We recognize that the first term is equivalent to $(1-\alpha)H(\mathbf{y}, \mathbf{q})$, and the second term to $\alpha H(\frac{\mathbf{1}}{K}, \mathbf{q})$. Thus:

$$H(\mathbf{s},\mathbf{q}) = (1-\alpha)H(\mathbf{y},\mathbf{q}) + \alpha H\left(\frac{\mathbf{1}}{K},\mathbf{q}\right) \tag{E.6}$$

Rearranging the terms:

$$H(\mathbf{s},\mathbf{q}) = H(\mathbf{y},\mathbf{q}) + \alpha\left(H\left(\frac{\mathbf{1}}{K},\mathbf{q}\right) - H(\mathbf{y},\mathbf{q})\right) \tag{E.7}$$

We can now identify $H(\mathbf{y},\mathbf{q})$ as the original cross-entropy loss, and define the label smoothing loss as:

$$L_{LS} = \alpha\left(H\left(\tfrac{\mathbf{1}}{K},\mathbf{q}\right) - H(\mathbf{y},\mathbf{q})\right).$$

Therefore, we have demonstrated that:

$$H(\mathbf{s},\mathbf{q}) = H(\mathbf{y},\mathbf{q}) + L_{LS} \tag{E.8}$$

with L_{LS} as defined in the lemma. It is noteworthy that the original cross-entropy loss $H(\mathbf{y},\mathbf{q})$ remains unweighted by α in this decomposition, which is consistent with the statement in Lemma 3.2 □

2. Proof of Theorem 3.3

Proof. We aim to prove the equation:

$$L_{LS} = \alpha(z_{gt} - \frac{1}{K}\sum_{k=1}^{K} z_k) \tag{E.9}$$

Let $\mathbf{s}$ be the smoothed label vector and $\mathbf{q}$ be the predicted probability vector. We start with the cross-entropy between $\mathbf{s}$ and $\mathbf{q}$:

$$H(\mathbf{s},\mathbf{q}) = -\sum_{k=1}^{K} s_k \log q_k \tag{E.10}$$

With label smoothing, $s_k = (1-\alpha)y_k + \frac{\alpha}{K}$, where $\mathbf{y}$ is the one-hot ground truth vector and α is the smoothing parameter. Substituting this:

$$H(\mathbf{s},\mathbf{q}) = -\sum_{k=1}^{K}[(1-\alpha)y_k + \frac{\alpha}{K}]\log q_k \tag{E.11}$$

Expanding:

$$H(\mathbf{s},\mathbf{q}) = -(1-\alpha)\sum_{k=1}^{K} y_k \log q_k - \frac{\alpha}{K}\sum_{k=1}^{K}\log q_k \tag{E.12}$$

Since $\mathbf{y}$ is a one-hot vector, $\sum_{k=1}^{K} y_k \log q_k = \log q_{gt}$, where gt is the index of the ground truth class:

$$H(\mathbf{s},\mathbf{q}) = -(1-\alpha)\log q_{gt} - \frac{\alpha}{K}\sum_{k=1}^{K}\log q_k \tag{E.13}$$

Using the softmax function, $q_k = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}}$, we can express $\log q_k$ in terms of logits:

$$\log q_k = z_k - \log(\sum_{j=1}^{K} e^{z_j}) \tag{E.14}$$

Substituting this into our expression:

$$\begin{aligned} H(\mathbf{s}, \mathbf{q}) =& -(1-\alpha)[z_{gt} - \log(\sum_{j=1}^{K} e^{z_j})] \\ & - \frac{\alpha}{K} \sum_{k=1}^{K} [z_k - \log(\sum_{j=1}^{K} e^{z_j})] \\ =& -(1-\alpha) z_{gt} + (1-\alpha) \log(\sum_{j=1}^{K} e^{z_j}) \\ & - \frac{\alpha}{K} \sum_{k=1}^{K} z_k + \alpha \log(\sum_{j=1}^{K} e^{z_j}) \\ =& -(1-\alpha) z_{gt} - \frac{\alpha}{K} \sum_{k=1}^{K} z_k + \log(\sum_{j=1}^{K} e^{z_j}) \end{aligned} \tag{E.15}$$

Rearranging:

$$H(\mathbf{s}, \mathbf{q}) = -z_{gt} + \log(\sum_{j=1}^{K} e^{z_j}) + \alpha[z_{gt} - \frac{1}{K} \sum_{k=1}^{K} z_k] \tag{E.16}$$

We can identify:

- $H(\mathbf{y}, \mathbf{q}) = -z_{gt} + \log(\sum_{j=1}^{K} e^{z_j})$ (cross-entropy for one-hot vector $\mathbf{y}$)
- $L_{LS} = \alpha[z_{gt} - \frac{1}{K} \sum_{k=1}^{K} z_k]$

Thus, we have proven:

$$H(\mathbf{s}, \mathbf{q}) = H(\mathbf{y}, \mathbf{q}) + L_{LS} \tag{E.17}$$

Due to the broad usage of CutMix and Mixup in the training recipe of modern Neural Networks, we additionally take their impact into account together with Label Smoothing. Now we additionally prove the case **with Cutmix and Mixup**:

$$L'_{LS} = \alpha((\lambda z_{gt1} + (1-\lambda) z_{gt2}) - \frac{1}{K} \sum_{k=1}^{K} z_k) \tag{E.18}$$

With Cutmix and Mixup, the smoothed label becomes:

$$s_k = (1-\alpha)(\lambda y_{k1} + (1-\lambda) y_{k2}) + \frac{\alpha}{K} \tag{E.19}$$

where y_{k1} and y_{k2} are one-hot vectors for the two ground truth classes from mixing, and λ is the mixing ratio.

Starting with the cross-entropy:

$$H(\mathbf{s},\mathbf{q}) = -\sum_{k=1}^{K} s_k \log q_k \tag{E.20}$$

$$= -\sum_{k=1}^{K}[(1-\alpha)(\lambda y_{k1} + (1-\lambda)y_{k2}) + \frac{\alpha}{K}] \log q_k \tag{E.21}$$

$$= -(1-\alpha)\sum_{k=1}^{K}(\lambda y_{k1} + (1-\lambda)y_{k2}) \log q_k - \frac{\alpha}{K}\sum_{k=1}^{K}\log q_k \tag{E.22}$$

Since y_{k1} and y_{k2} are one-hot vectors:

$$H(\mathbf{s},\mathbf{q}) = -(1-\alpha)(\lambda \log q_{gt1} + (1-\lambda)\log q_{gt2}) - \frac{\alpha}{K}\sum_{k=1}^{K}\log q_k \tag{E.23}$$

where $gt1$ and $gt2$ are the indices of the two ground truth classes.
Using $q_k = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}}$, we express in terms of logits:

$$H(\mathbf{s},\mathbf{q}) = -(1-\alpha)[\lambda(z_{gt1} - \log(\sum_{j=1}^{K} e^{z_j})) + (1-\lambda)(z_{gt2} - \log(\sum_{j=1}^{K} e^{z_j}))] \tag{E.24}$$

$$- \frac{\alpha}{K}\sum_{k=1}^{K}[z_k - \log(\sum_{j=1}^{K} e^{z_j})] \tag{E.25}$$

Simplifying:

$$H(\mathbf{s},\mathbf{q}) = -(1-\alpha)[\lambda z_{gt1} + (1-\lambda)z_{gt2}] + (1-\alpha)\log(\sum_{j=1}^{K} e^{z_j}) \tag{E.26}$$

$$- \frac{\alpha}{K}\sum_{k=1}^{K} z_k + \alpha\log(\sum_{j=1}^{K} e^{z_j}) \tag{E.27}$$

$$= -(1-\alpha)[\lambda z_{gt1} + (1-\lambda)z_{gt2}] - \frac{\alpha}{K}\sum_{k=1}^{K} z_k + \log(\sum_{j=1}^{K} e^{z_j}) \tag{E.28}$$

Rearranging:

$$H(\mathbf{s},\mathbf{q}) = -[\lambda z_{gt1} + (1-\lambda)z_{gt2}] + \log(\sum_{j=1}^{K} e^{z_j}) \tag{E.29}$$

$$+ \alpha[\lambda z_{gt1} + (1-\lambda)z_{gt2} - \frac{1}{K}\sum_{k=1}^{K} z_k] \tag{E.30}$$

We can identify:

- $H(\mathbf{y}',\mathbf{q}) = -[\lambda z_{gt1} + (1-\lambda)z_{gt2}] + \log(\sum_{j=1}^{K} e^{z_j})$ (cross-entropy for mixed label $\mathbf{y}'$)
- $L'_{LS} = \alpha[\lambda z_{gt1} + (1-\lambda)z_{gt2} - \frac{1}{K}\sum_{k=1}^{K} z_k]$

Thus, we have proven:

$$H(\mathbf{s}, \mathbf{q}) = H(\mathbf{y}', \mathbf{q}) + L'_{LS} \tag{E.31}$$

□

This completes the proof for both cases of Theorem 3.3.

3. Gradient Analysis

In the following, we provide gradient analysis to further support the effectiveness of our proposed MaxSup method.

New Objective Function

The Cross Entropy with Max Suppression is defined as:

$$L_{\text{MaxSup},t}(x, y) = H\left(y_k + \frac{\alpha}{K} - \alpha \cdot \mathbf{1}_{k=\text{argmax}(\boldsymbol{q})}, \boldsymbol{q}_t^S(x)\right)$$

where $H(\cdot, \cdot)$ denotes the cross-entropy function.

The gradient of the loss with respect to the logit z_i for each class i is derived as:

$$\partial_i^{\text{MaxSup},t} = y_{t,i} - y_i - \frac{\alpha}{K} + \alpha \cdot \mathbf{1}_{i=\text{argmax}(\boldsymbol{q})}$$

We analyze this gradient under two scenarios:

Scenario 1: Model makes correct prediction

In this case, Max Suppression is equivalent to Label Smoothing. When the model correctly predicts the target class ($\text{argmax}(\boldsymbol{q}) = \text{GT}$), the gradients are:

- For the target class (GT): $\partial_{\text{GT}}^{\text{MaxSup},t} = q_{t,\text{GT}} - \left(1 - \alpha\left(1 - \frac{1}{K}\right)\right)$
- For non-target classes: $\partial_i^{\text{MaxSup},t} = q_{t,i} - \frac{\alpha}{K}$

Scenario 2: Model makes wrong prediction

When the model incorrectly predicts the most confident class ($\text{argmax}(\boldsymbol{q}) \neq \text{GT}$), the gradients are:

- For the target class (GT): $\partial_{\text{GT}}^{\text{MaxSup},t} = q_{t,\text{GT}} - \left(1 + \frac{\alpha}{K}\right)$
- For non-target classes (not most confident): $\partial_i^{\text{MaxSup},t} = q_{t,i} - \frac{\alpha}{K}$
- For the most confident non-target class: $\partial_i^{\text{MaxSup},t} = q_{t,i} + \alpha\left(1 - \frac{1}{K}\right)$

The Max Suppression regularization technique implements a sophisticated gradient redistribution strategy, particularly effective when the model misclassifies samples. When the model's prediction ($\text{argmax}(\boldsymbol{q})$) differs from the ground truth (GT), the gradient for the incorrectly predicted class is increased by $\alpha(1 - \frac{1}{K})$, resulting in $\partial_{\text{argmax}(\boldsymbol{q})}^{\text{MaxSup},t} = q_{t,\text{argmax}(\boldsymbol{q})} + \alpha(1 - \frac{1}{K})$. Simultaneously, the gradient for the true class is decreased by $\frac{\alpha}{K}$, giving $\partial_{\text{GT}}^{\text{MaxSup},t} = q_{t,\text{GT}} - (1 + \frac{\alpha}{K})$, while for all other classes, the gradient is slightly reduced by $\frac{\alpha}{K}$: $\partial_i^{\text{MaxSup},t} = q_{t,i} - \frac{\alpha}{K}$. This redistribution adds a substantial positive gradient to the misclassified class while slightly reducing the gradients for other classes. The magnitude of this adjustment, controlled by the hyperparameter α, effectively penalizes overconfident errors and encourages the model to focus on challenging examples. By amplifying the learning signal for misclassifications, Max Suppression regularization promotes more robust learning from difficult or ambiguous samples.

Algorithm 1 Gradient Descent with Max Suppression (MaxSup)

```
Require: Training set $D = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{N}$; learning rate $\eta$; number of iterations $T$; smoothing
    parameter $\alpha$; a neural network $f_\theta(\cdot)$; batch size $B$; total classes $K$.
Initialize network weights $\theta$ (e.g., randomly).
for $t = 1$ to $T$ do
    // Each iteration processes mini-batches of size $B$.
    for each mini-batch $\{(\mathbf{x}^{(j)}, \mathbf{y}^{(j)})\}_{j=1}^{B}$ in $D$ do
        Compute logits: $\mathbf{z}^{(j)} \leftarrow f_\theta(\mathbf{x}^{(j)})$ for each sample in the batch
        Compute predicted probabilities: $\mathbf{q}^{(j)} \leftarrow \text{softmax}(\mathbf{z}^{(j)})$
        Compute cross-entropy loss:
            $L_{\text{CE}} \leftarrow \frac{1}{B} \sum_{j=1}^{B} H\big(\mathbf{y}^{(j)}, \mathbf{q}^{(j)}\big)$
        // MaxSup component: penalize the top-1 logit
        For each sample $j$:
            $z_{max}^{(j)} = \max_{k \in \{1,\ldots,K\}} z_k^{(j)}, \quad \bar{z}^{(j)} = \frac{1}{K} \sum_{k=1}^{K} z_k^{(j)}$
            $L_{\text{MaxSup}} \leftarrow \frac{1}{B} \sum_{j=1}^{B} \big[z_{max}^{(j)} - \bar{z}^{(j)}\big]$
        Total loss:
            $L \leftarrow L_{\text{CE}} + \alpha\, L_{\text{MaxSup}}$
        Update parameters:
            $\theta \leftarrow \theta - \eta\, \nabla_\theta\, L$
    end for
end for
```

4. Pseudo Code

Algorithm 1 presents pseudo code illustrating gradient descent with Max Suppression (MaxSup). The main difference from standard Label Smoothing lies in penalizing the highest logit rather than the ground-truth logit.

5. Robustness Under Different Training Recipes

We assess MaxSup's robustness by testing it under a modified training recipe that reduces total training time and alters the learning rate schedule. This setup models scenarios where extensive training is impractical due to limited resources.

Concretely, we adopt the **TorchVision V1 Weight** strategy, reducing the total number of epochs to 90 and replacing the cosine annealing schedule with a step learning-rate scheduler (step size = 30). We also set the initial learning rate to 0.1 and use a batch size of 512. This streamlined recipe aims to reach reasonable accuracy within a shorter duration.

As reported in Table E.1, MaxSup continues to deliver strong performance across multiple convolutional architectures, generally surpassing Label Smoothing and its variants. Although all methods see a performance decline in this constrained regime, MaxSup remains among the top performers, reinforcing its effectiveness across diverse training conditions.

Table E.1.: Performance comparison on ImageNet for various convolutional neural network architectures. Results are presented as "mean ± std" (percentage). **Bold** and underlined entries indicate best and second-best, respectively. (*: implementation details adapted from the official repositories.)

Method	**ResNet-18**	**ResNet-50**	**ResNet-101**	**MobileNetV2**
Baseline	69.11±0.12	76.44±0.10	76.00±0.18	71.42±0.12
Label Smoothing	69.38±0.19	76.65±0.11	77.01±0.15	71.40±0.09
Zipf-LS*	69.43±0.13	76.89±0.17	76.91±0.14	71.24±0.16
OLS*	69.45±0.15	76.81±0.21	77.12±0.17	71.29±0.11
MaxSup	**69.59**±0.13	**77.08**±0.07	**77.33**±0.12	**71.59**±0.17
Logit Penalty	66.97±0.11	74.21±0.16	75.17±0.12	70.249±0.14

6. Increasing Smoothing Weight Schedule

Building on the intuition that a model's confidence naturally grows as training progresses, we propose a linearly increasing schedule for the smoothing parameter α. Concretely, α is gradually raised from an initial value (e.g., 0.1) to a higher value (e.g., 0.2) by the end of training. This schedule aims to counteract the model's increasing overconfidence, ensuring that regularization remains appropriately scaled throughout.

Experimental Evidence As shown in Table E.2, both Label Smoothing and MaxSup benefit from this α scheduler. For Label Smoothing, accuracy improves from 75.91% to 76.16%, while MaxSup sees a more pronounced gain, from 76.12% to 76.58%. This greater improvement for MaxSup (+0.46%) compared to Label Smoothing (+0.25%) corroborates our claim that MaxSup successfully addresses the inconsistent regularization and error-enhancement issues of Label Smoothing during misclassifications.

Table E.2.: Effect of an α scheduler on model performance. Here, t and T denote the current and total epochs, respectively. The baseline model does not involve any label smoothing parameter (α).

Configuration	**Formulation**	$\alpha = 0.1$	$\alpha = 0.1 + 0.1\,\frac{t}{T}$	**Remarks**
Baseline	–	74.21	74.21	α not used
LS	$\alpha\left(z_{gt} - \frac{1}{K}\sum_k z_k\right)$	75.91	76.16	
MaxSup	$\alpha\left(z_{max} - \frac{1}{K}\sum_k z_k\right)$	76.12	**76.58**	

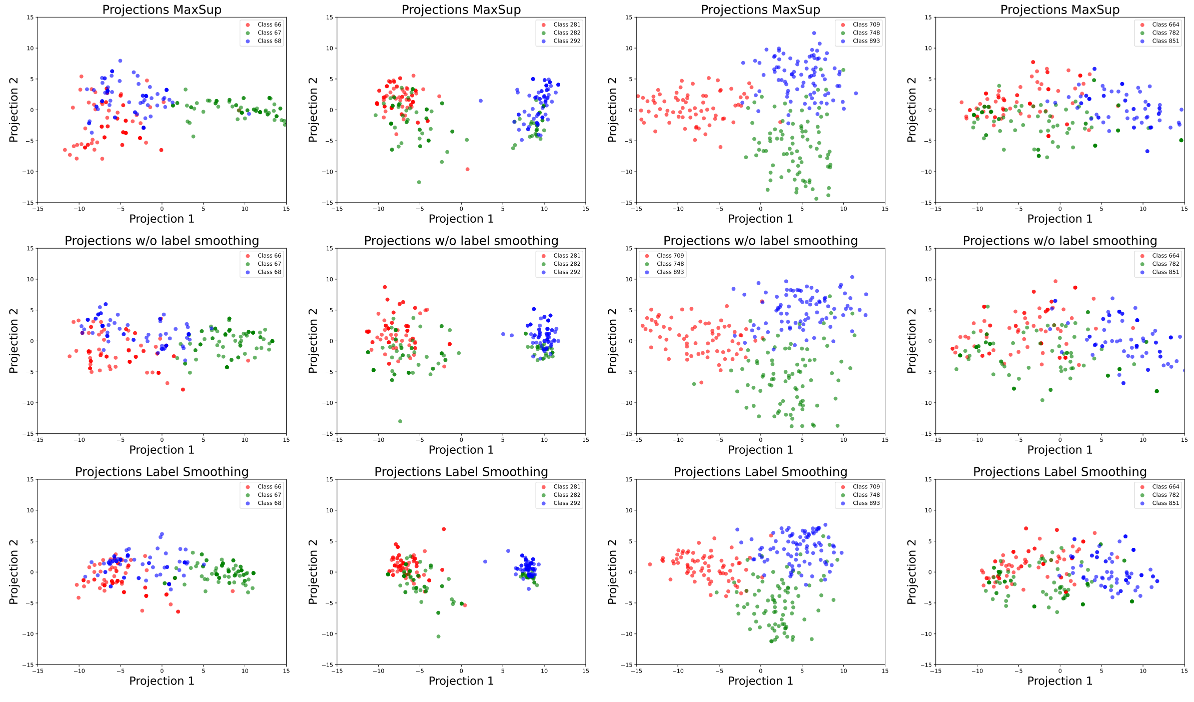
7. Visualization of the Learned Feature Space

To illustrate the differences between Max Suppression Regularization and Label Smoothing, we follow the projection technique of Müller, Kornblith, and Hinton [161]. Specifically, we select three semantically related classes and construct an orthonormal basis for the plane intersecting their class templates in feature space. We then project each sample's penultimate-layer activation vector onto this plane. To ensure the visual clarity of the resulting plots, we randomly sample 80 images from the training or validation set for each of the three classes.

Selection Criteria We choose these classes according to two main considerations:

(i) **Semantic Similarity.** We pick three classes that are visually and semantically close.

(ii) **Confusion.** We identify a class that the Label Smoothing (LS)–trained model frequently misclassifies and select two additional classes involved in those misclassifications (Fig. E.1c, Fig. E.2c). Conversely, we also examine a scenario where a class under Max Suppression is confused with others, highlighting key differences (Fig. E.1d, Fig. E.2d).

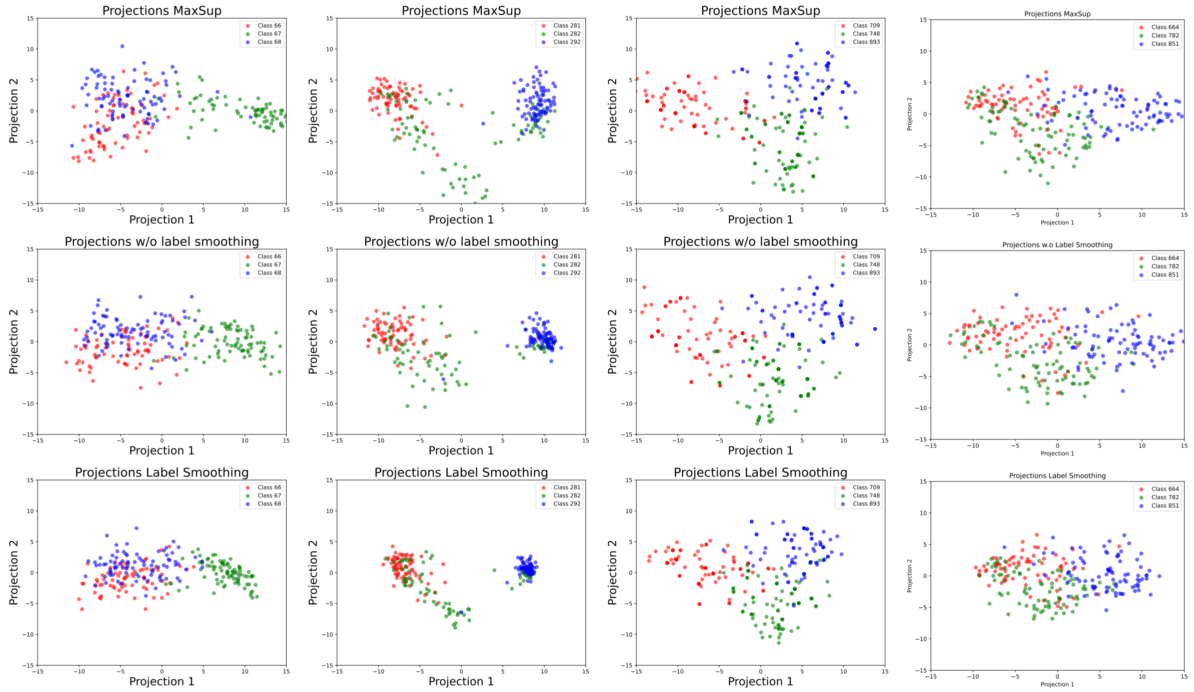


(a) Semantically Similar Classes (b) Semantically Similar Classes (c) Confusing Classes (LS) (d) Confusing Classes (MaxSup)

Figure E.1.: Visualization of penultimate-layer activations from DeiT-Small (trained with CutMix and Mixup) on the ImageNet **validation** set. The top row shows embeddings for a MaxSup-trained model, and the bottom row shows embeddings for a Label Smoothing (LS)–trained model. In each subfigure, classes are either *semantically similar* or *confusingly labeled*. Compared to LS, MaxSup yields more pronounced inter-class separability and richer intra-class diversity, suggesting stronger representation and classification performance.

Observations As shown in Figures E.1 and E.2, models trained with **Max Suppression** exhibit:

- **Enhanced inter-class separability.** Distinct classes occupy more clearly separated regions, aligning with improved classification performance.



(a) Semantically Similar Classes (b) Semantically Similar Classes (c) Confusing Classes (LS) (d) Confusing Classes (MaxSup)

Figure E.2.: Visualization of the penultimate-layer activations for DeiT-Small (trained with CutMix and Mixup) on selected ImageNet classes. The top row shows results for a MaxSup-trained model; the bottom row shows Label Smoothing (LS). In (a,b), the model must distinguish *semantically similar* classes (e.g., Saluki vs. Grey Fox; Tow Truck vs. Pickup), while (c,d) involve *confusing categories* (e.g., Jean vs. Shoe Shop, Stinkhorn vs. related objects). Compared to LS, MaxSup yields both improved inter-class separability and richer intra-class variation, indicating more robust representation learning.

- **Greater intra-class variation.** Instances within a single class are not overly compressed, indicating a richer representation of subtle differences.

For instance, images of *Schipperke* dogs can differ markedly in viewpoint, lighting, background, or partial occlusions. Max Suppression preserves such intra-class nuances in the feature space, enabling the semantic distances to visually related classes (e.g., Saluki, Grey Fox, or Belgian Sheepdog) to dynamically adjust for each image. Consequently, Max Suppression provides a more flexible, fine-grained representation that facilitates better class discrimination.

Table E.3.: Feature representation metrics for a ResNet-50 model trained on ImageNet-1K, reported on both Training and Validation sets. We measure intra-class variation ($\bar{d}_{\text{within}}$) and overall average distance ($\bar{d}_{\text{total}}$). Inter-class separability (R^2) is calculated as $R^2 = 1 - \frac{\bar{d}_{\text{within}}}{\bar{d}_{\text{total}}}$. Higher values (↑) of $\bar{d}_{\text{within}}$ and R^2 are preferred.

Method	$\bar{d}_{\text{within}} \uparrow$		$\bar{d}_{\text{total}}$		$R^2 \uparrow$	
	Train	**Val**	**Train**	**Val**	**Train**	**Val**
Baseline	0.24114	0.24313	0.5212	0.5949	0.4025	0.4451
LS	0.2632	0.2543	0.4862	0.4718	0.4690	0.4611
OLS	0.2707	0.2820	0.6672	0.6570	0.5943	0.5708
Zipf's	0.2611	0.2932	0.5813	0.5628	0.5522	0.4790
MaxSup	**0.2926 (+0.03)**	**0.2998 (+0.05)**	0.6081 (+0.12)	0.5962 (+0.12)	0.5188 (+0.05)	0.4972 (+0.04)
Logit Penalty	0.2840	0.24144	0.7996	0.7909	0.6448	0.6024

Appendix F.

Balancing Diversity and Risk in Sampling-Based Decoding for Large Language Models

In this section, we provide the supplementary materials for Chapter VIII.

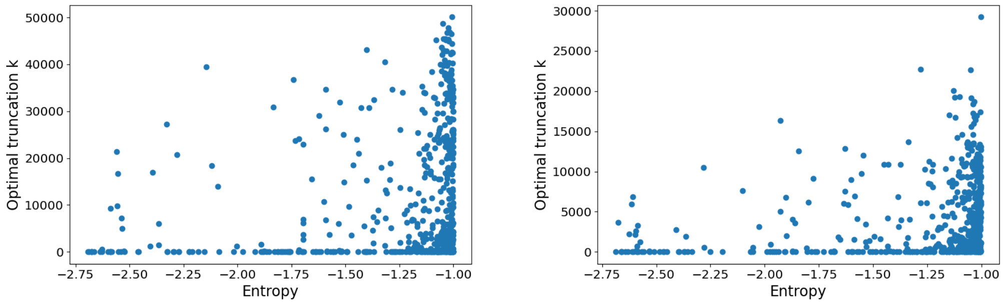
1. Complete Record of the Experiment Runs

Methods	Evaluation Runs									Mean/Std		
	Run 1 at average Risks			Run 2 at average Risks			Run 3 at average Risks			average Risks		
	1	5	15	1	5	15	1	5	15	1	5	15
Greedy Decoding	0.338											
Naive Sampling	0.420			0.426			0.416			0.421(0.004)		
Top-k Sampling	0.412	0.447	0.410	0.389	0.432	0.435	0.402	0.428	0.419	0.401(0.010)	0.436(0.008)	0.421(0.010)
Top-p Sampling	0.337	0.370	0.382	0.367	0.393	0.379	0.362	0.370	0.405	0.355(0.013)	0.378(0.011)	0.389(0.012)
Adaptive Sampling	0.403	0.416	0.433	0.403	0.416	0.419	0.378	0.440	0.411	0.395(0.012)	0.424(0.011)	0.421(0.009)
Eta Sampling	0.395	0.419	0.442	0.387	0.394	0.419	0.382	0.389	0.379	0.388(0.005)	0.401(0.013)	0.413(0.026)
Mirostat	0.424	0.417	0.430	0.399	0.443	0.433	0.415	0.414	0.412	0.413(0.010)	0.425(0.013)	0.425(0.009)

Table F.1.: Evaluation on the TruthfulQA benchmark. Since the GPT-3 API is no longer available, we use the by the authors recommended BLEURT accuracy for comparison under the open-ended generation setup.

The scores of the individual runs on TruthfulQA benchmark are recorded in Table F.1, and the means and standard errors of recalls and risks at all average Risks are listed in Table F.2. Note that due to a fixed amount of computation budget, we search the corresponding parameter value for each truncation sampling method till the average risk is close enough to the predefined value, thus resulting in the variations of the average risks. However, such variations are negligible given the minor differences.

Although Top-p sampling is indeed also adaptive regarding the truncation position, we show that Top-p sampling have a inherent limitation. When a larger portion of the probability mass is concentrated in the first few tokens (this often indicates smaller entropy), a fixed cumulative probability threshold will cut a longer tail off, and vice versa. However, there's merely a weak correlation between the entropy of the LLM's prediction and optimal truncation values, see Figure F.1.



(a) The Pearson's correlation is 0.24777 for GPT2-XL.

(b) The Pearson's correlation is 0.24784 for Llama-2-7B.

Figure F.1.: Scatter plots between the entropy values and optimal truncation values.

2. The Advantage of Probability-Independent Metrics

In this section, we explain the practical advantages of our proposed probability-independent recall and risk metrics. As can be seen in Figure F.2, the empirical distribution aligns with the by gpt2-xl predicted distribution given the same prefix in general: most of the tokens which posses high likelihood in the prediction also has a high probability based on the word frequencies of our collected CP-Trie data. However, there exists two differences:

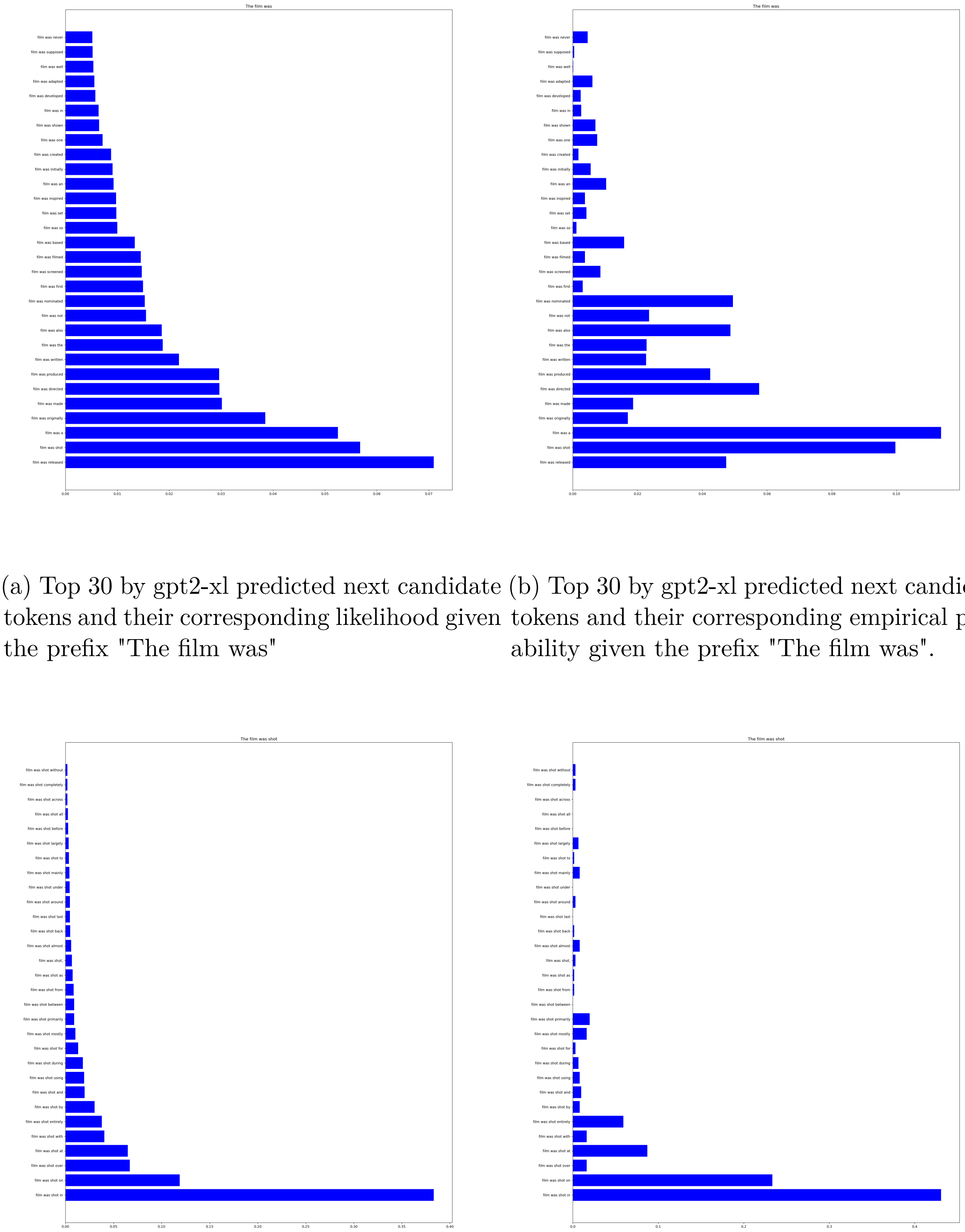
- Some tokens with high likelihood according to gpt2-xl have much lower probability according to the empirical distribution. The ranking of each tokens w.r.t. probability also differ in the two distributions.
- A few tokens which should be reasonable candidates (by manual check) have 0 probability according to the empirical distribution.

For the first issue, as discussed in Section 3.2, there exists no ideal probabilities for each token, and the discrepancy is not solvable by simply increasing the size of the data. For example, the "perfect" probabilities of the candidate tokens "with" and "at" are undefined and could even be regarded as equivalently important for open-ended text generation.

The second difference highlights the reliability of LLMs, i.e., the tokens which are assigned high likelihoods are in most cases reasonable. Note that we ignore the risk within the estimated optimal allowed set by design: All the tokens are counted as reasonable till the last token which has non-zero empirical probability, when they are arranged in a descending order according to the predicted probabilities. Thus these tokens with zero probabilities in the empirical distribution will not affect our evaluation of risk, making our method robust to noises and insufficient data support.

Method					GPT2-XL				
	Parameter	Risk	Recall	Parameter	Risk	Recall	Parameter	Risk	Recall
Top-k	15	1.029 (0.006)	0.220(0.0006)	64	5.040 (0.613)	0.290 (0.017)	184	14.983(1.781)	0.340 (0.018)
Top-p	0.5705	0.999 (0.015)	0.170 (0.0005)	0.746	5.011(2.129)	0.240 (0.015)	0.8555	15.022 (6.210)	0.338 (0.016)
Adaptive	9.5e-4	1.000 (0.006)	0.252 (0.0007)	0.00011	4.997 (0.679)	0.339(0.018)	2.5e-05	14.995 (2.241)	0.413 (0.018)
Eta	0.318	1.000 (0.013)	0.198 (0.0005)	0.011	4.945 (1.484)	0.301 (0.016)	0.001	14.998 (4.261)	0.404 (0.017)
Mirostat	4.425	0.999 (0.005)	0.236 (0.0007)	5.9475	5.001 (0.717)	0.326 (0.018)	6.76	14.982 (2.501)	0.401 (0.018)
Method					Llama-2-7b				
	Parameter	Risk	Recall	Parameter	Risk	Recall	Parameter	Risk	Recall
Top-k	14	0.986 (0.126)	0.226 (0.016)	61	4.987 (0.587)	0.296 (0.017)	177	14.961 (1.722)	0.369 (0.018)
Top-p	0.54	0.999 (0.529)	0.156 (0.012)	0.7665	4.990 (2.331)	0.254 (0.015)	0.9	14.989 (6.208)	0.400 (0.016)
Adaptive	0.0011	1.051 (0.154)	0.257 (0.016)	0.00014	4.991 (0.856)	0.364 (0.017)	3.1e-5	14.995 (2.966)	0.470 (0.017)
Eta	0.512	1.000 (0.563)	0.192 (0.014)	0.023	5.007 (2.599)	0.297 (0.016)	0.002	13.487 (6.531)	0.407 (0.017)
Mirostat	4.253	1.000 (0.133)	0.236 (0.016)	5.82	4.993 (0.650)	0.349 (0.018)	6.628	15.022 (2.286)	0.474 (0.017)
Method					Llama-3-8B				
	Parameter	Risk	Recall	Parameter	Risk	Recall	Parameter	Risk	Recall
Top-k	14	1.023 (0.128)	0.228 (0.016)	59	4.982 (0.576)	0.290 (0.017)	172	15.025 (1.701)	0.346 (0.018)
Top-p	0.5395	1.000 (0.451)	0.154 (0.013)	0.736	4.998 (2.061)	0.224 (0.014)	0.855	14.993 (5.770)	0.326 (0.016)
Adaptive	0.0011	1.133 (0.167)	0.260 (0.017)	0.00017	5.006 (0.787)	0.343 (0.018)	3.7e-5	15.007 (2.685)	0.418 (0.018)
Eta	0.673	1.000 (0.445)	0.181 (0.014)	0.029	5.009 (2.112)	0.271 (0.016)	0.002	15.012 (6.009)	0.373 (0.017)
Mirostat	4.24	1.001 (0.139)	0.230 (0.016)	5.8175	5.001 (0.804)	0.318 (0.018)	6.6925	14.996 (2.630)	0.393 (0.018)
Method					Llama-3-70B				
	Parameter	Risk	Recall	Parameter	Risk	Recall	Parameter	Risk	Recall
Top-k	14	1.014 (0.127)	0.230 (0.016)	60	5.038 (0.581)	0.295 (0.017)	173	15.024 (1.695)	0.352 (0.018)
Top-p	0.5695	1.001 (0.502)	0.158 (0.013)	0.758	4.999 (2.386)	0.237 (0.015)	0.8705	14.960 (6.685)	0.332 (0.016)
Adaptive	0.0011	1.004 (0.137)	0.263 (0.017)	0.00014	5.013 (0.787)	0.353 (0.018)	3.16e-5	14.986 (2.778)	0.424 (0.018)
Eta	0.37	1.004 (0.137)	0.263 (0.017)	0.014	5.032 (2.231)	0.295 (0.016)	0.001	15.076 (6.265)	0.398 (0.018)
Mirostat	4.21	1.001 (0.138)	0.230 (0.016)	5.91	5.001 (0.708)	0.332 (0.018	6.84	15.021 (2.193)	0.417 (0.018)
Method					Llama-2-70b				
	Parameter	Risk	Recall	Parameter	Risk	Recall	Parameter	Risk	Recall
Top-k	14	1.002 (0.128)	0.232 (0.016)	60	4.982 (0.583)	0.307 (0.017)	174	14.964 (1.712)	0.375 (0.018)
Top-p	0.6535	0.999 (0.475)	0.189 (0.013)	0.8465	4.988 (2.136)	0.316 (0.016)	0.9395	15.019 (5.522)	0.468 (0.016)
Adaptive	0.0011	1.000 (0.142)	0.269 (0.017)	1.2e-4	4.995 (0.796)	0.374 (0.017)	2.3e-5	15.007 (2.697)	0.485 (0.017)
Eta	0.092	1.002 (0.304)	0.236 (0.015)	0.003	5.057 (1.590)	0.378 (0.017)	0.00021	15.001 (4.243)	0.510 (0.017)
Mirostat	4.16	1.001 (0.135)	0.238 (0.016	5.7875	5.004 (0.684)	0.353 (0.018)	6.67	14.991 (2.125)	0.478 (0.017)
Method					Mixtral-8x7B				
	Parameter	Risk	Recall	Parameter	Risk	Recall	Parameter	Risk	Recall
Top-k	15	1.028 (0.134)	0.229 (0.016)	63	4.978 (0.598)	0.301 (0.017)	183	14.967 (1.757)	0.366 (0.018)
Top-p	0.6505	1.000 (0.535)	0.192 (0.014)	0.8375	5.007 (2.423)	0.303 (0.015)	0.9325	14.966 (6.139)	0.456 (0.016)
Adaptive	0.00105	1.000 (0.148)	0.265 (0.017)	0.00011	4.994 (0.798)	0.372 (0.018)	2.1e-5	15.014 (2.802)	0.476 (0.017)
Eta	0.087	1.001 (0.335)	0.241 (0.015)	0.003	5.061 (1.822)	0.375 (0.017)	0.000215	14.991 (4.922)	0.506 (0.017)
Mirostat	4.2775	1.000 (0.143)	0.238 (0.016)	5.845	4.995 (0.710)	0.346 (0.018)	6.6875	14.998 (2.213)	0.461 (0.018)
Method					Mistral-7B				
	Parameter	Risk	Recall	Parameter	Risk	Recall	Parameter	Risk	Recall
Top-k	14	0.965 (0.126)	0.224 (0.016)	62	4.968 (0.596)	0.297 (0.017)	181	15.006 (1.759)	0.364 (0.018)
Top-p	0.6565	1.001 (0.539)	0.194 (0.014)	0.8375	4.996 (2.476)	0.303 (0.016)	0.9315	15.038 (6.315)	0.447 (0.016)
Adaptive	0.00105	1.001 (0.152)	0.260 (0.016)	0.000115	4.993 (0.809)	0.364 (0.018)	2.2e-5	14.999 (2.757)	0.466 (0.017)
Eta	0.075	0.997 (0.307)	0.243 (0.015)	0.003	4.640 (1.542)	0.368 (0.017)	0.000196	15.009 (4.712)	0.505 (0.017)
Mirostat	4.1825	1.000 (0.141)	0.236 (0.016)	5.8125	4.999 (0.721)	0.345 (0.018)	6.71	14.978 (2.213)	0.468 (0.018)

Table F.2.: Critical Parameters of different truncation sampling methods at different average Risks using different models.



(a) Top 30 by gpt2-xl predicted next candidate tokens and their corresponding likelihood given the prefix "The film was"

(b) Top 30 by gpt2-xl predicted next candidate tokens and their corresponding empirical probability given the prefix "The film was".

(c) Top 30 by gpt2-xl predicted next candidate tokens and their corresponding likelihood given the prefix "The film was shot".

(d) Top 30 by gpt2-xl predicted next candidate tokens and their corresponding empirical probability given the prefix "The film was shot".

Figure F.2.: Comparing the probabilities predicted by gpt2-xl and calculated using the word frequencies based on our collected CP-Trie data.