SimReg: Achieving Higher Performance in the Pretraining via Embedding Similarity Regularization

Yan Sun[12*], **Guoxia Wang**[1*], **Jinle Zeng**[1], **Jiabin Yang**[1], **Shuai Li**[1],
Li Shen[3], **Dacheng Tao**[4†], **Dianhai Yu**[1†], **Haifeng Wang**[1]
[1]Baidu Inc. [2]The University of Sydney
[3]Shenzhen Campus of Sun Yat-sen University [4]Nanyang Technological University
{sunyan25,wangguoxia,zengjinle,yangjiabin01,yudianhai,wanghaifeng}@baidu.com
lishuai_math@163.com, mathshenli@gmail.com, dacheng.tao@ntu.edu.sg

Abstract

Pretraining large language models (LLMs) with next-token prediction has led to remarkable advances, yet the context-dependent nature of token embeddings in such models results in high intra-class variance and inter-class similarity, thus hindering the efficiency of representation learning. While similarity-based regularization has demonstrated benefit in supervised fine-tuning and classification tasks, its application and efficacy in large-scale LLM pretraining remain underexplored. In this work, we propose the SIMREG, an embedding similarity regularization loss that explicitly encourages token representations with the same ground-truth label within each sequence to be more similar, while enforcing separation from different-label tokens via a contrastive loss. Our analysis reveals that this mechanism introduces gains by enlarging multi-classification margins, thereby enabling more efficient classification. Extensive experiments across dense and Mixture-of-Experts (MoE) architectures demonstrate that SIMREG consistently accelerates training convergence by over 30% and improves average zero-shot downstream performance by over 1% across standard benchmarks. Further ablation studies and analyses offer practical insights into hyperparameter tuning and loss effectiveness.

1 Introduction

LLMs have emerged as a cornerstone of modern artificial intelligence and have demonstrated remarkable capabilities across a wide range of domains such as natural language understanding [Radford et al., 2019], reasoning [Wei et al., 2022], and multimodal interaction [Lin et al., 2025b]. While LLMs are advancing along diverse directions, they all fundamentally share a consistent underlying principle, i.e., next-token prediction. The essential mechanism of LLMs is to predict the categorical distribution of the next token from the embeddings of the prior context, which can also be viewed as a classification problem defined over the combined representations of the preceding context. By leveraging enormous model parameters and vast training data, it exhibits exceptional generalization capability, introduces novel solutions in diverse research domains, and further drives the adoption of a wide range of applications [Topsakal and Akinci, 2023] with growing challenges in efficiency [Shen et al., 2024]. Both data-specific [Fan et al., 2025, Deng et al., 2026] and weight-specific [Li et al., 2024a, Sun et al., 2025, Lin et al., 2025a] approaches have attracted considerable research interest.

Unlike conventional classification, language model prediction does not rely on a stable object strictly tied to its label. In image classification, for instance, a cat image is consistently associated with its label, leading to highly consistent embeddings within the same class. In contrast, on language tasks,

*Equal contributions. This work was conducted during Yan Sun's internship at Baidu Inc.
†Corresponding authors: Dianhai Yu and Dacheng Tao.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

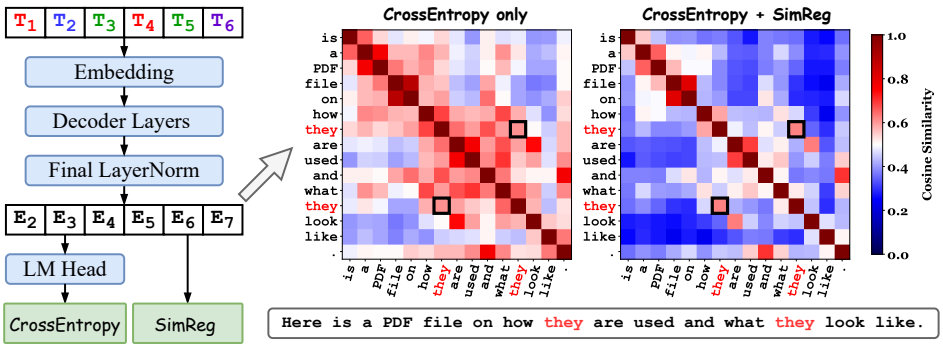


Figure 1: (left) Workflow of the SIMREG loss. (Right) We compare the cosine similarity of token embeddings in a sample on the LLaMA-7B model trained via "CrossEntropy only" and "CrossEntropy+SIMREG". Using "CrossEntropy only" fails to enforce sufficient separability among token features, whose cosine values of all token pairs almost exceed 0.5. With the introduction of SIMREG, feature separability is generally enhanced (averaged cosine value is reduced by at least 0.1), thereby providing stronger support for classification training. More results are stated in Appendix A.4.

the representation used to predict a token is composed of diverse contextual features, many unrelated to the label itself. As a result, embeddings predicting the same token can vary significantly. For example, the representations for "walls" in "The cat jumps over walls" and "A child paints near walls" originate from entirely different contexts, making the classification process more challenging.

Recent advances in consistency learning for finetuning language models shed light on potential solutions to this challenge [Huang et al., 2021, Gunel et al., 2021, Yin et al., 2023]. However, this line of research has not yet been extended to pretraining and has not been widely adopted in the large-scale pretraining practices. Post-training is typically performed with a small learning rate and limited datasets, which makes it difficult to significantly modify the geometric structure of the learned parameters. These insights motivate us to further extend this approach to large-scale pretraining.

In this work, we show that large-scale pretraining with cross-entropy alone fails to impose strong consistency on token embeddings. To address this, we then add a consistency regularization term, SIMREG, to strengthen the representational capacity of large models during pretraining. For each token in a sequence, all tokens are partitioned into positive and negative groups. The objective penalizes the similarity across groups, which pulls embeddings toward same-class samples and pushes them away from different-class samples. To ensure valid contrastive pairs for every token, SIMREG introduces self-sample similarity in each positive group and further computes the loss with group-level rather than sample-level averaging, which balances the contributions of different tokens, which allows it to preserve a high level of stability over the long pretraining runs. We also provide a thorough theoretical understanding to explain how it contributes to improving cross-entropy loss. Extensive evaluations are conducted on both dense and MoE models, including LLaMA-350M, 1.3B, 3B, 7B [Touvron et al., 2023], and Mixtral-8$\times$1B [Jiang et al., 2024]. The SIMREG loss can consistently accelerate convergence by over 30% in pretraining. When training with over 52B tokens, it can yield an improvement of more than 1% in average performance across downstream general tasks. We investigate the hyperparameter sensitivity of SIMREG and find that it maintains a wide range of applicability. We summarize our main contributions as follows:

- We explore the advantages of employing consistency regularization in large-scale pretraining tasks and propose a series of improvements to address the training instabilities of existing methods, thereby enabling stable performance gains throughout long-term pretraining.
- We provide a detailed theoretical analysis of the benefits of the SIMREG loss for the cross-entropy loss, and how it improves the multi-classification margins.
- We conduct extensive experiments to validate its substantial improvements for pretraining tasks, achieving an average training acceleration of over 30% and yielding over 1% gains on downstream tasks, and state detailed empirical insights for the community.

2 Related Work

Contrastive learning. The systematic exploration of feature similarity constraints in machine learning can be traced back to their early development in computer vision (CV) tasks and contrastive learning [Oord et al., 2018, Khosla et al., 2020]. They enhance the training of baseline classification models by constructing virtual data pairs and incorporating additional supervised loss signals, which helped the models extract more discriminative features. It is typically employed to counteract noise perturbations at the input level, thereby improving generalization ability [Geng et al., 2021, Shi et al., 2022, Huang and Gong, 2022, Zhou et al., 2024, Wang et al., 2024]. Generally, a data pair is constructed from a raw sample and its perturbed counterpart, and the model is trained to maximize their representation similarity. Subsequently, supervised contrastive learning has been extended to incorporate class information. By leveraging available labels to construct class-consistent data pairs, the model is trained not only to pull together samples from the same class but also to push apart samples from different classes [Wang and Liu, 2021, Wen and Li, 2021, Ye et al., 2022, Denize et al., 2023]. Recent studies have revealed that contrastive learning can also achieve more efficient feature extraction across tasks and data originating from different domains [Verma et al., 2021, Wang et al., 2022, Xie et al., 2022, Azuma et al., 2023]. In multimodal large model training, this learning paradigm is often employed to align the mapping of knowledge across domains and to capture the representation capacity of the same knowledge under different modalities [Yuan et al., 2021, Mai et al., 2022, Liu et al., 2024b, Sun et al., 2024]. In summary, contrastive learning offers an efficient and general paradigm for representation learning to the machine learning community.

Embedding Consistency in LLMs. The study of feature similarity has also been considered as compositional generalization [Lake, 2019, Wiedemer et al., 2023] and embedding consistency regularization [Yin et al., 2023]. Gao et al. [2021] learn the sentence embeddings and achieve higher generalization efficiency. Then it is widely expanded to the token-level [Gao et al., 2023, Wang and Yu, 2023], word-level [Kenter and De Rijke, 2015, Antoniak and Mimno, 2018], context-level [Laskar et al., 2020]. Most of these tasks have primarily focused on small-scale or fine-tuning settings. As the cornerstone of modern language models, the next-token prediction paradigm has been widely applied across various downstream tasks [Li et al., 2024b, Chen et al., 2024]. Recent research has further investigated the similarity and dispersion of token embeddings, which highlights the separability of embeddings to be a key direction [de Andrade et al., 2023, Tao et al., 2024, Hu et al., 2024].

3 Problem Setup and Methodology

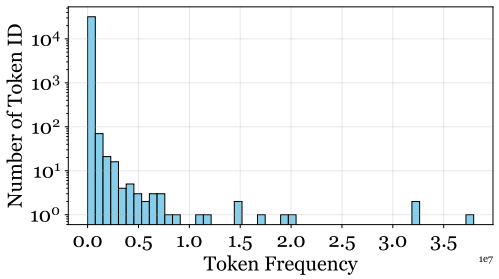
In this section, we introduce how SIMREG can be incorporated into the pretraining of LLMs and explain why it helps improve performance. Before proceeding, we formalize the overall pretraining setup of LLMs and introduce the notations used throughout the subsequent analysis.

General Pretraining. Before introducing the training framework, we first define the notation in this work. We consider the progress of LLM pretraining as learning the optimal weight $\mathbf{w}$ by minimizing the cross-entropy loss ℓ under a general distribution $\mathcal{D}$. We decompose the model into two cascaded functions $f_P \circ f_E$, where f_P (the logits generation module) is parameterized by $\mathbf{w}_P$ and f_E (the embedding generation module) is parameterized by $\mathbf{w}_E$, with the overall parameters denoted as $\mathbf{w} = [\mathbf{w}_P, \mathbf{w}_E]$. Based on this decomposition, the general pretraining objective of language models can then be formally formulated as:

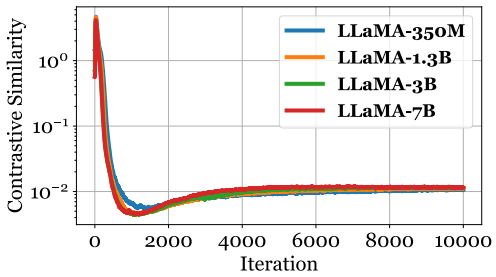
$$\min_{\mathbf{w}} \mathbb{E}_{(\mathbf{x}_i, y_i)\sim\mathcal{D}} \left[\ell\left(f_P \circ f_E\left(\mathbf{x}_i\right), y_i\right)\right], \tag{1}$$

where $(\mathbf{x}_i, y_i)$ is the $(\text{data}, \text{label})$ pair sampled from the distribution $\mathcal{D}$. Here, the choice of f_E and f_P is entirely flexible, meaning that the SIMREG loss can in principle be applied to any valid token embedding across the network. We further explore the optimal placement of this component in subsequent experiments of Sec. 5.2.

Cross-entropy loss serves as the fundamental training objective in language modeling. It measures the discrepancy between the predicted token distribution and the ground-truth one-hot distribution, thereby guiding the model to maximize the likelihood of the correct next token. The models typically employ large-scale feature extractors to obtain separable representations. By denoting the token embedding as $\mathbf{e}_i = f_E(\mathbf{x}_i)$ and corresponding logits as $\mathbf{z}_i = f_P(\mathbf{e}_i)$, the empirical risk of sample-



(a) Imbalanced Token Frequency in Pretraining.



(b) Limited Contrastive Similarity in Pretraining.

Figure 2: (a) We analyze the token ID distribution over 1B training samples from the C4 dataset and find that only about 2% of tokens occur with extremely high frequency, resulting in a pronounced long-tail effect in the classification data. (b) We observe that the contrastive similarity loss of embeddings does not continue to decrease after reaching a basic threshold and then the feature similarity is no longer further optimized. **Simply increasing the model size does not improve this performance.**

wise cross-entropy loss is:

$$L^{\text{ce}} = \frac{1}{n}\sum_{i}\left(-\mathbf{z}_{i,y_i} + \log\left(\sum_{j}\exp\left(\mathbf{z}_{i,j}\right)\right)\right). \tag{2}$$

Generally, larger separability can enhance the distinction between different samples, leading to more robust and discriminative representations. Although Eq. (2) averages over samples, the unique characteristics of language tasks introduce a challenge: *the distribution of words (tokens) is highly imbalanced*, which causes frequent tokens to dominate the loss while rare but informative ones contribute disproportionately little, yielding a heavy long-tail dataset. When training classification tasks on such dataset, the inter-class margin is greatly influenced by the number of samples per class. As shown in Figure 2 (a), we empirically investigate the token distribution of C4 dataset and the behavior of contrastive similarity. **A primary challenge we investigate in the LLM pretraining is:**

*cross-entropy **stops** driving stronger representation learning after a basic separability level of tokens.*

Figure 2.(b) indicates that during the early stage of cross-entropy training, the model rapidly constrains the contrastive similarity of embeddings. However, once the contrastive diversity becomes sufficient to sustain classification training, the model no longer enforces heterogeneity among token embeddings. Subsequently, even though the cross-entropy loss continues to decrease, the contrastive similarity exhibits little further change. Another interesting phenomenon we observe is that, even as the model depth increases and the embedding dimension grows, the supervision of token embedding contrastive similarity under cross-entropy remains nearly at the same level. This limits the potential for further improvement in classification tasks, while also motivating us to impose the contrastive similarity.

Embedding Similarity Regularization. Here, we introduce the generalized form of our similarity regularization. For each token $\mathbf{x}$, its embedding can be denoted by $\mathbf{e} = f_E(\mathbf{x})$. For each data sample $(\mathbf{x}_i, y_i)$, we can define a positive embedding set $\mathcal{P}_i = \{k : y_k = y_i\}$ and a negative embedding set $\mathcal{N}_i = \{k : y_k \neq y_i\}$. The consistency loss aims to minimize the distance between embeddings of positive pairs, while simultaneously maximizing the separation between negative pairs:

$$L_i^{\text{sr}} \triangleq \log\sum_{j\in\mathcal{N}_i}\phi_{i,j} - \log\sum_{j\in\mathcal{P}_i}\phi_{i,j}, \tag{3}$$

where L_i^{sr} is the similarity loss of i-th token. ϕ denotes a similarity function between two embeddings. We explore two primary forms: the exponential of the inner-product $\langle \mathbf{e}_i, \mathbf{e}_j\rangle$ and that of the cosine similarity $\frac{\langle \mathbf{e}_i, \mathbf{e}_j\rangle}{\|\mathbf{e}_i\|\cdot\|\mathbf{e}_j\|}$. Both similarity measures provide effective supervision for feature similarity, yet their applicable scenarios differ. It often yields stronger statistical constraints, thereby enforcing supervision on both geometric structure and feature norms. However, this advantage may also introduce ambiguity: for instance, when a embedding has an abnormally large norm, the inner-product value becomes dominated by the magnitude, rendering the loss function almost insensitive to

angular differences. In such cases, the optimization may overly rely on vector norms while neglecting the discriminative power of directional alignment. Therefore, for numerical stability, we adopt cosine similarity as the similarity measure in Eq. (3) and introduce a constant temperature coefficient τ to adjust the sharpness of the distribution. Since words in natural language are inherently distributed in an imbalanced manner, a sequence may contain only a single occurrence of a particular token type, we add the self-similarity to $\mathcal{P}_i$ to enforce that there exists at least a positive data pair. Moreover, to ensure that the regularization loss is non-negative, we introduce the `softplus` function to further scale it. Therefore, the final form of the loss is computed as $L_i = L_i^{\text{ce}} + \lambda \cdot \text{softplus}\left(L_i^{\text{sr}}\right)$. The entire optimization process involves two hyperparameters τ and λ. We discuss them in the experiments.

Chunk-wise SIMREG for Sequence Parallelism. The computation of SIMREG is centered on the embedding of each token within a sequence sample, whose complexity is $\mathcal{O}(n^2)$. Its computation can naturally support parallelization strategies like data parallelism (DP), tensor parallelism (TP), and pipeline parallelism (PP). However, during long-text training, sequence parallelism (SP) splits the data of each sequence across different nodes for training, which introduces additional redundant communications. To alleviate this issue, we divide SIMREG into b chunks, where every $\frac{n}{b}$ tokens form a chunk to compute the SIMREG loss internally. The losses across different nodes are then weighted according to the ratio of positive and negative samples, while the overall computational complexity is reduced to $\mathcal{O}(n^2/b)$. Moreover, we further point out that there exists a fundamental trade-off between the strength of supervision and the expressive capacity of feature representations with respect to the choice of chunk size. When a chunk contains a larger number of tokens, the estimation of SIMREG becomes more accurate. However, its constraining power on each individual token is weakened, as the loss must balance relationships among a larger set of tokens. Therefore, selecting an appropriate chunk size is of critical importance, which is empirically explored in Sec. 5.1.

4 A Theoretical Perspective on Embedding Similarity

In this section, we demonstrate how SIMREG improves the lower bound of the classification margins. Although this does not directly establish its superiority, it provides a theoretical perspective for understanding the potential advantages of feature clustering. We first introduce classification margins, i.e., $m_i = \mathbf{z}_{i,y_i} - \max_{j \neq y_i} \mathbf{z}_{i,j}$, which is the gap between the top two logits. Then the cross entropy loss can be upper bounded by:

$$\ell_i = \log\left(1 + \sum_{j \neq y_i} \exp\left(\mathbf{z}_{i,j} - \mathbf{z}_{i,y_i}\right)\right) \leq C \exp\left(-m_i\right), \tag{4}$$

where C is the number of total classes. The above formula explicitly characterizes the relationship between the classification margin and the training loss, and enlarging the margin m leads to a further reduction in the loss. Our supervision on the embedding variable $\mathbf{e}$ is propagated through a function $f_P(\cdot)$ to the logits $\mathbf{z}$ used for the classification with cross-entropy, i.e., $\mathbf{z} = f_P(\mathbf{e})$. This mapping can take the form of a simple linear projection (e.g., the LM head) or several intermediate layers of the LLM. Without loss of generality, we assume it to be a general Lipschitz and non-convex function with smoothness coefficient L_P. Thus, we consider the margin. By defining $\mathbb{I}$ as the standard basis vector where $\mathbb{I}_j$ means 1 in the j-th coordinate and 0 elsewhere, we measure the pair-wise gap in logits by $g_{y_k,j}(\mathbf{e}_k) = \left(\mathbb{I}_{y_k} - \mathbb{I}_j\right)^\top f_P(\mathbf{e}_k)$, which also holds smoothness and non-convexity. Furthermore, we can transfer the property by: $|g_{y_i,j}(\mathbf{e}_p) - g_{y_i,j}(\mathbf{e}_q)| \leq \|\mathbb{I}_{y_i} - \mathbb{I}_j\| \|f_P(\mathbf{e}_p) - f_P(\mathbf{e}_q)\| \leq \sqrt{2} L_P \|\mathbf{e}_p - \mathbf{e}_q\|$. To investigate their relationships, we have the following lemma.

Lemma 1 *For each token $\mathbf{x}_i$ where its embedding is $\mathbf{e}_i = f_E(\mathbf{x}_i)$, we further define a weighted center of the embedding in the original space, where the positive and negative centers are $\overline{\mathbf{e}}_k^+ = \frac{\sum_{i \in \mathcal{P}_k} \alpha_{k,i} \mathbf{e}_i}{\sum_{i \in \mathcal{P}_k} \alpha_{k,i}}$ and $\overline{\mathbf{e}}_k^- = \frac{\sum_{i \in \mathcal{N}_k} \alpha_{k,i} \mathbf{e}_i}{\sum_{i \in \mathcal{N}_k} \alpha_{k,i}}$ where $\alpha_{k,i} \propto \exp\left(\mathbf{e}_k^\top \mathbf{e}_i\right)$. Then we have the averaged group margins are $\overline{m}_k^+ = \min_{c \neq y_k} g_{y_k,c}(\overline{\mathbf{e}}_k^+)$ and $\overline{m}_k^- = \min_{c \neq y_k} g_{y_k,c}(\overline{\mathbf{e}}_k^-)$. Therefore, the classification margin bound of each token m_k is the Central–Eccentric lower bound within the group margin:*

$$\overline{m}_k^+ - \sqrt{2} L_P \|\mathbf{e}_k - \overline{\mathbf{e}}_k^+\| \leq m_k \leq \overline{m}_k^- + \sqrt{2} L_P \|\mathbf{e}_k - \overline{\mathbf{e}}_k^-\|. \tag{5}$$

Intuitively, $\overline{m}$ can be regarded as an idealized margin, obtained by evaluating the logit of the correct class at the positive center and that of the strongest competing class at the negative center. A key

point is that it separates the upper and lower bounds of the classification margin for each individual sample, showing that the lower bound is influenced by the distance to positive samples $\|\mathbf{e}_k - \overline{\mathbf{e}}_k^+\|$ and $\overline{m}_k^+$, while the upper bound is determined by the distance to negative samples $\|\mathbf{e}_k - \overline{\mathbf{e}}_k^-\|$ and $\overline{m}_k^-$. We can implicitly show the following points:

- The dynamics of the central distance of the positive set would decrease: $\frac{d}{dt}\|\mathbf{e}_k - \overline{\mathbf{e}}_k^+\|^2 \leq 0$;
- The classification margin at the positive center would increase: there exists a positive constant δ that
$$g_{y_k,j}(\mathbf{e}_k^+ + \epsilon_+) - g_{y_k,j}(\mathbf{e}_k^+) \geq \delta\|\epsilon_+\|, \tag{6}$$
where ϵ_+ is the perturbation caused by similarity loss.

By minimizing the objective L^{sr}, the positive center $\overline{\mathbf{e}}_k^+$ shifts its weights toward same-class samples that are more similar to the anchor $\mathbf{e}_k$, causing $\|\mathbf{e}_k - \overline{\mathbf{e}}_k^+\|$ to decrease γ. Simultaneously the positive group margin can increase at least $\delta\|\epsilon_+\|$. Therefore, the classification margin of the k-th token can improve at least $m'_k \geq m_k + \delta\|\epsilon_+\| + \sqrt{2}L_P\gamma$. Therefore, the cross-entropy loss will decrease at least by $\ell'_k \leq \ell_k \cdot \exp\left(-\left(\delta\|\epsilon_+\| + \sqrt{2}L_P\gamma\right)\right)$, which can also accelerate the pretraining process.

5 Experiments

In this section, we show the empirical studies of the proposed SIMREG loss. We primarily investigate the advantages in pretraining tasks, including its acceleration on the training loss, improvements of the evaluation on the downstream tasks, and influence the dynamics of the embedding similarity during training. We also examine its sensitivity to hyperparameters and its behavior. Moreover, we explore the practical effects of inserting the SIMREG loss at different positions within the model. These experiments can provide useful technical guidance for the community.

Model Backbones. We mainly select LLaMA [Touvron et al., 2023] and Mixtral [Jiang et al., 2024] as the dense and MoE backbones for pretraining, including the core modules of the mainstream models in the current community, e.g. for RoPE [Su et al., 2024], RMSNorm [Zhang and Sennrich, 2019], and SwiGLU [Shazeer, 2020]. We conduct experiments on dense models with 350M, 1.3B, 3B, and 7B parameters, and on the MoE model with 8B parameters.

Training Hyperparameters. We follow the experimental setups reported in several recent classical LLM pretraining studies [Touvron et al., 2023, Liu et al., 2024a, Jiang et al., 2024, Baidu-ERNIE-Team, 2025] to configure the baseline hyperparameters. We employ the AdamW optimizer [Loshchilov and Hutter, 2019] with $\beta_1 = 0.9$, $\beta_2 = 0.95$, and let the weight decay equals to 0.1. The standard deviation of the weight initialization is set to 0.01. The global batch size is set to 512 for the 350M and MoE-8$\times$1B models, and 2048 for the 1.3B, 3B, and 7B dense models. The input sequence length is fixed to 2048. For the learning rate schedule, we adopt a 2000-step warm-up phase to linearly increase the learning rate from 0 to 3×10^{-4}, followed by a cosine decay strategy that gradually reduces it to one-tenth of its peak value. For dense models, we train about 13B tokens for the 350M model and 52B tokens for the larger dense models. For MoE models, we train approximately 52B tokens. To avoid loss spikes, we adopt the AdaGC [Wang et al., 2025] to clip gradients for all methods. Other details are stated in the Appendix.

Baselines. We select the Simple Contrastive Sentence Embedding Gao et al. [2021] (SimCSE), Contrastive Pretraining Neelakantan et al. [2022] (CPretrain), Consistency Regularization Yin et al. [2023] (CReg), Similarity Contrastive Estimation Denize et al. [2023] (SCE). SimCSE adopts the contrastive loss on the sentence embedding. CPretrain minimizes the similarity distribution. CReg treats each token pair as an independent negative example. SCE adopts a weighted similarity via latent distributions. The above works are not all designed for pretraining; however, they share certain conceptual similarities. In our experiments, we uniformly adapt them to the pretraining framework.

5.1 Empirical Studies on Performance

Higher Generalization. Table 1 reports the zero-shot generalization performance on various downstream tasks across four model scales. Overall, existing consistency-based baselines bring only marginal or inconsistent improvements in average accuracy, with some methods even showing slight performance degradation. In contrast, our method achieves the highest average accuracy across all

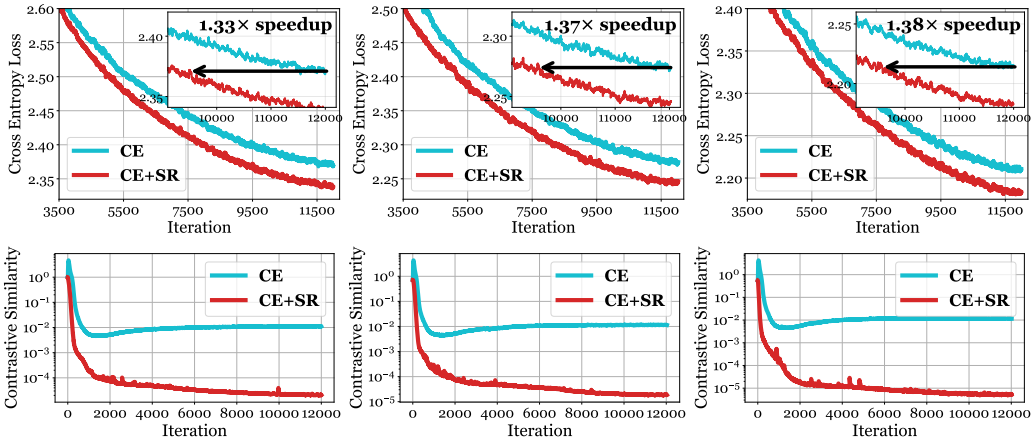
Table 1: Generalization performance comparisons: Zero-shot evaluations on the downstream tasks.

	Arc-E	Arc-C	BoolQ	HellaS.	Obqa	Piqa	Mmlu	WinoG.	Sciq	Avg.
LLaMA-350M	38.64	22.95	57.09	36.49	28.40	66.51	22.95	51.30	63.20	43.06
◦ SimCSE	38.75	23.02	57.15	36.61	28.50	66.59	23.03	51.42	63.30	43.15
◦ CPretrain	38.23	22.88	56.62	36.14	28.10	65.86	22.86	50.85	64.20	42.86
◦ CReg	38.91	23.26	57.49	36.83	28.70	66.91	23.23	51.78	64.10	43.47
◦ SCE	39.14	23.43	57.80	37.17	28.90	67.20	23.48	51.64	64.20	43.66
◦ SIMREG	40.15	24.49	57.51	37.66	29.40	68.26	22.94	52.13	64.40	44.10
LLaMA-1.3B	46.21	25.09	58.01	49.60	31.80	72.14	23.07	52.83	68.90	47.52
◦ SimCSE	46.34	25.25	58.08	49.44	31.70	72.05	23.26	52.89	68.80	47.53
◦ CPretrain	46.16	25.04	57.96	49.55	31.60	72.09	23.18	52.72	68.70	47.44
◦ CReg	46.41	26.29	58.31	49.90	32.10	72.62	23.25	53.13	69.00	47.89
◦ SCE	46.71	25.59	58.41	50.10	32.10	72.58	23.37	53.55	69.60	48.00
◦ SIMREG	46.51	26.79	61.01	52.51	30.40	72.95	24.02	54.23	69.50	48.66
LLaMA-3B	48.91	27.30	58.29	55.67	33.00	74.16	23.65	55.49	73.50	50.00
◦ SimCSE	49.21	27.35	58.61	55.87	33.10	74.16	23.65	55.54	73.30	50.09
◦ CPretrain	49.04	27.32	58.39	55.72	33.20	74.27	23.45	55.53	73.70	50.07
◦ CReg	49.50	27.36	59.20	56.10	33.20	74.08	23.42	55.55	73.80	50.25
◦ SCE	49.76	27.34	60.18	56.94	33.20	74.03	23.34	55.56	73.30	50.41
◦ SIMREG	50.59	28.07	58.65	57.65	33.40	74.32	23.95	56.67	75.30	50.96
LLaMA-7B	53.07	28.84	54.07	60.41	33.80	76.12	23.79	57.30	75.70	51.45
◦ SimCSE	52.45	28.62	57.78	60.28	34.40	75.35	23.41	57.58	74.80	51.63
◦ CPretrain	51.87	27.95	58.10	60.55	34.20	75.42	23.35	57.17	75.80	51.60
◦ CReg	51.95	28.10	58.65	60.75	34.10	75.20	23.50	57.45	76.60	51.81
◦ SCE	52.30	28.45	58.40	60.90	34.50	75.30	23.60	57.80	75.30	51.84
◦ SIMREG	52.57	29.01	59.79	62.01	35.80	75.14	24.47	59.04	76.20	52.67

four model scales, with improvements of 1.04, 1.14, 0.96, and 1.22 percentage points over the original LLaMA-350M, LLaMA-1.3B, LLaMA-3B, and LLaMA-7B models, respectively. Compared with the strongest baseline at each model scale, our method also maintains a clear advantage in average accuracy. Although the improvements vary across individual tasks, the overall gains remain consistent across different model sizes. These results suggest that the proposed similarity regularization can improve downstream generalization beyond what is achieved by existing consistency-based methods.

Higher Convergence. We first demonstrate the training acceleration of SIMREG in large-scale pretraining tasks. As shown in Figure 3 upper part, on the 1.3B model, the speedup can reach nearly 33%, and after training on 52B tokens, the cross entropy loss can be reduced by about 0.05. On larger-scale models, including the 3B model and the 7B model, SIMREG achieves more than 37% speedup when training reaches 52B tokens, with the final training loss reduced by about 0.03. In the lower part, we present the SIMREG loss. It can be observed that cross-entropy does not impose a mandatory constraint on feature similarity. When training with cross-entropy alone, the feature similarity undergoes a rapid decline in the early stage, and then gradually tends to stabilize. At this point, the network no longer additionally learns to accelerate classification training by enhancing feature separability. An interesting phenomenon we observe is that, when trained solely with cross-entropy, the similarity regularization value for almost all networks eventually converges to around 0.01, which implies that the average angle between words of different classes is approximately 61.3 degrees. After introducing the SIMREG loss, the embedding similarity decreases significantly, with the regularization loss converging to about 0.00001, indicating that the average angle achieves approximately 74 degrees among tokens.

Chunk-wise SIMREG vs. Full SIMREG. Table 2 shows that chunk-wise SIMREG achieves comparable performance to Full SIMREG, even outperforming it on the 7B model. With larger chunks, the expressive capacity of the SIMREG loss becomes limited, as the effective supervisory signal for each token weakens when too many tokens are involved. This suggests a trade-off between loss expressiveness and supervision strength, which becomes more pronounced as model size increases. Larger models have higher-dimensional hidden states, where embedding angles tend to



(a) LLaMA-1.3B. (b) LLaMA-3B. (c) LLaMA-7B.

Figure 3: Cross-entropy loss acceleration (upper) and contrastive similarity improvements (lower) in the pretraining. "CE" denotes the cross-entropy loss and "SR" denotes the similarity regularization loss. SIMREG loss helps to further reduce the contrastive similarity.

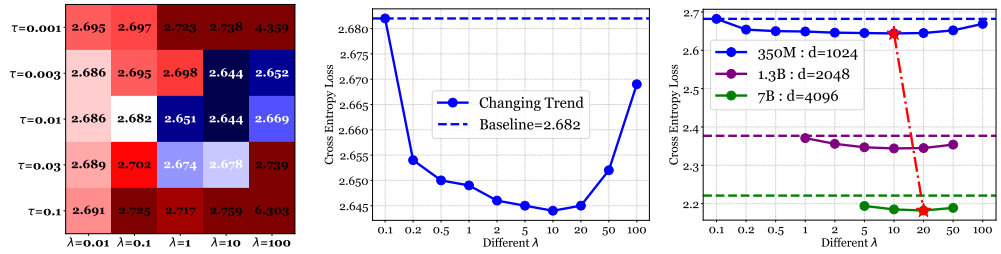
Table 2: Full v.s. Chunk-wise: Zero-shot evaluations on the general downstream tasks.

	Arc-E	Arc-C	BoolQ	HellaS.	Obqa	Piqa	Mmlu	WinoG.	Sciq	Avg.
LLaMA-350M	38.64	22.95	57.09	36.49	28.40	66.51	22.95	51.30	63.20	43.06
○ SIMREG	40.15	24.49	57.51	37.66	29.40	68.26	22.94	52.13	64.40	44.10
○ SIMREG-Chunk	39.77	24.23	58.14	37.25	29.40	67.59	23.02	51.84	64.40	43.96
LLaMA-1.3B	46.21	25.09	58.01	49.60	31.80	72.14	23.07	52.83	68.90	47.52
○ SIMREG	46.51	26.79	61.01	52.51	30.40	72.95	24.02	54.23	69.50	48.66
○ SIMREG-Chunk	46.80	26.11	59.17	51.94	31.80	72.25	23.12	54.78	69.00	48.33
LLaMA-3B	48.91	27.30	58.29	55.67	33.00	74.16	23.65	55.49	73.50	50.00
○ SIMREG	50.59	28.07	58.65	57.65	33.40	74.32	23.95	56.67	75.30	50.96
○ SIMREG-Chunk	50.80	27.39	62.48	58.49	33.60	73.88	22.95	55.64	73.20	50.94
LLaMA-7B	53.07	28.84	54.07	60.41	33.80	76.12	23.79	57.30	75.70	51.45
○ SIMREG	52.57	29.01	59.79	62.01	35.80	75.14	24.47	59.04	76.20	52.67
○ SIMREG-Chunk	51.60	29.69	62.39	61.80	35.80	75.46	23.51	58.72	76.00	52.77
Mixtral-8×1B	48.86	29.18	54.62	59.57	34.00	73.88	24.17	56.99	72.40	50.41
○ SIMREG	51.81	28.75	60.03	62.53	35.00	75.08	23.59	54.30	74.10	51.69
○ SIMREG-Chunk	52.04	28.98	60.26	62.76	35.23	73.88	23.82	54.53	73.10	51.62

increase. In such spaces, the number of participating tokens has a greater influence on the quality of similarity-based supervision for individual tokens. Therefore, chunk-wise SIMREG offers an effective alternative to full SIMREG for large-scale model pretraining.

Scaling to Large Models. Table 2 reports the performance on larger-scale models. Overall, introducing the SIMREG loss consistently improves the average performance from 350M to 7B. SIMREG can bring +1.14% average improvement on LLaMA-1.3B, +0.96% on LLaMA-3B, +1.22% on LLaMA-7B, and +1.28% on Mixtral-8×1B. These results highlight that SIMREG provides stable and non-trivial gains as the model scale increases. Moreover, SIMREG achieves the largest single-task gain of +5.72% on BoolQ with LLaMA-7B. Besides BoolQ, we also observe clear improvements on HellaSwag, WinoGrande, and SciQ across multiple scales, showing that it is particularly effective for reasoning-heavy and multi-choice tasks. These consistent improvements further suggest that SIMREG is a simple yet broadly applicable strategy for the large-scale pretraining to enhance generalization.

Hyperparameter Sensitivity. We first grid search (τ, λ) on the 350M model to identify a valid range, followed by a fine-grained search to determine their optimal combinations. Subsequently, we conduct scaling experiments on the 1.3B and 7B models to examine how the optimal choices



(a) Grid Search of (τ, λ). (b) Fine-grained Search on λ. (c) Scaling with Model Size.

Figure 4: (a) Grid search over hyperparameters τ and λ. The blue blocks indicate the values where the final training loss under the corresponding combination (τ, λ) is lower than baseline, with darker colors representing lower losses. (b) We further conduct a fine-grained search over different λ values at the generally optimal $\tau = 0.01$, using an approximate $2\times$ scaling ratio. (c) We explore the trends on different λ across different model sizes (the red line indicates the optimal trend).

Table 3: Changing trend of CrossEntropy and SIMREG loss on different λ.

	$\lambda = 1$	$\lambda = 2$	$\lambda = 5$	$\lambda = 10$	$\lambda = 20$	$\lambda = 50$
CrossEntropy	2.38	2.35	2.32	**2.30**	2.33	2.35
SIMREG ($\times$1e-6)	104.6	56.2	18.3	9.5	6.0	3.1

vary as the model size increases and the corresponding token embedding dimension grows. As shown in Figure 4 (a), to explore the stable results, we grid search the temperature coefficient τ from $[0.001, 0.003, 0.01, 0.03, 0.1]$ with a $3\times$ skip, and coarsely choose the coefficient λ from $[0.01, 0.1, 1, 10, 100]$ with a $10\times$ skip. The valid range for τ is relatively limited, with 0.01 proving to be a robust selection for all models. λ spans a broad effective range from 0.1 to 100. Figure 4 (b) presents a fine-grained exploration of λ, varying it from 0.1 to 100 with roughly $2\times$ resolution. The results reveal a stable region between 2 and 20. In Figure 4 (c), we explore the scaling of hyperparameters and infer from results across different model sizes how to select optimal hyperparameters. Specifically, when the embedding dimension increases, each token is represented in a higher-dimensional space. Therefore, it becomes necessary to increase λ to maintain training efficiency. Our experiments confirm this trend, and current results suggest that every time the embedding dimension doubles, the optimal hyperparameter increases by approximately a factor of $\sqrt{2}$.

Trade-off of λ. Although we generally hope that greater feature separability will lead to better performance, the pretraining process involves not only learning representations but also learning classification. If λ is increased without bound, the weight of SIMREG may eventually become too dominant and interfere with the optimization of cross-entropy. This phenomenon can be directly observed from the changes in gradient behavior, which provide an intuitive reflection of the trade-off between the two objectives. Table 3 shows the comparison clearly illustrates the effects of cross-entropy and SIMREG under different parameter settings.

5.2 Optimal Position of Adopting SIMREG

In this part, we empirically investigate at which positions in the model embedding supervision yields the best results. We divide the network according to its natural layer-wise structure and apply supervision at different depths. As shown in Figure 5, supervision on intermediate layers brings only negligible performance gains. This is expected, as token representations in the middle layers are not simply tied to independent word meanings; instead, they encode blended contextual information aggregated from preced-

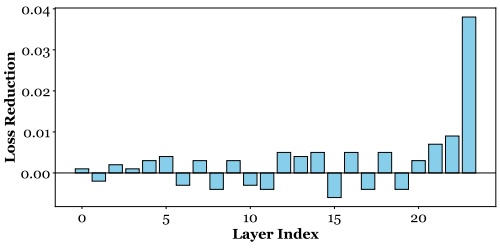


Figure 5: Loss changes of adopting our SIMREG loss at different layers on 1B model.

ing tokens. Enforcing similarity regularization on such entangled representations may therefore provide limited useful signal. In contrast, the final layers gradually project these broad contextual representations into more distinct semantic spaces that are directly used for next-token prediction. Our experiments further show that applying SIMREG only at the last layer is sufficient to achieve efficient pretraining.

5.3 Runtime and Memory Consumptions

We evaluate the training efficiency of our method on a 7B-scale model with a token embedding dimension of 4096. For SIMREG-Chunk, we use a chunk size of 1024 to further reduce the computational footprint. All reported statistics are collected on H800 GPUs, and memory usage is measured by the maximum GPU memory allocation. As shown in Table 4, incorporating the full SIMREG loss results in less than a 2% increase in runtime and under a 1% increase in GPU memory consumption. In comparison, SIMREG-Chunk introduces only negligible computational and memory overhead, making it effectively resource-neutral in practice. These results show that SIMREG delivers meaningful performance gains with minimal additional training cost, highlighting its practicality as a lightweight and effective auxiliary component for large-scale pretraining.

Table 4: Training profiles in Time and Memory.

	Time (s / iter)	Memory (GB)
w/o SIMREG	2.295	373.45
w/ SIMREG	2.296	375.28
w/ SIMREG-Chunk	2.295	373.84

6 Conclusion

In this work, we introduced SIMREG, a similarity regularization loss for large-scale pretraining. We show that cross-entropy alone does not sufficiently enforce embedding consistency, whereas SIMREG strengthens representation learning by aligning same-class tokens while separating different classes. Experiments on both dense and MoE models demonstrate that SIMREG consistently accelerates convergence by more than 30% and improves downstream performance by over 1%. Moreover, it remains robust across different model scales and hyperparameter settings, indicating its practical applicability. These findings highlight consistency regularization as a promising direction for improving the efficiency and generalization of LLM pretraining.

Limitations. Our evaluation focuses on selected LLaMA and Mixtral models. The effectiveness of SIMREG at larger scales and across other architectures and data distributions remains to be investigated. Moreover, the optimal chunk size may vary across models and training settings.

Broader Impact. SIMREG has the potential to reduce the computational resources required for language model pretraining. Models trained with SIMREG remain subject to risks such as biased outputs, misinformation, and misuse, highlighting the need for appropriate safety evaluation and responsible deployment.

References

Maria Antoniak and David Mimno. Evaluating the stability of embedding-based word similarities. *Transactions of the Association for Computational Linguistics*, 6:107–119, 2018.

Chiori Azuma, Tomoyoshi Ito, and Tomoyoshi Shimobaba. Adversarial domain adaptation using contrastive learning. *Engineering Applications of Artificial Intelligence*, 123:106394, 2023.

Baidu-ERNIE-Team. Ernie 4.5 technical report, 2025.

Liang Chen, Zekun Wang, Shuhuai Ren, Lei Li, Haozhe Zhao, Yunshui Li, Zefan Cai, Hongcheng Guo, Lei Zhang, Yizhe Xiong, et al. Next token prediction towards multimodal intelligence: A comprehensive survey. *arXiv preprint arXiv:2412.18619*, 2024.

Claudio MV de Andrade, Fabiano M Belem, Washington Cunha, Celso França, Felipe Viegas, Leonardo Rocha, and Marcos Andre Goncalves. On the class separability of contextual embeddings representations–or "the classifier does not matter when the (text) representation is so good!". *Information Processing & Management*, 60(4):103336, 2023.

Xun Deng, Han Zhong, Rui Ai, Fuli Feng, Zheng Wang, and Xiangnan He. Less is more: Improving llm alignment via preference data selection. *Advances in Neural Information Processing Systems*, 38:161259–161285, 2026.

Julien Denize, Jaonary Rabarisoa, Astrid Orcesi, Romain Hérault, and Stéphane Canu. Similarity contrastive estimation for self-supervised soft contrastive learning. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2706–2716, 2023.

Ziqing Fan, Yuqiao Xian, Yan Sun, and Li Shen. Joint selection for large-scale pre-training data via policy gradient-based mask learning. *arXiv preprint arXiv:2512.24265*, 2025.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024. URL https://zenodo.org/records/12608602.

Pengzhi Gao, Ruiqing Zhang, Zhongjun He, Hua Wu, and Haifeng Wang. An empirical study of consistency regularization for end-to-end speech-to-text translation. *arXiv preprint arXiv:2308.14482*, 2023.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*, 2021.

Yuanling Geng, Shuqun Li, Fan Zhang, Shaowu Zhang, Liang Yang, and Hongfei Lin. Context-aware and data-augmented transformer for interactive argument pair identification. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 579–589. Springer, 2021.

Beliz Gunel, Jingfei Du, Alexis Conneau, and Veselin Stoyanov. Supervised contrastive learning for pre-trained language model fine-tuning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=cu7IUiOhujH.

Jun Hu, Wenwen Xia, Xiaolu Zhang, Chilin Fu, Weichang Wu, Zhaoxin Huan, Ang Li, Zuoli Tang, and Jun Zhou. Enhancing sequential recommendation via llm-based semantic embedding learning. In *Companion Proceedings of the ACM Web Conference 2024*, pages 103–111, 2024.

Hao-Gan Huang and Yue-Jiao Gong. Contrastive learning: An alternative surrogate for offline data-driven evolutionary computation. *IEEE Transactions on Evolutionary Computation*, 27(2):370–384, 2022.

Qiushi Huang, Tom Ko, Lilian Tang, Xubo Liu, and Boyong Wu. Token-level supervised contrastive learning for punctuation restoration. In *Interspeech*, 2021. URL https://api.semanticscholar.org/CorpusID:236134216.

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. *arXiv preprint arXiv:2401.04088*, 2024.

Tom Kenter and Maarten De Rijke. Short text similarity with word embeddings. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pages 1411–1420, 2015.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

Brenden M Lake. Compositional generalization through meta sequence-to-sequence learning. *Advances in neural information processing systems*, 32, 2019.

Md Tahmid Rahman Laskar, Xiangji Huang, and Enamul Hoque. Contextualized embeddings based transformer encoder for sentence similarity modeling in answer selection task. In *Proceedings of the twelfth language resources and evaluation conference*, pages 5505–5514, 2020.

Jinhao Li, Jiaming Xu, Shiyao Li, Shan Huang, Jun Liu, Yaoxiu Lian, and Guohao Dai. Fast and efficient 2-bit llm inference on gpu: 2/4/16-bit in a weight matrix with asynchronous dequantization. In *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*, pages 1–9, 2024a.

Yingcong Li, Yixiao Huang, Muhammed E Ildiz, Ankit Singh Rawat, and Samet Oymak. Mechanics of next token prediction with self-attention. In *International Conference on Artificial Intelligence and Statistics*, pages 685–693. PMLR, 2024b.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Guangxuan Xiao, and Song Han. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *GetMobile: Mobile Computing and Communications*, 28(4):12–17, 2025a.

Zihao Lin, Samyadeep Basu, Mohammad Beigi, Varun Manjunatha, Ryan A Rossi, Zichao Wang, Yufan Zhou, Sriram Balasubramanian, Arman Zarei, Keivan Rezaei, et al. A survey on mechanistic interpretability for multi-modal foundation models. *arXiv preprint arXiv:2502.17516*, 2025b.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024a.

Rui Liu, Haolin Zuo, Zheng Lian, Björn W Schuller, and Haizhou Li. Contrastive learning based modality-invariant feature acquisition for robust multimodal emotion recognition with missing modalities. *IEEE Transactions on Affective Computing*, 15(4):1856–1873, 2024b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Sijie Mai, Ying Zeng, Shuangjia Zheng, and Haifeng Hu. Hybrid contrastive learning of tri-modal representation for multimodal sentiment analysis. *IEEE Transactions on Affective Computing*, 14(3):2276–2289, 2022.

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, et al. Text and code embeddings by contrastive pre-training. *arXiv preprint arXiv:2201.10005*, 2022.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Li Shen, Yan Sun, Zhiyuan Yu, Liang Ding, Xinmei Tian, and Dacheng Tao. On efficient training of large-scale deep learning models. *ACM Computing Surveys*, 57(3):1–36, 2024.

Lida Shi, Fausto Giunchiglia, Rui Song, Daqian Shi, Tongtong Liu, Xiaolei Diao, and Hao Xu. A simple contrastive learning framework for interactive argument pair identification via argument-context extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10027–10039, 2022.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Lijing Sun, Mengyi Zhang, Yu Lu, Wenjun Zhu, Yang Yi, and Fei Yan. Nodule-clip: Lung nodule classification based on multi-modal contrastive learning. *Computers in Biology and Medicine*, 175:108505, 2024.

Yan Sun, Qixin Zhang, Zhiyuan Yu, Xikun Zhang, Li Shen, and Dacheng Tao. Maskpro: Linear-space probabilistic learning for strict (n: M)-sparsity on large language models. *arXiv preprint arXiv:2506.12876*, 2025.

Chongyang Tao, Tao Shen, Shen Gao, Junshuo Zhang, Zhen Li, Kai Hua, Wenpeng Hu, Zhengwei Tao, and Shuai Ma. Llms are also effective embedding models: An in-depth overview. *arXiv preprint arXiv:2412.12591*, 2024.

Oguzhan Topsakal and Tahir Cetin Akinci. Creating large language model applications utilizing langchain: A primer on developing llm apps fast. In *International conference on applied engineering and natural sciences*, volume 1, pages 1050–1056, 2023.

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. *arXiv preprint arXiv:2307.09288*, 2023.

Vikas Verma, Thang Luong, Kenji Kawaguchi, Hieu Pham, and Quoc Le. Towards domain-agnostic contrastive learning. In *International Conference on Machine Learning*, pages 10530–10541. PMLR, 2021.

Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2495–2504, 2021.

Guoxia Wang, Shuai Li, Congliang Chen, Jinle Zeng, Jiabin Yang, Tao Sun, Yanjun Ma, Dianhai Yu, and Li Shen. Adagc: Improving training stability for large language model pretraining. *arXiv preprint arXiv:2502.11034*, 2025.

Hongwei Wang and Dong Yu. Going beyond sentence embeddings: A token-level matching algorithm for calculating semantic textual similarity. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 563–570, 2023.

Rui Wang, Zuxuan Wu, Zejia Weng, Jingjing Chen, Guo-Jun Qi, and Yu-Gang Jiang. Cross-domain contrastive learning for unsupervised domain adaptation. *IEEE Transactions on Multimedia*, 25:1665–1673, 2022.

Yifei Wang, Jizhe Zhang, and Yisen Wang. Do generated data always help contrastive learning? *arXiv preprint arXiv:2403.12448*, 2024.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Zixin Wen and Yuanzhi Li. Toward understanding the feature learning process of self-supervised contrastive learning. In *International Conference on Machine Learning*, pages 11112–11122. PMLR, 2021.

Thaddäus Wiedemer, Prasanna Mayilvahanan, Matthias Bethge, and Wieland Brendel. Compositional generalization from first principles. *Advances in Neural Information Processing Systems*, 36:6941–6960, 2023.

Ruobing Xie, Qi Liu, Liangdong Wang, Shukai Liu, Bo Zhang, and Leyu Lin. Contrastive cross-domain recommendation in matching. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 4226–4236, 2022.

Yuntong Ye, Changfeng Yu, Yi Chang, Lin Zhu, Xi-Le Zhao, Luxin Yan, and Yonghong Tian. Unsupervised deraining: Where contrastive learning meets self-similarity. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5821–5830, 2022.

Yongjing Yin, Jiali Zeng, Yafu Li, Fandong Meng, Jie Zhou, and Yue Zhang. Consistency regularization training for compositional generalization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1294–1308, 2023.

Xin Yuan, Zhe Lin, Jason Kuen, Jianming Zhang, Yilin Wang, Michael Maire, Ajinkya Kale, and Baldo Faieta. Multimodal contrastive training for visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6995–7004, 2021.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in neural information processing systems*, 32, 2019.

Peilin Zhou, You-Liang Huang, Yueqi Xie, Jingqi Gao, Shoujin Wang, Jae Boum Kim, and Sunghun Kim. Is contrastive learning necessary? a study of data augmentation vs contrastive learning in sequential recommendation. In *Proceedings of the ACM Web Conference 2024*, pages 3854–3863, 2024.

Acknowledgments. This work was supported by the National Engineering Research Center of Deep Learning Technology and Application.

A Appendix: Experiments

A.1 Experimental Setups

Here we present the detailed experimental setups in this paper to ensure the reproducibility.

Model Hyperparameters. We mainly select LLaMA2 [Touvron et al., 2023] and Mixtral [Jiang et al., 2024] as the dense and MoE backbones for pretraining, including the core modules of the mainstream models in the current community, e.g. for RoPE [Su et al., 2024], RMSNorm [Zhang and Sennrich, 2019], and SwiGLU [Shazeer, 2020]. We follow the common practices in the community to scale models of different sizes, and the detailed configurations are shown in Table 5.

Table 5: Model Hyperparameters.

	Experts	Layers	Attention heads	Embedding dim	FFN hidden size
LLaMA2-350M	1	24	16	1024	2371
LLaMA2-1.3B	1	24	32	2048	5461
LLaMA2-3B	1	26	32	3072	8640
LLaMA2-7B	1	32	32	4096	11008
Mixtral-8×1B	8	24	32	2048	5632

Training Hyperparameters. We follow the experimental setups reported in several recent classical LLM pretraining studies [Touvron et al., 2023, Liu et al., 2024a, Jiang et al., 2024] to configure the baseline hyperparameters, ensuring comparability with prior work. Specifically, we employ the AdamW optimizer [Loshchilov and Hutter, 2019] with $\beta_1 = 0.9$ and $\beta_2 = 0.95$, and a weight decay of 0.1. The standard deviation of weight initialization is set to 0.01. To balance efficiency and stability, we use a global batch size of 512 for the 350M and MoE-8×1B models, and 2048 for the 1.3B, 3B, and 7B dense models, while the input sequence length is fixed at 2048.

For the learning rate schedule, we adopt a 2000-step warm-up phase that linearly increases the learning rate from 0 to $3e$-4, followed by a cosine decay strategy that gradually reduces it to one-tenth of its peak value. Regarding training length, dense models are trained for $12,500$ steps, corresponding to roughly 13B tokens for the 350M model and 52B tokens for the larger dense models. In contrast, MoE models are trained for $50,000$ steps to ensure comparable exposure of approximately 52B tokens. Finally, to mitigate potential instabilities caused by loss spikes, we adopt AdaGC [Wang et al., 2025] for adaptive gradient clipping. We summarize the details in Table 6.

Table 6: Training Hyperparameters.

	batchsize	seqlen	learning rate	λ_w	β_1	β_2	clip-λ	clip-β
LLaMA-350M	512	2048	4e-4 → 4e-5	0.1	0.9	0.95	1.04	0.99
LLaMA-1.3B	2048	2048	3e-4 → 3e-5	0.1	0.9	0.95	1.04	0.99
LLaMA-3B	2048	2048	3e-4 → 3e-5	0.1	0.9	0.95	1.04	0.99
LLaMA-7B	2048	2048	3e-4 → 3e-5	0.1	0.9	0.95	1.04	0.99
Mixtral-8×1B	512	2048	3e-4 → 3e-5	0.1	0.9	0.95	1.04	0.99

Specific Hyperparameters. Our proposed loss function is primarily characterized by two key hyperparameters, the temperature τ and the coefficient λ. We conduct extensive grid search experiments ($\tau \in [0.001, 0.01, 0.1]$ and $\lambda \in [0.2, 0.5, 1, 2, 5, 10, 20, 50, 100]$) on the 350M model to determine the effective range of these hyperparameters, and validate them on larger models according to scaling theory. The simple settings of $\tau = 0.01$ and $\lambda = 10$ are sufficient to achieve good performance for most experiments. To better adapt to the model scaling, we explore a more refined yet simple strategy to determine the selections, which is detailed in Sec.A.2.

Evaluations. To ensure a fair comparison, we conduct all evaluations on EleutherAI/lm-evaluation benchmark [Gao et al., 2024]. We mainly evaluate the performance of the pretrained model on downstream tasks of arc_easy, arc_challenge, openbookqa, boolq, hellaswag, piqa, winogrande,

mmlu, sciq (general reasoning ability) and the domain-specific downstream tasks of gsm8k, drop, race, squadv2, nq_open, humaneval, mbpp (three domains: math, code, and reading comprehension).

Training Resources. We conduct experiments on H800 GPUs. Pretraining the 350M model on 13B tokens requires approximately 56 GPU hours per experiment and the 7B model on 52B tokens takes over 2,000 GPU hours per experiment.

A.2 How to Scale Hyperparameters on Large Models?

In this part, we introduce a refined hyperparameter tuning mechanism to accommodate model scaling. Before introducing it, we first demonstrate the relationship between the representation ability of our SIMREG loss and the dimensionality of embeddings in the model. The SIMREG loss regularizes pretraining by leveraging the token embedding similarity between pairs of tokens. By assuming $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ are independent and identically distributed as isotropic random variables, e.g., $\mathbf{x}, \mathbf{y} \sim \mathcal{N}(0, I_d)$. Thus, we consider their cosine similarity $z = \frac{\langle \mathbf{x}, \mathbf{y} \rangle}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|} \in [-1, 1]$. Without loss of generality, we can assume $\frac{\mathbf{y}}{\|\mathbf{y}\|} = (1, 0, \cdots, 0)$ as the first basis of the spherical space S^{d-1}. Then the distribution of z can be transferred to the study of the first coordinate of $v \sim \text{Uinf}\left(S^{d-1}\right)$. Substitute v into the iterative form of spherical coordinates $v = (\cos\theta, \sin\theta \cdot \zeta)$ where $\zeta \in S^{d-2}$. According to the decomposition of the spherical surface unit, we have $d\sigma_{d-1}(v) = \sin^{d-2}(\theta)\, d\theta\, d\sigma_{d-1}(\zeta)$ and the marginal density of the polar angle:

$$f_p(\theta) = \frac{1}{|S^{d-1}|} \int_{S^{d-2}} \sin^{d-2}(\theta)\, d\sigma_{d-2}(v) = \frac{|S^{d-2}|}{|S^{d-1}|} \sin^{d-2}(\theta).$$

Then we consider the variable z. Due to the first coordinate $z = v_0 = \cos(\theta)$, we have:

$$f_p(z) = f_p(\theta) \left|\frac{d\theta}{dz}\right| = \frac{|S^{d-2}|}{|S^{d-1}|} \cdot \frac{\sin^{d-2}(\theta)}{\sin(\theta)} = \frac{|S^{d-2}|}{|S^{d-1}|} \left(1 - z^2\right)^{\frac{d-3}{2}} = \frac{\Gamma(\frac{d}{2})}{\sqrt{\pi}\Gamma(\frac{d-1}{2})} \left(1 - z^2\right)^{\frac{d-3}{2}}.$$

It is easy to check that $\mathbb{E}[z] = 0$ and $\mathbb{E}\left[z^2\right] = \frac{1}{d}$. Therefore, as the model size increases and the embedding dimensionality changes from d_0 to d_1, the capacity of SIMREG loss decreases by a factor of $\sqrt{\frac{d_1}{d_0}}$. To preserve the representation capability, we can revise the λ coefficient.

We next investigate the feasibility of this scaling method from an empirical perspective. We separately sweep the hyperparameters and report the evaluation perplexity (ppl) at the end of training.

Table 7: Validation perplexity (generalization performance) of different $(\tau, \lambda_{\text{reg}})$.

	$\tau = 0.005$	$\tau = 0.01$	$\tau = 0.02$	$\tau = 0.05$	$\tau = 0.1$
$\lambda_{\text{reg}} = 0$ (baseline)			15.06		
$\lambda_{\text{reg}} = 0.1$	14.98	15.01	14.92	14.95	14.98
$\lambda_{\text{reg}} = 0.2$	14.50	14.38	14.36	14.79	14.81
$\lambda_{\text{reg}} = 0.5$	14.45	14.41	14.43	14.54	14.88
$\lambda_{\text{reg}} = 1$	14.41	14.45	14.46	14.67	14.71
$\lambda_{\text{reg}} = 2$	14.39	14.41	14.49	14.65	14.74
$\lambda_{\text{reg}} = 5$	14.36	14.36	14.37	14.35	15.13
$\lambda_{\text{reg}} = 10$	14.34	**14.25**	14.36	14.44	15.65
$\lambda_{\text{reg}} = 20$	14.41	14.29	14.44	14.59	-
$\lambda_{\text{reg}} = 50$	14.74	14.33	14.55	14.78	-

The optimal range and variation trend in Table 7 and Figure 4 are almost identical to those observed in the optimization process, **indicating that the improvements brought by the SIMREG loss in both optimization and generalization are consistent**. The optimal choice of τ remains concentrated around 0.01. Next, we evaluate models of different scales (primarily with increased embedding hidden sizes).

Table 8: Optimal validation perplexity (generalization performance) of different model size.

	350M ($d = 1024$)	1.3B ($d = 2048$)	3B ($d = 3072$)	7B ($d = 4096$)
$\lambda_{\text{reg}} = 0$ (baseline)	15.06	10.72	9.70	8.99
$\lambda_{\text{reg}} = 5$	14.36	10.46	9.50	8.92
$\lambda_{\text{reg}} = 10$	**14.25**	**10.41**	9.46	8.84
$\lambda_{\text{reg}} = 20$	14.29	10.42	**9.44**	**8.78**
$\lambda_{\text{reg}} = 50$	14.33	10.49	9.49	8.81

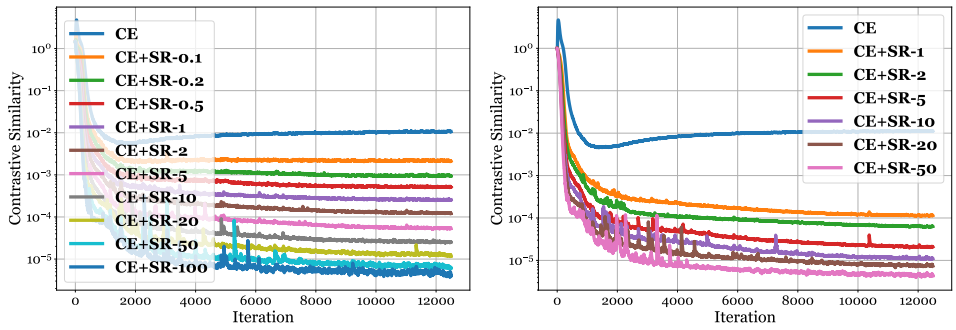
It can be observed that the trend largely aligns with our hypothesis. Therefore, we propose the following estimation method for the optimal hyperparameters:

$$\tau = 0.01,\ \lambda_{\text{reg}} \approx 10 \times \sqrt{\frac{d}{1024}},$$

where d is the dimension of the hidden-size of the token embedding. Of course, the scale of the model also affects the results. In practice, a simple grid search within this range of choices can be performed to identify the optimal combination.

A.3 SimReg Loss Curves

In this section, we mainly present the variations of the SIMREG loss. We explore the limitations of cross-entropy in LLM pretraining, namely, that it cannot achieve better classification performance simply by further increasing feature separability. This is because cross-entropy focuses solely on aligning predictions with ground-truth labels, while leaving the underlying structure of token embeddings insufficiently constrained. As the model scales up, this weakness becomes more pronounced: embeddings of the same class may still scatter in the representation space, leading to instability in optimization and slower convergence. By contrast, the SIMREG loss explicitly regularizes intra-class consistency and inter-class separation, complementing cross-entropy with a more direct control of embedding geometry. This additional constraint not only improves convergence speed but also yields more robust generalization in downstream tasks.



(a) LLaMA-350M. (b) LLaMA-1.3B.

Figure 6: The training curve of the SIMREG loss.

Figure 6 illustrates the loss behavior when increasing the coefficient of the SIMREG loss. It can be observed that even with a large weighting ratio, SIMREG does not cause the training to diverge. At the same time, we also note that the feature consistency loss exhibits a nearly monotonic trend. This phenomenon suggests that SIMREG serves as a stable regularization term: rather than interfering with the optimization of cross-entropy, it progressively strengthens the alignment of token embeddings as its weight grows. In practice, this means that a wide range of coefficient values can be applied without destabilizing training, making SIMREG highly robust and easy to integrate into large-scale pretraining pipelines.

A.4 Visualization of the Token Embedding Similarity

Here we provide more visualization demos of the true pretraining data samples from C4 dataset on LLaMA-7B.

Text1: [so I'm not sure if there's anything holding the back. I do not think there is by wiggling on it but could possibly have a strap or the like. I would think there must be a way to remove the panel blocking the bottom of the washer. We installed our own washer and used the clips mentioned in the previous post. Here is a PDF file on how they are used and what they look like. You may want to run your fingers over the entire carpeted lip ... typically, the]

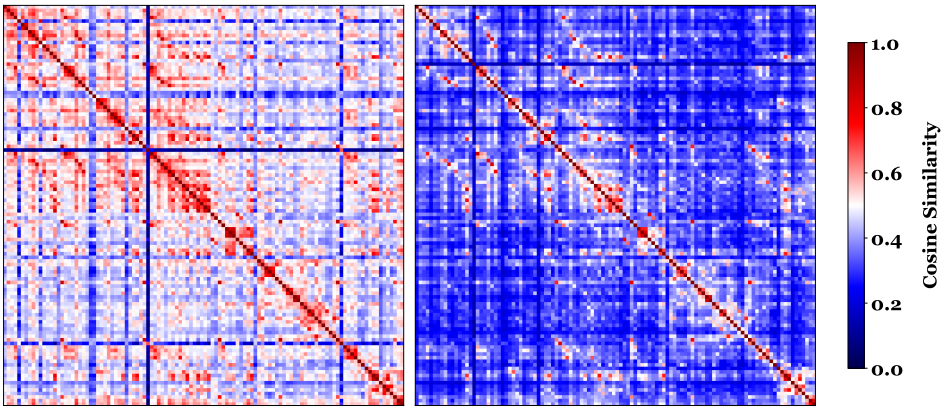


Figure 7: The averaged cosine similarity values are 0.488 (CrossEntropy only - left) and 0.354 (CrossEntropy + SIMREG - right).

Text2: [manufacturer runs screws into the floors/cabinets and the heads are buried in the carpet. There are two screws with square heads in the top of the carpet. Have you tired to do the recommended procedure to clean the lint out of the drain. 1. Run the unit without clothes and with the dry time off on cycle # 11. 2. When the water stops entering the unit push and hold the start button until all the lights come on then release the button.]

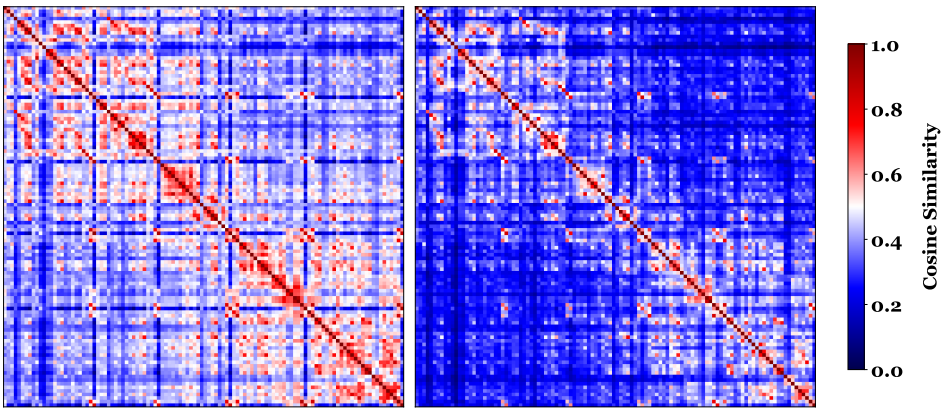


Figure 8: The averaged cosine similarity values are 0.445 (CrossEntropy only - left) and 0.333 (CrossEntropy + SIMREG - right).

B Appendix: Theoretical Analysis

In this section, we mainly demonstrate the theoretical understanding to show how the SIMREG loss improves the convergence and generalization efficiency. To this end, we first establish the fundamental properties of the proposed objective and analyze its impact on representation learning. We then present rigorous bounds and intuitive explanations that highlight its advantages over conventional cross-entropy training. These insights not only provide a deeper understanding of why SIMREG is effective but also offer useful guidance for its broader application in large-scale pretraining.

B.1 Relationship between Empirical loss and Margins

We first introduce the simplified modeling and corresponding notations of the LLM pretraining. Without loss of generality, we decompose the model into two simple parts. The first part is the front-end structure, which takes the raw data as input and outputs the embedding representations. The second part is the back-end structure, which transforms the embeddings into logits, followed by a cross-entropy loss function. We denote $X = [x_1, x_2, \cdots, x_n] \in \mathbb{R}^{n\times d}$ as the embeddings and $Z = f_h(X) \in \mathbb{R}^{n\times c}$ as the logits. The category label are denoted by $Y = [y_1, y_2, \cdots, y_n] \in \{1, 2, \cdots, C\}^n$. For the sample-wise cross entropy loss, we have:

$$\ell\left(x_i, y_i\right) = -z_{i,y_i} + \log\left(\sum_{j=1}^{K} e^{z_{i,j}}\right).$$

The empirical loss is $L = \frac{1}{n}\sum_{i=1}^{n} \ell(x_i, y_i)$. Then we consider the margin value in multi-class classification, which is also the joint gaps of different categories $m_i = z_{i,y_i} - \max_{j\neq y_i} z_{i,j}$. Therefore, we have:

$$\ell\left(x_i, y_i\right) = \log\left(1 + \sum_{j\neq y_i} e^{-\left(z_{i,y_i} - z_{i,j}\right)}\right) \leq \log\left(1 + (C-1)\, e^{-m_i}\right) \leq (C-1)\, e^{-m_i},$$

where the empirical loss is $L = \frac{1}{n}\sum_{i=1}^{n} \ell(x_i, y_i) \leq \frac{C-1}{n}\sum_{i=1}^{n} e^{-m_i}$. Generally, if the classification margins of all samples are increased by at least $\Delta \geq 0$, the loss will be multiplicatively reduced by a factor of $e^{-\Delta}$.

B.2 Equivalent Constraint of the SimReg Loss

Here we learn how the SIMREG loss affect the embeddings and the model performance. Here we let each embedding $\mathbf{e}_i = r_i\mathbf{a}_i$ where $r_i = \|\mathbf{e}_i\| \geq 0$ is the magnitude and $\mathbf{a}_i$ is the normalized embedding. SIMREG loss evaluates the exponential of the cosine similarity of two embeddings. Its core focus lies in the geometric information of the term a. To learn the performance of the SIMREG, for each label y_i, we define a positive set $\mathcal{P}_i = \{a_j : y_j = y_i\}$ and a negative set $\mathcal{N}_i = \{a_j : y_j \neq y_i\}$. The union of $\mathcal{P}_i$ and $\mathcal{N}_i$ always combines a complete sequence.

To understand the performance of SIMREG in detail, we first introduce a general kernel function $\kappa\left(\mathbf{u}, \mathbf{v}\right) = \exp\left(\mathbf{u}^\top\mathbf{v}\right)$, which admits the Maclaurin series $\kappa\left(\mathbf{u}, \mathbf{v}\right) = \sum_{m=0}^{\infty} \frac{\left(\mathbf{u}^\top\mathbf{v}\right)^m}{m!}$. It is a positive definite kernel on the unit sphere. By introducing an explicit map: $h : \mathbb{S}^{d-1} \to \mathcal{H}$ on the symmetric tensor powers:

$$h(\mathbf{u}) = \left[1, \frac{1}{\sqrt{\pi}}\mathbf{u}, \frac{1}{\sqrt{2!\pi^2}}\mathrm{vec}\left(\mathbf{u}^{\otimes 2}\right), \frac{1}{\sqrt{3!\pi^3}}\mathrm{vec}\left(\mathbf{u}^{\otimes 3}\right), \cdots\right], \tag{7}$$

thus we have the transformation of $\langle h(\mathbf{u}), h(\mathbf{v})\rangle = \kappa\left(\mathbf{u}, \mathbf{v}\right)$. The mapping h is to construct a linear expansion of κ in the reproducing kernel Hilbert space (RKHS) $\mathcal{H}$. Therefore, we have:

$$\log\left(\sum_{i\in\mathcal{P}_k} \exp\left(\mathbf{e}_k^\top\mathbf{e}_i\right)\right) = \log\left(\sum_{i\in\mathcal{P}_k} \langle h(\mathbf{e}_k), h(\mathbf{e}_i)\rangle\right) = \log\left(\left\langle h(\mathbf{e}_k), \mu_k^+\right\rangle\right) + \log\left(|\mathcal{P}_k|\right),$$

where $\mu_k^+ = \frac{1}{|\mathcal{P}_k|}\sum_{i\in\mathcal{P}_k} h(\mathbf{e}_i)$ is the positive kernel means. Here $|\mathcal{P}_k|$ can be considered as a offset to scale the positive samples. The theoretical analysis can be symmetrically extended to negative samples, yielding an equivalent conclusion.

Therefore, the SIMREG loss consider the difference between the positive and negative set by:

$$\min_{\mathbf{e}=f_E(\mathbf{x})} J = \mathbb{E}_{\mathbf{x}} \log \left(\frac{\left\langle h(\mathbf{e}_k), \mu_k^- \right\rangle}{\left\langle h(\mathbf{e}_k), \mu_k^+ \right\rangle} \right) + \log \left(\frac{|\mathcal{N}_k|}{|\mathcal{P}_k|} \right).$$

The ratio of positive to negative samples only affects the scale of the loss, but does not alter the primary optimization objective of the first term. It pushes the anchor direction to align with the positive kernel mean and to anti-align with the negative kernel mean. It also nudges the group means themselves: positives move toward anchors that they are already close to, and negatives move away in the RKHS sense. We also have the nearest positive prototype for each class:

$$\max_{\|\mathbf{e}\|} \left\langle h(\mathbf{e}), \mu_k^+ \right\rangle = \|h(\mathbf{e})\| \|\mu_k^+\| = \kappa(\mathbf{e}, \mathbf{e}) \|\mu_k^+\| = \sqrt{e} \|\mu_k^+\|.$$

The same, the $\sqrt{e}$ scaling also hold for the negative set. Beyond the optimization objective itself, we can further consider the problem from the perspective of gradient directions to refine the learning target. By considering the Fréchet gradient, we have:

$$\nabla_{h(\mathbf{e}_k)} J = \frac{\mu_k^-}{\left\langle h(\mathbf{e}_k), \mu_k^- \right\rangle} - \frac{\mu_k^+}{\left\langle h(\mathbf{e}_k), \mu_k^+ \right\rangle}.$$

Generally, $\mu_k^- \neq \mu_k^+$. From the gradient expression, we can see that the optimization dynamics naturally combine both "attractive" and "repulsive" effects. Specifically, the first term pushes the representation $h(\mathbf{e}_k)$ away from the negative center μ_k^-, while the second term pulls it closer to the positive center μ_k^+. As a result, the overall update direction is shaped by the joint effect of being attracted to positives and repelled from negatives, thereby optimizing the representation space effectively. From the above two perspectives, it is clear that SIMREG enforces feature consistency alignment in the RKHS sense.

B.3 Center-aligned Embeddings Can Enhance Optimization

Then we consider the performance of the center-aligned embedding. To learn the transferred impact from the mapping $h(\mathbf{e}_k)$ to vanilla variable $\mathbf{e}_k$, we first consider the normalized $\mathbf{a}_k$ term, where the cosine similarity can be considered as $\mathbf{a}_k^\top \mathbf{a}_j$. To simplify the notation, we additionally define the weighted average direction of a variable $\mathbf{a}$ over its associated positive and negative sets by $\mathbf{v}_k^+ = \frac{1}{\|\mathcal{P}_k\|} \sum_{i \in \mathcal{P}_k} \exp\left(\mathbf{a}_k^\top \mathbf{a}_i\right) \mathbf{a}_i$ and $\mathbf{v}_k^- = \frac{1}{\|\mathcal{N}_k\|} \sum_{j \in \mathcal{N}_k} \exp\left(\mathbf{a}_k^\top \mathbf{a}_j\right) \mathbf{a}_j$. Similarly, we also define the loss of positive set and negative set as P_k and N_k. Therefore, we have the following gradient form:

$$\nabla_{\mathbf{a}_k} L_{\text{sr}} = \frac{N_k}{P_k + N_k} \left(\mathbf{v}_k^- - \mathbf{v}_k^+ \right).$$

Since the $\mathbf{a}_k$ is constrainted by $\|\mathbf{a}_k\| = 1$, the true update direction is obtained by projecting the gradient onto the tangent space: $-\prod_{\mathbf{v}_k} \nabla_{\mathbf{a}_k} L_{\text{sr}} = -\frac{N_k}{P_k + N_k} \left(I - \mathbf{a}_k \mathbf{a}_k^\top\right) \left(\mathbf{v}_k^- - \mathbf{v}_k^+\right)$. Next, we analyze how the gradient dynamics associated with the positive sample set vary along the update direction. This dynamic essentially characterizes how strongly the representation is pulled toward the positive center during optimization. A larger value indicates that the update direction aligns well with the attraction force from positive samples, thereby accelerating convergence. Conversely, a smaller value reflects weaker alignment, suggesting limited contribution from positive samples in shaping the optimization trajectory. For the positive sample loss, we obtain (for clarity of exposition, we omit constant scalar terms):

$$\frac{d}{dt} \|\mathbf{a}_k - \mathbf{v}_k^+\|^2 = 2 \left(\mathbf{a}_k - \mathbf{v}_k^+\right)^\top \left(I - \mathbf{a}_k \mathbf{a}_k^\top\right) \mathbf{v}_k^+ \underbrace{- 2 \left(\mathbf{a}_k - \mathbf{v}_k^+\right)^\top \left(I - \mathbf{a}_k \mathbf{a}_k^\top\right) \mathbf{v}_k^-}_{\text{negative perturbation}}.$$

When treating the update on the negative sample set as a small perturbation to that on the positive samples, we have $\frac{d}{dt} \|\mathbf{a}_k - \mathbf{v}_k^+\|^2 \leq 2 \left(\mathbf{a}_k^\top \mathbf{v}_k^+\right)^2 - 2\|\mathbf{v}_k^+\|^2 \leq 0$. Similarly, the gradient dynamics on the negative sample set can be obtained as $\frac{d}{dt} \|\mathbf{a}_k - \mathbf{v}_k^-\|^2 \geq 0$. In conclusion, taking a small step along the tangent update direction inherently drives the representation closer to the weighted center of the positive class while simultaneously pushing it away from that of the negative class. In

other words, such updates reinforce the consistency among positive samples and reduce the influence of negatives, thereby shaping a clearer separation in the feature space. Importantly, this property does not rely on any assumptions about the underlying functional form, but rather arises directly from the optimization objective itself, ensuring both generality and robustness. To further refine the update dynamics, a temperature coefficient can be introduced as a scaling factor. By adjusting the sharpness of the similarity distribution, the temperature effectively controls the relative strength of attraction toward positive samples and repulsion from negative samples. In particular, incorporating a temperature into the formulation normalizes the gradient magnitudes and ensures that the update direction satisfies the desired balance condition between positive and negative contributions. This modification not only stabilizes training but also enhances the flexibility of the loss function in adapting to different representation scales. This result can be directly extended from the normalized variables to the original embedding variables $\mathbf{e}$, thereby completing the proofs.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: This discussion is stated after the Conclusion section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide complete proofs for all theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide detailed computational instructions and hyperparameter tables to support reproducibility.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We have not publicly released the code at this stage. We provide very detailed reproducibility instructions in the paper, including hyperparameter choices and all other key experimental details.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide all the necessary details to understand the experimental results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The experiments in this paper involve pre-training, which is computationally very expensive. Repeating experiments over multiple random seeds would require substantial computational resources. Therefore, we do not provide error bars or other statistical analyses in this paper. Nevertheless, our experimental presentation is consistent with the standard practice in most works of this type.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The appendix provides the computational resources for all experiments, including the number of GPUs and the running time.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conducted in this paper fully conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: This discussion is stated after the Conclusion section.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We properly cite and acknowledge the creators and original owners of all assets used in this paper, including code, datasets, and models where applicable. We also respect the corresponding licenses and terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.