MIPIC: Matryoshka Representation Learning via Self-Distilled Intra-Relational and Progressive Information Chaining

Anonymous ACL submission

Abstract

Representation learning is fundamental to NLP, but building embeddings that work well at different computational budgets is challenging. Matryoshka Representation Learning (MRL) offers a flexible inference paradigm through nested embeddings; however, learning such structures requires explicit coordination of how information is arranged across embedding dimensionality and model depth. In this work, we propose MIPIC (Matryoshka Representation Learning via Self-Distilled Intra-Relational Alignment and Progressive Information Chaining), a unified training framework designed to produce structurally coherent and semantically compact Matryoshka representations. MIPIC promotes cross-dimensional structural consistency through Self-Distilled Intra-Relational Alignment (SIA), which aligns token-level geometric and attention-driven relations between full and truncated representations using top-k CKA self-distillation. Complementarily, it enables depth-wise semantic consolidation via Progressive Information Chaining (PIC), a scaffolded alignment strategy that incrementally transfers mature task semantics from deeper layers into earlier layers. Extensive experiments on STS, NLI, and classification benchmarks (spanning models from TinyBERT to BGEM3, Qwen3) demonstrate that MIPIC yields Matryoshka representations that are highly competitive across all capacities, with significant performance advantages observed under extreme low-dimensional.

1 Introduction

Learned dense representations are the cornerstone of modern NLP (Mikolov et al., 2013; Pennington et al., 2014; Devlin et al., 2019), yet their high computational demands often limit deployment (Wang et al., 2020). Matryoshka Representation Learning (MRL) (Kusupati et al., 2022) addresses this by introducing nested embeddings that enable adaptive inference-time truncation, allowing a single model to satisfy diverse computational budgets without retraining. Training effective Matryoshka representations, however, presents challenges that extend beyond conventional embedding learning. Unlike standard representations, MRL requires semantic information to be explicitly organized such that meaningful structure is preserved under progressive dimensional truncation. Existing MRL formulations (Kusupati et al., 2022; LI et al., 2025) primarily enforce nested usability through independent supervision at different truncation levels or sentence-level alignment objectives. While such strategies encourage prefix usability, they leave open the broader question of how semantic information should be internally arranged to support robust Matryoshka structures. In particular, learning low-dimensional prefix embeddings that remain semantically expressive requires the model to carefully organize semantic features across both embedding dimensions and network layers, involving a non-trivial restructuring of how information is represented and propagated.

First, effective MRL can be viewed through the lens of **cross-dimensional structural alignment**, where low-dimensional prefixes are encouraged to reflect not only sentence-level outputs but also the internal relational structure (like token-level geometric relationships, hidden states) of the full-dimensional representations. From this perspective, aligning relational patterns across different dimension spaces provides a natural way to support stable semantic behavior in compact prefixes. Second, MRL may also be understood from the perspective of involving **depth-wise semantic consolidation**, in which task-relevant information is gradually condensed from deeper layers, where representations are typically more expressive, into earlier layers. This process is inherently progressive: rather than emerging at a single truncation point, core semantic features can be refined and transferred as information flows through the net-

work. Viewing Matryoshka training through this lens highlights the benefits of intermediate guidance that supports a smooth, coarse-to-fine refinement of representations across depth.

Motivated by these considerations, we propose **MIPIC** (**M**atryoshka Representation Learning via Self-Distilled **I**ntra-Relational Alignment and **P**rogressive **I**nformation **C**haining), a unified training framework that explicitly organizes information across both embedding dimensionality and model depth, enabling the learning of structurally coherent and semantically compact Matryoshka representations. To encourage cross-dimensional structural consistency, MIPIC employs Self-Distilled Intra-Relational Alignment (SIA). Rather than aligning representations solely at the sentence level like prior MRL approaches (Kusupati et al., 2022; LI et al., 2025), SIA focuses on token-level intra-relations derived from geometric similarity and attention-based importance. Using Centered Kernel Alignment (CKA) (Kornblith et al., 2019) with a hard top-k selection strategy, SIA selectively aligns the most salient relational patterns between full and truncated representations. This targeted self-distillation goes beyond simple alignment; it effectively reorganizes the internal feature hierarchy, forcing critical semantic structures to be prioritized and concentrated within the low-dimensional prefix dimensions. Complementing cross dimensional alignment, MIPIC introduces Progressive Information Chaining (PIC) to guide the flow of semantic information across the network depth. Unlike prior methods that restrict task supervision solely to the final representation, which can be unstable when compressing rich knowledge into narrow dimensions, PIC builds a scaffolded alignment pipeline in which each step acts as a semantic bridge. This design is motivated by empirical findings that lower Transformer layers predominantly encode low level linguistic features such as syntax and basic semantics, while higher layers increasingly capture task specific and discriminative signals (Jawahar et al., 2019; Liu et al., 2019). By propagating task aware cues from upper layers into earlier ones, PIC performs a controlled early intervention that gently biases the formation of low level representations toward task relevant subspaces, ensuring that useful discriminative structure is available from the beginning of the network processing. Importantly, by aligning only the most essential components of each layer representation, which have been reorganized by SIA to concentrate the most critical task relevant knowledge into a compact subspace, PIC allows the remaining dimensions to continue modeling low level and general linguistic features. This selective supervision preserves the model natural representational flexibility, avoids over regularization, and enables a stable and progressive refinement of task relevant information across depth. Together, SIA and PIC provide a principled training framework for MRL that explicitly organizes semantic information across both dimensionality and depth.

Our main contributions are:

- We propose **MIPIC**, a unified framework that addresses structural and semantic incoherence in MRL by explicitly coordinating cross-dimensional alignment and depth-wise information consolidation.

- We introduce two complementary mechanisms: Self-Distilled Intra-Relational Alignment (SIA), which utilizes top-k CKA self distillation to enforce cross-dimensional topological consistency via self-distillation, and Progressive Information Chaining (PIC), a scaffolded learning strategy that enables lower-dimensional representations to acquire task-relevant information early in training through progressive semantic guidance across network depth.

- We present extensive experiments on STS, NLI, and text classification benchmarks showing that MIPIC significantly improves representation efficiency across diverse backbones, from TinyBERT-6L and BERT-base to large-scale models such as Qwen3 embedding 0.6B and BGE-M3. MIPIC remains competitive at full capacity while consistently outperforming state-of-the-art baselines under severe truncation, particularly in low-dimensional regimes.

2 Related Work and Background

2.1 Related Works

Learning Sentence Embeddings Sentence embeddings have evolved from static word vectors (Pennington et al., 2014; Mikolov et al., 2013) and contextual models (Peters et al., 2018; Devlin et al., 2019), which initially lacked optimized sentence-level semantics. Sentence-BERT (Reimers and

Gurevych, 2019) addressed this via siamese fine-tuning, dramatically improving retrieval efficiency. Subsequent work introduced contrastive learning (Gao et al., 2022; Nishikawa et al., 2022) to enhance robustness, while recent advances leverage Large Language Models to further refine representation quality (He et al., 2025; BehnamGhader et al., 2024; Tao et al., 2025; Li and Zhou, 2025).

Matryoshka Representation Learning methods While sentence embeddings have advanced significantly, their fixed dimensionality remains a computational bottleneck at scale. Matryoshka Representation Learning (MRL) addresses this by learning nested coarse-to-fine prefixes within a single embedding, enabling low-dimensional prefixes to function as standalone representations with only $O(\log d)$ supervision points and no additional forward passes (Kusupati et al., 2022). This design enables substantial compression with minimal accuracy loss and faster inference. Building on MRL, Espresso Sentence Embeddings (ESE) further enhance scalability across depth and dimensionality through learn to express and learn to compress mechanisms (LI et al., 2025). Beyond sentence embeddings, the Matryoshka principle has been applied to image generation (Gu et al., 2024), multimodal representation learning (Cai et al., 2024), and multimodal LLMs (Hu et al., 2024).

2.2 Background

2.2.1 Matryoshka Representation Learning

Let an encoder f_θ map an input sentence x to a high-dimensional embedding $z = f_\theta(x) \in \mathbb{R}^D$. In Matryoshka Representation Learning (Kusupati et al., 2022) we fix an ordered set of nested prefix dimensions $\mathcal{D} = \{d_1, d_2, \ldots, d_n\}, 0 < d_1 < d_2 < \cdots < d_n = D$, so that for each $d_i \in \mathcal{D}$ the truncated embedding is the prefix $z^{(d_i)} := z_{1:d_i} \in \mathbb{R}^{d_i}$. Each prefix $z^{(d_i)}$ is treated as an independent representation used by a (possibly shared) task head $g_{\phi_i} : \mathbb{R}^{d_i} \to \mathcal{Y}$ and evaluated with a task loss $\mathcal{L}_{\text{task}}^{(d_i)}(x, y) = \ell\big(g_{\phi_i}(z^{(d_i)}), y\big)$ for a target y. The Matryoshka training objective jointly supervises all prefixes; a common choice is the unweighted sum

$$\mathcal{L}_{\text{MRL}}(\theta, \{\phi_i\}) = \sum_{i=1}^{n} \mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{train}}}\big[\mathcal{L}_{\text{task}}^{(d_i)}(x, y)\big], \tag{1}$$

or, when desired, a weighted variant $\sum_{i=1}^{n} \alpha_i \, \mathbb{E}[\mathcal{L}_{\text{task}}^{(d_i)}]$ with $\alpha_i \geq 0$. By optimizing this objective, Matryoshka models learn a single high-dimensional embedding whose low-dimensional prefixes $z^{(d_1)}, \ldots, z^{(d_{n-1})}$ are immediately usable as standalone embeddings at inference time, avoiding extra forward passes.

2.2.2 Measuring Representational Similarity: Centered Kernel Alignment (CKA)

Canonical Correlation Analysis (CCA) (Hardoon et al., 2004) has been widely used to compare learned representations by seeking linear projections that maximize correlation between two feature sets. However, CCA can be fragile in practice: it is sensitive to simple transformations of the features and can be costly to compute for very high-dimensional activations common in deep models (Kornblith et al., 2019). Centered Kernel Alignment (CKA) (Kornblith et al., 2019) provides a more robust and efficient alternative by shifting the comparison from individual feature coordinates to the *pairwise similarity structure* induced by each representation. Given two representation matrices $\mathbf{X} \in \mathbb{R}^{m\times S}$ and $\mathbf{Y} \in \mathbb{R}^{m\times T}$ for the same m inputs, CKA builds kernel (Gram) matrices $\mathbf{K}$ and $\mathbf{L}$ with entries $K_{ij} = k(x_i, x_j)$ and $L_{ij} = l(y_i, y_j)$, and measures their dependence via the Hilbert–Schmidt Independence Criterion (HSIC) after centering:

$$\text{HSIC}(\mathbf{K}, \mathbf{L}) = \frac{1}{(m-1)^2} \operatorname{tr}(\mathbf{KHLH}), \tag{2}$$

where $\mathbf{H} = I_m - \frac{1}{m}\mathbf{1}\mathbf{1}^\top$ removes mean effects. CKA then normalizes HSIC to obtain:

$$\text{CKA}(\mathbf{X}, \mathbf{Y}) = \frac{\text{HSIC}(\mathbf{K}, \mathbf{L})}{\sqrt{\text{HSIC}(\mathbf{K}, \mathbf{K})\,\text{HSIC}(\mathbf{L}, \mathbf{L})}}. \tag{3}$$

3 Methodology

We introduce **MIPIC**, a framework designed to synchronize the learning of nested representations with the model's internal information flow. Our approach operates through two synergistic mechanisms: **Self-Distilled Intra-Relational Alignment (SIA)** reorganizes the internal representation hierarchy to prioritize core structural features within low-dimensional subspaces, and **Progressive Information Chaining (PIC)** establishes a scaffolded pipeline to condense task-specific information into these prioritized dimensions across the network's depth. Together, these components ensure that the resulting embeddings are both structurally consistent and semantically rich across all granularities. The overall architecture is shown in Figure 1.

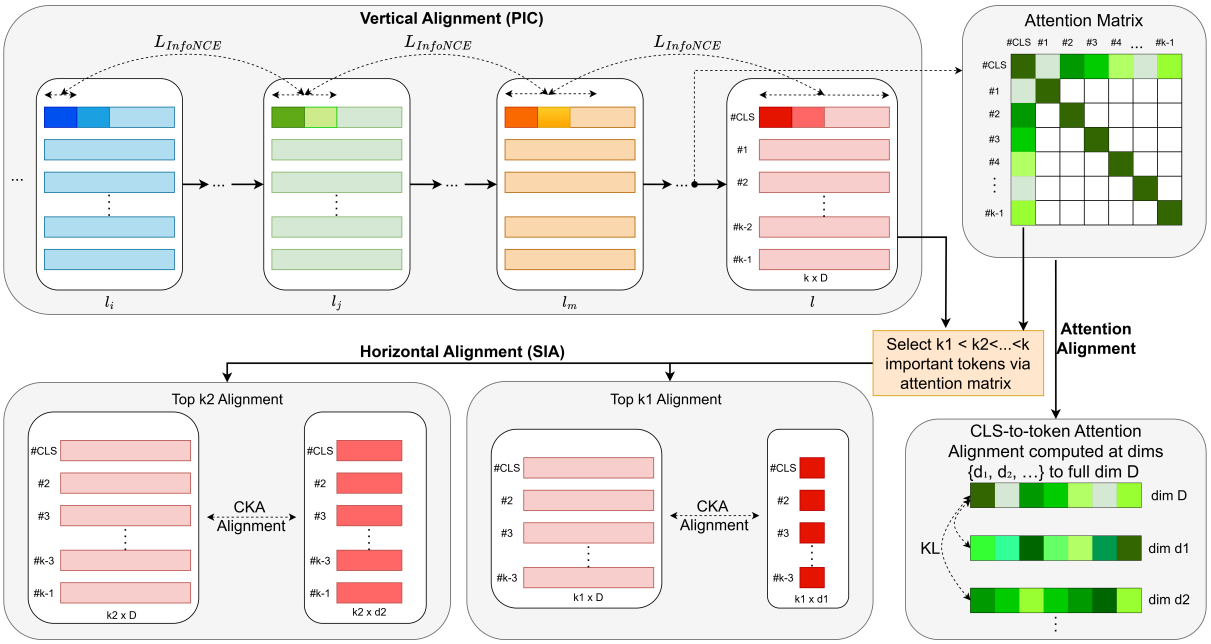

Figure 1: **Overview of MIPIC:** The framework operates via two mechanisms:**Vertical Alignment (PIC)** and **Horizontal Alignment (SIA)**

3.1 Preliminaries and Notation

Consider a Transformer-based encoder with L layers. Let the input sequence length be m, and the model hidden dimension be D. We define a strictly increasing sequence of nested feature sizes $\mathcal{D} = \{d_1 < d_2 < \cdots < d_n = D\}$ and a corresponding sequence of layer $\mathcal{L} = \{l_1 < l_2 < \cdots < l_n\}$ at which both SIA and PIC mechanisms are applied.

3.2 SIA: Self-Distilled Intra-Relational Alignment

Standard Matryoshka methods enforce nestedness by minimizing distances between sentence-level vectors, often ignoring token-level information. We propose SIA, which frames low-dimensional prefix alignment as a **reorganization of internal information** rather than mere distance matching. Using layer-wise self-distillation, each layer's full-dimensional representation teaches its lower-dimensional prefix, transferring only essential structural signals. SIA ensures truncated subspaces preserve the intra-relational structure of the full representation, emphasizing core information. This is achieved via Attention Distribution Matching and Top-k Hidden State Alignment with CKA.

3.2.1 Attention Distribution Matching

The core intuition of this module is to preserve the **relative importance ordering** of tokens within a sequence. We adopt a self-distillation strategy where the full-dimensional representation acts as a teacher to guide the lower-dimensional representation. Let $h_{\text{CLS}} \in \mathbb{R}^D$ and $h_j \in \mathbb{R}^D$ denote the hidden states of the [CLS] token and the j-th contextual token in the full dimension D. The teacher's attention scores $s_j^{(D)}$ and the resulting distribution a_D serve as the ground truth for alignment:

$$s_j^{(D)} = \frac{h_{\text{CLS}} \cdot h_j}{\sqrt{D}}, \quad a_D = \text{softmax}\left(\frac{s^{(D)}}{\tau}\right) \tag{4}$$

For dim d_i, sliced tokens $h_j[1:d_i]$ are up-projected via a learnable matrix $P_i \in \mathbb{R}^{d_i \times D}$. We use h_{CLS} as the query anchor since it encodes global sequence semantics, deriving attention scores $s_j^{(i)}$ that quantify each token's contribution:

$$s_j^{(i)} = \frac{h_{\text{CLS}} \cdot P_i^\top h_j^{(i)}}{\sqrt{D}}, \quad a_i = \text{softmax}\left(\frac{s^{(i)}}{\tau}\right) \tag{5}$$

The matrix P_i is trained jointly with the backbone to minimize the Kullback-Leibler (KL) divergence between the student and teacher distributions:

$$\mathcal{L}_{\text{att}}^{(i)} = \sum_{d_i \in \mathcal{D}} \text{KL}(a_i \| a_D) \tag{6}$$

We use the full-dimensional h_{CLS} as a fixed semantic reference during training. The similarity between each token and this [CLS] vector serves as an

estimate of the token importance. By aligning these similarities across dimensions, we enforce a **ranking consistency constraint**: the lower-dimensional representation is encouraged to preserve the same ordering of important tokens learned by the full-dimensional model, so that the most informative tokens remain emphasized even after compression.

3.2.2 Top-k Hidden State Alignment via CKA

Low-dimensional representations possess limited capacity, making it impractical to encode exhaustive token-level details without introducing noise. We argue that forcing a small bottleneck to align with the entire sequence leads to information saturation. Instead, we use a **Top-k Selection** with Centered Kernel Alignment (CKA) strategy to reorganize information, focusing exclusively on a compact subset of the most informative tokens.

Hard Top-k Selection For each truncated dimension d_i, we select the top k_i tokens based on their importance scores in a_D, where k_i increases monotonically with d_i. This selective strategy offers dual benefits: it significantly reduces the computational overhead of the alignment process and acts as a denoising filter, preventing low-capacity prefixes from being saturated by irrelevant or redundant context. This also creates a dual Matryoshka structure: low-dimensional cores capture a coarse semantic skeleton, while higher-dimensional spaces progressively incorporate finer details. Crucially, the selected token sets are nested: $\mathcal{S}_{k_1} \subset \mathcal{S}_{k_2} \subset \cdots \subset \mathcal{S}_{k_n} \subseteq \{1, \ldots, m\}$, where $\mathcal{S}_{k_i}$ denotes the indices of the k_i most important tokens. This ensures that higher-capacity embeddings refine and enrich the semantic core learned by lower-capacity ones, rather than relearning it.

Representation Consistency via Linear CKA. Given these nested token subsets, we require an alignment objective that enforces structural consistency between the resulting student and teacher representations, even though they differ in feature dimensionality (d_i versus D). To this end, we adopt Centered Kernel Alignment (CKA) with a linear kernel, which measures similarity between representation spaces in a dimension-agnostic manner. We specifically choose the linear formulation for its computational efficiency, which is critical when scaling to LLMs. Let us define two kernel matrices:

$$\mathbf{p}_i = \mathbf{h}_i \mathbf{h}_i^\top, \quad \mathbf{P}_i = \mathbf{H}_i \mathbf{H}_i^\top \tag{7}$$

We extract the submatrices for the student $\mathbf{h}_i \in \mathbb{R}^{k_i \times d_i}$ and the teacher $\mathbf{H}_i \in \mathbb{R}^{k_i \times D}$ using the nested indices defined previously. First, we compute the centered feature matrices $\tilde{\mathbf{h}}_i$ and $\tilde{\mathbf{H}}_i$ as:

$$\tilde{\mathbf{h}}_i = \mathbf{h}_i(\mathbf{I}_{k_i} - \frac{1}{k_i}\mathbf{1}\mathbf{1}^\top), \quad \tilde{\mathbf{H}}_i = \mathbf{H}_i(\mathbf{I}_{k_i} - \frac{1}{k_i}\mathbf{1}\mathbf{1}^\top). \tag{8}$$

where $\mathbf{I}_{k_i}$ is the identity matrix and $\mathbf{1}$ is a vector of ones. For linear kernels, the HSIC can be efficiently computed as (Kornblith et al., 2019):

$$\text{HSIC}(\mathbf{p}_i, \mathbf{P}_i) = \left\| \text{cov}(\tilde{\mathbf{h}}_i^\top, \tilde{\mathbf{H}}_i^\top) \right\|_F^2, \tag{9}$$

where $\text{cov}(\mathbf{X}, \mathbf{Y}) = \mathbf{X}^\top \mathbf{Y}$ denotes the covariance function. Moreover, from section 2.2.2, the Linear CKA metric between $\mathbf{h}_i$ and $\mathbf{H}_i$ is defined as:

$$\text{CKA}\left(\mathbf{h}_i, \mathbf{H}_i\right) = \frac{HSIC(\mathbf{p}_i, \mathbf{P}_i)}{\sqrt{HSIC(\mathbf{p}_i, \mathbf{p}_i) \cdot HSIC(\mathbf{P}_i, \mathbf{P}_i)}} \tag{10}$$

Combining two equations Eq.9 and Eq.10, we obtain the formulation of linear CKA between two hidden state matrices:

$$\text{CKA}\left(\mathbf{h}_i, \mathbf{H}_i\right) = \frac{\left\| \text{cov}(\tilde{\mathbf{h}}_i^\top, \tilde{\mathbf{H}}_i^\top) \right\|_F^2}{\left\| \text{cov}(\tilde{\mathbf{h}}_i^\top, \tilde{\mathbf{h}}_i^\top) \right\|_F \cdot \left\| \text{cov}(\tilde{\mathbf{H}}_i^\top, \tilde{\mathbf{H}}_i^\top) \right\|_F} \tag{11}$$

Finally, at each d_i in the Matryoshka dimensions, we define the CKA loss between the lower-dimension full-dimensional hidden state:

$$\mathcal{L}_{\text{CKA}}^{(i)} = 1 - \text{CKA}\left(\mathbf{h}_i, \mathbf{H}_i\right) \tag{12}$$

We minimize this loss to ensure that the lower-dimensional bottleneck learns a geometric structure to that of the full-dimensional representation space. By maximizing CKA on these nested subsets, we ensure that the geometric relationships and structural organization within the semantic core are preserved robustly across all granularities. We specifically adopt CKA because it provides a robust measure of representational similarity that is invariant to orthogonal transformations and isotropic scaling (Kornblith et al., 2019). This invariance properties of CKA is particularly critical in our framework, as we must simultaneously align multiple nested dimensions of varying capacities.

3.2.3 Final SIA loss

We define the composite SIA loss at layer k by integrating both attention-based importance and

CKA loss consistency across the entire Matryoshka dimensions $\mathcal{D} = \{d_1 < d_2 < \cdots < d_n = D\}$:

$$\mathcal{L}_{\text{SIA}}^{(k)} = \sum_{i=1}^{n} (\mathcal{L}_{\text{att}}^{(i)} + \mathcal{L}_{\text{CKA}}^{(i)}) \tag{13}$$

The total SIA loss is then computed by summing these local alignment constraints over the network's depth, specifically across the target layers $\mathcal{L}$:

$$\mathcal{L}_{\text{SIA}} = \sum_{k \in \mathcal{L}} \mathcal{L}_{\text{SIA}}^{(k)} \tag{14}$$

3.3 PIC: Progressive Information Chaining

Building upon the priority-based structure induced by **SIA**, which reorganizes internal representations to emphasize prefix dimensions, we introduce **PIC** to guide the flow of semantic information across network depth. While many existing MRL approaches apply task supervision primarily at the final representation, such a design places the burden of semantic compression at a single endpoint. **PIC** instead adopts a **scaffolded alignment pipeline** that provides intermediate guidance throughout the network. Motivated by the progressive specialization of Transformer layers, PIC propagates task-relevant cues from deeper layers to earlier ones, enabling lower-dimensional representations to acquire essential task semantics at early stages of learning. Supervision is restricted to the compact subspace reorganized by SIA, concentrating alignment on the most informative dimensions while preserving low-level features and representational flexibility in the remaining space.

Formal Objective. Let the scaffolding structure be defined by a sequence of n checkpoints $\mathcal{C} = \{(l_i, d_i)\}_{i=1}^{n}$, ordered by depth and capacity such that $l_1 < l_2 < \cdots < l_n$ and $d_1 < d_2 < \cdots < d_n$. Here, l_i denotes the layer index and d_i the embedding dimension. We denote the truncated `[CLS]` representation at checkpoint i as $\mathbf{z}_i \in \mathbb{R}^{d_i}$.

Our aim is to promote a progressive condensation of task-relevant information, such that each checkpoint learns to supply the essential signals required by the next, allowing early, low-capacity representations to serve as reliable building blocks for deeper ones. Intuitively, this can be framed as encouraging high mutual information (MI) between adjacent checkpoints: maximizing $I(\mathbf{z}_i; \mathbf{z}_{i+1})$ requires that the earlier representation $\mathbf{z}_i$ retain the core features needed to predict the next representation $\mathbf{z}_{i+1}$. In the context of Matryoshka learning this enforces a progressive condensation of task-relevant signals rather than a single long semantic jump from early checkpoints to the final output. Since calculating exact MI is computationally impossible, we use **InfoNCE** as an approximation:

$$I(\mathbf{z}_i; \mathbf{z}_{i+1}) \geq \log N - \mathcal{L}_{\text{InfoNCE}} \tag{15}$$

where N is the batch size. To bridge the dimensionality mismatch ($d_i \neq d_{i+1}$), we employ a lightweight nonlinear projector $\phi_i : \mathbb{R}^{d_i} \to \mathbb{R}^{d_{i+1}}$ to map the lower-dimensional upstream representation into the semantic space of the downstream target. The local alignment loss for each step is:

$$\mathcal{L}_{\text{chain}}^{(i)} = -\mathbb{E}\left[\log \frac{\exp(\text{sim}(\phi_i(\mathbf{z}_i), \mathbf{z}_{i+1})/\tau)}{\sum_{\mathbf{z}' \in \mathcal{B}_{i+1}} \exp(\text{sim}(\phi_i(\mathbf{z}_i), \mathbf{z}')/\tau)}\right] \tag{16}$$

where $\mathcal{B}_{i+1}$ contains the positive pair $\mathbf{z}_{i+1}$ and $N-1$ negative samples from the batch, and τ is the temperature scaling parameter. The total Progressive Information Chaining loss accumulates these local alignment signals:

$$\mathcal{L}_{\text{PIC}} = \sum_{i=1}^{n-1} \mathcal{L}_{\text{chain}}^{(i)} \tag{17}$$

This chain-based alignment fundamentally differs from direct supervision by focusing on **essential information transfer**. By aligning only the most salient parts of adjacent layers via Mutual Information, PIC excessive information loss while avoiding the rigidity of element-wise matching. It ensures that the early layers act as a flexible foundation, organizing task-relevant features in a way that supports, rather than strictly replicates, the subsequent refinement stages. This maintains the natural diversity of features across depth, leading to a more robust final representation.

3.4 Overall Training Objective

The full objective of MIPIC integrates our proposed self-distillation mechanisms with the standard Matryoshka formulation. The total loss is:

$$\mathcal{L}_{\text{MIPIC}} = \alpha \mathcal{L}_{\text{MRL}} + (1 - \alpha)[\mathcal{L}_{\text{SIA}} + \mathcal{L}_{\text{PIC}}]\,. \tag{18}$$

Here, $\mathcal{L}_{\text{MRL}}$ denotes the Matryoshka task-specific loss (like in Equation 1). One hyperparameter is used: $\alpha \in [0, 1]$ trades off the MRL and MIPIC.

Datasets	Dim.	TinyBERT 6L				BERT				Datasets	Dim.	TinyBERT 6L				BERT			
		Unsup SimCSE	MRL	ESE	MIPIC	Unsup SimCSE	MRL	ESE	MIPIC			Unsup SimCSE	MRL	ESE	MIPIC	Unsup SimCSE	MRL	ESE	MIPIC
	16	30.17	40.64	43.30	**48.93**	35.92	46.39	45.86	**54.65**		16	43.90	58.07	58.12	**60.78**	48.46	60.13	61.14	**62.42**
	32	56.23	63.48	63.78	**65.68**	54.23	64.90	66.39	**71.67**		32	45.86	59.85	59.35	**61.71**	55.11	62.42	62.64	**63.28**
	64	66.10	**74.89**	73.96	74.88	67.78	76.84	79.04	**79.23**		64	51.14	61.35	59.95	**62.72**	57.20	64.28	63.57	**64.53**
Banking77	128	75.32	81.97	81.53	**82.43**	75.43	83.49	86.06	**86.24**	WIC	128	59.27	62.78	60.42	**63.50**	63.17	65.24	66.14	**66.64**
	256	85.82	85.51	85.76	**86.78**	85.84	86.45	87.59	**87.64**		256	61.29	63.71	60.76	**64.50**	62.93	66.42	66.92	**66.98**
	512	87.78	87.20	87.04	**88.15**	88.34	87.85	88.93	**89.22**		512	62.23	63.14	61.50	**64.01**	65.78	65.85	**67.77**	67.12
	768	88.29	87.70	**88.89**	88.76	89.15	88.43	**89.83**	89.45		768	63.21	63.07	61.85	**64.37**	65.71	65.91	**67.42**	67.28
	16	35.12	53.66	52.71	**60.57**	48.85	55.96	50.73	**60.38**		16	39.27	59.07	58.42	**61.03**	45.75	59.35	61.50	**62.07**
	32	46.06	59.08	59.17	**63.32**	55.50	60.51	54.75	**62.30**		32	51.06	65.08	64.98	**67.38**	51.86	63.34	62.36	**64.24**
	64	49.72	60.22	65.16	**65.36**	63.27	65.68	60.55	**65.64**		64	55.38	**68.47**	66.92	67.32	52.97	66.11	62.07	**66.26**
TweetEval	128	55.89	66.71	66.75	**67.61**	66.23	68.14	65.51	**68.24**	SICK	128	61.78	**69.84**	68.73	68.46	59.33	66.57	66.74	**67.00**
	256	63.54	68.03	**69.93**	69.29	68.12	69.38	68.73	**70.45**		256	68.93	68.85	69.03	**70.82**	65.72	67.23	66.43	**67.44**
	512	66.47	68.25	69.32	**70.43**	70.02	70.86	70.22	**71.58**		512	70.46	70.09	70.57	**70.80**	67.41	**68.33**	67.04	68.02
	768	70.16	69.10	70.02	**70.89**	70.22	70.72	70.75	**71.26**		768	70.55	70.00	**70.67**	70.65	68.15	**68.94**	68.19	68.73
	16	51.32	61.12	65.12	**72.34**	57.83	65.93	66.73	**72.46**		16	45.63	60.00	57.45	**62.43**	49.24	61.65	59.39	**63.43**
	32	57.34	69.87	69.38	**74.02**	66.72	70.11	71.07	**73.33**		32	58.78	63.51	60.12	**65.47**	51.34	65.09	64.39	**66.16**
	64	68.25	72.28	70.12	**74.55**	70.53	70.14	72.57	**73.68**		64	58.37	65.45	65.78	**67.37**	61.22	67.32	67.51	**68.02**
MRPC	128	68.15	72.34	72.89	**73.97**	73.56	72.96	71.71	**74.02**	STSB	128	60.83	67.16	68.52	**68.81**	67.53	69.59	70.12	**70.55**
	256	71.82	72.52	73.97	**74.37**	73.73	72.98	72.29	**74.14**		256	66.61	67.79	68.36	**69.18**	68.23	70.41	**70.91**	70.10
	512	72.16	72.76	74.31	**74.67**	74.08	73.16	73.27	**74.62**		512	67.82	68.85	68.92	**69.54**	70.67	**71.33**	70.83	71.22
	768	72.51	72.87	74.02	**74.60**	74.02	73.22	72.46	**74.44**		768	69.45	68.93	69.02	**69.74**	71.58	71.30	**71.99**	71.96

Table 1: Results on in-domain datasets with TinyBERT 6L and BERT backbone embedding models. Bold indicates the best result at the same representation size, backbone model, and dataset.

Datasets	Dim.	TinyBERT 6L				BERT				Datasets	Dim.	TinyBERT 6L				BERT			
		Unsup SimCSE	MRL	ESE	MIPIC	Unsup SimCSE	MRL	ESE	MIPIC			Unsup SimCSE	MRL	ESE	MIPIC	Unsup SimCSE	MRL	ESE	MIPIC
	16	45.68	51.48	50.90	**59.34**	47.88	55.13	51.34	**62.08**		16	51.26	59.21	60.26	**61.05**	50.78	54.78	59.67	**64.32**
	32	47.09	55.25	56.02	**61.60**	53.84	59.78	53.78	**64.87**		32	53.46	63.45	65.63	**66.23**	53.34	60.96	64.54	**68.00**
	64	53.48	55.67	58.32	**62.70**	61.22	61.24	55.23	**65.88**		64	60.72	66.29	66.04	**68.43**	57.23	63.45	68.15	**69.32**
STS12	128	59.50	59.80	60.38	**63.74**	65.45	62.34	59.97	**66.32**	STS16	128	66.98	67.90	68.12	**69.86**	65.89	66.78	70.14	**70.30**
	256	61.14	60.17	61.23	**63.82**	65.83	64.13	60.33	**67.08**		256	67.25	68.10	**70.55**	69.92	66.86	69.89	70.82	**70.86**
	512	61.77	61.52	60.82	**64.14**	66.16	65.09	61.06	**67.55**		512	68.94	68.60	**70.87**	70.57	70.25	70.60	**71.47**	70.78
	768	62.19	61.78	64.16	**64.57**	66.31	64.84	60.43	**67.64**		768	69.26	68.69	70.25	**70.93**	71.07	70.53	70.54	**71.56**
	16	49.58	55.03	53.51	**64.30**	55.34	62.13	60.48	**65.39**		16	57.35	60.43	60.79	**61.17**	55.67	63.45	64.06	**64.78**
	32	50.60	63.48	60.46	**67.51**	61.23	63.44	65.78	**68.81**		32	66.01	66.71	66.88	**67.45**	57.23	65.92	66.77	**67.34**
	64	58.43	62.96	64.56	**68.57**	67.88	68.98	68.97	**71.22**		64	68.43	**68.45**	68.30	67.93	60.86	68.25	69.09	**70.89**
STS13	128	60.93	67.88	66.40	**68.73**	71.09	70.97	71.68	**71.87**	SickR	128	70.02	69.82	69.39	**70.73**	64.23	69.61	70.30	**70.43**
	256	61.24	67.80	68.53	**69.84**	71.88	71.24	72.32	**72.62**		256	70.20	69.87	69.63	**70.55**	66.78	70.49	71.08	**72.56**
	512	67.92	68.96	69.65	**70.22**	72.13	72.35	**73.63**	73.33		512	70.49	70.16	70.38	**71.03**	67.34	70.83	71.48	**71.78**
	768	70.54	69.16	69.94	**70.87**	74.78	73.56	73.81	73.49		768	70.58	70.10	70.49	**71.94**	67.89	70.98	71.58	**72.56**
	16	49.35	50.42	50.12	**57.27**	50.67	51.76	54.20	**56.93**		16	28.44	31.30	29.76	**32.87**	24.78	26.98	27.95	**30.24**
	32	48.23	53.67	53.27	**60.09**	57.23	54.78	56.89	**61.45**		32	32.65	37.94	36.11	**41.56**	34.15	36.87	37.49	**41.55**
	64	51.28	59.57	57.64	**61.32**	60.86	60.88	60.44	**63.13**		64	41.35	43.53	44.29	**51.15**	42.46	42.52	44.61	**50.97**
STS14	128	55.98	60.84	61.42	**62.15**	61.25	62.05	61.45	**63.97**	Emotion	128	46.39	49.50	52.37	**55.25**	49.76	49.15	50.37	**53.43**
	256	60.13	60.91	62.36	**62.84**	61.56	63.68	63.81	**64.89**		256	52.53	53.92	56.12	**57.55**	54.42	**54.67**	54.33	53.75
	512	**63.93**	62.22	63.47	63.22	62.45	64.56	64.63	**65.43**		512	54.93	56.39	56.82	**60.87**	54.36	55.98	56.63	**57.98**
	768	63.01	62.32	**63.85**	63.31	65.58	64.65	64.85	65.37		768	55.30	58.53	**60.43**	60.39	53.78	54.08	57.42	**58.65**
	16	60.92	64.17	63.84	**66.85**	61.45	61.99	64.78	**68.85**		16	65.34	72.25	71.87	**73.97**	66.34	68.45	69.61	**72.48**
	32	65.45	69.88	69.23	**71.05**	65.34	70.62	69.46	**71.35**		32	66.45	74.92	73.32	**75.54**	70.12	68.78	73.51	**74.01**
	64	69.32	72.51	72.08	**73.20**	65.78	73.26	72.64	**74.47**		64	70.73	75.40	73.75	**75.67**	71.23	71.34	**75.02**	73.96
STS15	128	70.54	73.24	73.75	**74.41**	66.23	75.28	**75.63**	74.72	SciTail	128	71.88	75.68	74.90	**75.78**	74.78	72.56	75.77	**75.89**
	256	74.76	73.29	**76.12**	75.21	71.32	76.24	**76.93**	75.94		256	72.45	75.92	75.53	**75.97**	76.95	75.77	76.29	**77.24**
	512	**75.38**	73.87	75.24	76.23	74.56	**76.73**	76.12	76.38		512	72.84	**75.82**	75.77	75.34	76.66	75.72	76.38	**76.89**
	768	75.43	73.90	76.24	**76.34**	**77.39**	76.62	76.94	76.22		768	75.54	**75.90**	75.72	75.61	76.57	76.34	**76.85**	76.54

Table 2: Results on out-domain datasets with TinyBERT 6L and BERT backbone embedding models. Bold indicates the best result at the same representation size, backbone model, and dataset.

4 Experiments

4.1 Experimental Setup

We evaluate embeddings across in-domain and out-of-domain settings to assess both task performance and generalization.

Task categories. We report results across three representative task families, all of which rely critically on embedding quality. First, for **Text Classification**, which requires models to capture the overall semantics of a single text input, we evaluate on TweetEval (Barbieri et al., 2020), as well as Emotion and Banking77 from the MTEB benchmark (Muennighoff et al., 2023). Second, regarding **NLI**, which focuses on modeling semantic relationships between two inputs, we consider MRPC from GLUE (Wang et al., 2018), WiC (Pilehvar and Camacho-Collados, 2019), and SciTail (Khot et al., 2018). Finally, for **Semantic Textual Similarity (STS)**, which measures the ability to capture fine-grained semantic similarity, we evaluate on STS-B and SICK (Marelli et al., 2014) for in-domain testing, and extend to STS12–16 and SickR (Muennighoff et al., 2023) for OOD evaluation. Additional details regarding model architectures, training procedures, datasets, and evaluation protocols are provided in Appendix A.

Baselines. We compare against SOTA Matryoshka baselines: **MRL** (Kusupati et al., 2022),

which learns nested prefixes via multi-scale supervision, and **ESE** (LI et al., 2025), which scales across both dimensionality and model depth. These provide a strong benchmark for evaluating low-dimensional representation quality.

4.2 Results

Results for in-domain and out-of-domain benchmarks (Table 1 and Table 2) indicate that MIPIC achieves competitive or superior performance compared to baselines, including Unsup SimCSE, MRL, and Espresso. While maintaining comparable efficacy at higher dimensions, the performance gap becomes particularly distinct at extremely low dimensions (16 and 32), demonstrating the framework's superior capability in semantic compression. These trends hold consistent across TinyBERT and BERT architectures. Furthermore, scalability evaluations on larger backbones such as BGEM3 and Qwen3 0.6B (Appendix C) validate the method's robustness, confirming that while MIPIC maintains parity with baselines in full-capacity regimes, it significantly excels in retaining semantic density under extreme truncation.

5 Analysis

We analyze the individual contributions of SIA and PIC, alongside the effectiveness of our progressive dimension scaling strategy to validate MIPIC's design. Further investigations, including layer selection impacts, are detailed in Appendix B.

Impact of Framework Components Table 3 highlights the synergy between SIA and PIC. SIA is vital for structural organization; its removal causes sharp drops at small sizes, proving geometric order is key for information density. Meanwhile, PIC acts as a semantic bridge to stabilize task knowledge across layers. While SIA ensures geometric efficiency, PIC prevents semantic loss in early stages. Together, they enable a coarse-to-fine refinement that maintains high quality at every dimension.

Effectiveness of Progressive Dimension Design Table 4 shows that progressively increasing dimensions works better than keeping all layers at full size. Using full dimensions everywhere removes the bottleneck needed for Matryoshka learning. Our design forces early layers to pack the most important information into small prefixes, then refine it later. This leads to more stable and effective representations. Without this constraint, the model does not learn to focus on key information early, which hurts performance.

Datasets	Rep. size	MIPIC	MIPIC w/o SIA	MIPIC w/o PIC
STS12	16	**62.08**	60.12	61.27
	32	**64.87**	62.39	63.90
	64	**65.88**	64.82	65.54
	128	**66.32**	65.34	65.72
	256	**67.08**	66.87	66.97
	512	**67.55**	66.43	67.12
	768	**67.64**	67.12	67.53
TweetEval	16	**60.38**	57.12	59.13
	32	**62.30**	59.32	61.87
	64	**65.64**	65.02	65.11
	128	**68.24**	67.35	67.92
	256	**70.45**	69.54	70.12
	512	**71.58**	70.24	71.55
	768	**71.26**	70.22	70.37

Table 3: Ablation study results showing the impact of SIA and PIC. "w/o" denotes "without".

Datasets	Rep. size	Our design	All dim equal
STS12	16	**62.08**	61.98
	32	**64.87**	63.12
	64	**65.88**	64.77
	128	**66.32**	66.12
	256	**67.08**	66.52
	512	**67.55**	67.12
	768	**67.64**	67.56
Tweet Eval	16	**60.38**	58.99
	32	**62.30**	61.97
	64	**65.64**	65.33
	128	**68.24**	67.92
	256	**70.45**	70.08
	512	**71.58**	71.26
	768	**71.26**	71.15

Table 4: Ablation study on PIC design. "Our design" uses progressive dimension scaling across layers, while "All dim equal" maintains full dimensions at every stage.

6 Conclusion

We proposed MIPIC, a framework that coordinates Matryoshka Representation Learning across both embedding dimensions and network depth. By combining Self-Distilled Intra-Relational Alignment (SIA) for structural consistency and Progressive Information Chaining (PIC) for semantic consolidation, MIPIC establishes a global chaining of task signals. Extensive experiments on models ranging from TinyBERT to Qwen3 and BGE-M3 show that MIPIC remains robustly competitive with state-of-the-art baselines at full capacity, while achieving superior efficacy in extreme 16 and 32-dimensional compression.

7 Limitations

Despite its effectiveness, MIPIC introduces certain limitations, primarily the increased computational overhead during the training phase due to the calculation of auxiliary CKA and InfoNCE losses across multiple layers. The framework's performance also relies on the careful tuning of hyperparameters and the strategic selection of layer checkpoints, which may require additional optimization when adapting to backbone architectures with different depths. Furthermore, while we validated MIPIC on a wide array of discriminative NLP tasks using encoder-based models, its utility in decoder-only generative scenarios or multi-modal settings has not yet been explored. Lastly, our current progressive dimension scaling design represents a specific configuration for depth-wise transfer, and alternative scheduling methods for information condensation could potentially yield different results.

References

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online. Association for Computational Linguistics.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. LLM2Vec: Large language models are secretly powerful text encoders. In *First Conference on Language Modeling*.

Mu Cai, Jianwei Yang, Jianfeng Gao, and Yong Jae Lee. 2024. Matryoshka multimodal models.

Alexis Conneau and Douwe Kiela. 2018. Senteval: An evaluation toolkit for universal sentence representations.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2022. Simcse: Simple contrastive learning of sentence embeddings.

Jiatao Gu, Shuangfei Zhai, Yizhe Zhang, Josh Susskind, and Navdeep Jaitly. 2024. Matryoshka diffusion models.

David R. Hardoon, Sandor Szedmak, and John Shawe-Taylor. 2004. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12):2639–2664.

Liyang He, Chenglong Liu, Rui Li, Zhenya Huang, Shulan Ruan, Jun Zhou, and Enhong Chen. 2025. Refining sentence embedding model through ranking sentences generation with large language models.

Wenbo Hu, Zi-Yi Dou, Liunian Harold Li, Amita Kamath, Nanyun Peng, and Kai-Wei Chang. 2024. Matryoshka query transformer for large vision-language models.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. SciTail: A textual entailment dataset from science question answering. In *AAAI*.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. 2019. Similarity of neural network representations revisited.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and Ali Farhadi. 2022. Matryoshka representation learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 30233–30249. Curran Associates, Inc.

Xianming LI, Zongxi Li, Jing Li, Haoran Xie, and Qing Li. 2025. ESE: Espresso sentence embeddings. In *The Thirteenth International Conference on Learning Representations*.

Ziyue Li and Tianyi Zhou. 2025. Your mixture-of-experts LLM is secretly an embedding model for free. In *The Thirteenth International Conference on Learning Representations*.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. 2014. SemEval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 1–8, Dublin, Ireland. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2023. Mteb: Massive text embedding benchmark.

Sosuke Nishikawa, Ryokan Ri, Ikuya Yamada, Yoshimasa Tsuruoka, and Isao Echizen. 2022. EASE: Entity-aware contrastive learning of sentence embedding. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3870–3885, Seattle, United States. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1267–1273, Minneapolis, Minnesota. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

Chongyang Tao, Tao Shen, Shen Gao, Junshuo Zhang, Zhen Li, Kai Hua, Wenpeng Hu, Zhengwei Tao, and Shuai Ma. 2025. Llms are also effective embedding models: An in-depth overview.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers.

A Experimental Details

A.1 Model Architectures

To evaluate the scalability and generalizability of the MIPIC framework, we conduct experiments across a diverse range of backbone architectures varying in size and complexity. These include the compact **TinyBERT-6L** with 6 Transformer layers, the standard 12-layer **BERT-base** encoder, and modern large-scale models such as the **Qwen3-0.6B embedding** and the high-performance **BGE-M3**. By spanning from small-scale encoders to large-scale language model embeddings, we demonstrate that MIPIC consistently optimizes the internal organization of semantic information regardless of the model's depth or total parameter count.

A.2 Detailed Dataset Statistics

Table 5: Dataset Statistics for training and test set

Dataset	Train (Sampled)	Test Size
Banking77	3,000	3,080
TweetEval	3,000	3434
Emotion (OOD)	-	1,990
MRPC	1,500	1,730
WiC	1,500	1,400
SciTail (OOD)	-	2,130
SICK	1,500	4,823
STS-B	1,500	1,390
STS12 (OOD)	-	3,108
STS13 (OOD)	-	1,500
STS14 (OOD)	-	3,750
STS15 (OOD)	-	3,000
STS16 (OOD)	-	1,186
SickR (OOD)	-	9,927

We use a diverse collection of datasets for both training and evaluation, covering text classification, natural language inference (NLI), and semantic textual similarity (STS). To build training data for contrastive sentence representation learning, we sample sentences from multiple task categories to promote domain and objective diversity. Specifically, we collect 6,000 sentences from classification datasets (3,000 per dataset), 3,000 sentence pairs from STS datasets (1,500 per dataset), and 3,000 sentence pairs from pair classification datasets (1,500 per dataset). All sentence pairs are flattened into individual sentences, resulting in 24,000 unique sentences, which are then used to train the backbone encoder with unsupervised SimCSE-style contrastive learning under a unified training framework. For evaluation, we test the trained models on both in-domain test sets and unseen out-of-domain datasets, including Emotion, SciTail, and multiple STS benchmarks (STS12–STS16 and SickR), which differ from the training data in domain, style, and annotation protocol. Dataset statistics are reported in Table 5, and all baseline methods (MRL, ESE) are trained on the same training corpus and evaluated on the same set of test datasets for fair comparison

A.3 Training Configurations

The detailed training configurations for the MIPIC framework across our various backbones are summarized in Table 6. We maintain a consistent setup to ensure a fair comparison across different scales, applying specific hyperparameters like α to trade off MRL and MIPIC losses. We explored the loss balancing hyperparameter α over the set $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7\}$. The optimal configurations for each backbone are also reported in Table 6.

Table 6: Detailed training configurations for MIPIC across different backbones.

Configuration	TinyBERT-6L	BERT-base	Qwen3-0.6B	BGE-M3
Epochs	5	5	5	5
Learning Rate	2×10^{-5}	2×10^{-5}	2×10^{-5}	2×10^{-5}
Max Length	256	256	256	256
Batch Size	16	16	16	16
LR Scheduler	Cosine	Cosine	Cosine	Cosine
Optimizer	AdamW	AdamW	AdamW	AdamW
Temperature (τ)	0.05	0.05	0.05	0.05
α (MRL weight)	0.4	0.4	0.5	0.5

Task Objective Specification. In our experimental setting, the task-specific loss component within the MRL objective (Eq. 1) is instantiated as $\mathcal{L}_{\text{SimCSE}}$, the unsupervised contrastive loss adopted from (Gao et al., 2022) . Given a batch of input sentences, we feed them through the student encoder twice with different standard dropout masks z, z' to obtain two views of embeddings e_i^z and $e_i^{z'}$. The loss is formulated as:

$$\mathcal{L}_{\text{SimCSE}} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{\text{sim}(e_i^z, e_i^{z'})/\tau}}{\sum_{j=1}^{N} e^{\text{sim}(e_i^z, e_j^{z'})/\tau}}, \tag{19}$$

where N is the batch size, τ is the temperature hyperparameter, and $\text{sim}(\cdot)$ denotes cosine similarity.

A.4 Evaluation

To assess the efficacy of the learned sentence embeddings, we conduct evaluations across three distinct categories of downstream tasks. First, for classification tasks, we follow the standard protocol established by (Conneau and Kiela, 2018), which involves training a Logistic Regression classifier on top of frozen sentence representations. Second, in pair classification, we generate predictions by applying an optimal threshold to the cosine similarity of the sentence pairs, measuring performance via Accuracy. Finally, for Semantic Textual Similarity (STS), we evaluate the alignment between the embeddings' cosine similarity and human-annotated gold standards using Spearman correlation. In our method, all models are fully fine-tuned using the unified objective $\mathcal{L}_{total}$. We evaluate the quality of Matryoshka representations across a fixed set of nested dimensions $\mathcal{D} = \{16, 32, 64, 128, 256, 512, 768/1024\}$. For Text Classification, we report F1 Score. For NLI tasks, we report Accuracy. For Semantic Textual Similarity (STS) benchmarks, we report the Spearman Correlation Coefficient.

A.5 SIA and PIC hyperparameters

Top-k Schedule Specification. To ensure reproducibility and robustness, we implement a deterministic linear schedule for the token selection threshold k_i. For our standard backbone configurations with a representation hierarchy $\mathcal{D} = \{16, 32, 64, 128, 256, 512, 768\}$, we align the first 6 lower-dimensional prefixes against the full-dimensional teacher. We set $k_i = \max(8, \lceil \gamma_i \cdot m \rceil)$, where m is the sequence length. The ratio γ_i increases monotonically with the dimension size: specifically, we utilize $\gamma = [0.2, 0.3, 0.4, 0.5, 0.6, 0.7]$ corresponding to the dimensions $[16, 32, 64, 128, 256, 512]$. This schedule ensures that the most compressed representations ($d = 16$) focus on the top 20% salient tokens, while larger prefixes progressively incorporate up to 70% of the context, with a minimum floor of $k_{\min} = 8$ tokens to preserve basic sentence structure in short inputs.

Layers and checkpoints applied in MIPIC In our framework, let $\mathcal{L}$ denote the set of layers applied in MIPIC, and $\mathcal{C}$ represent the set of checkpoints applied in PIC, defined as tuples (d, l) where d indicates the target dimension and l corresponds to the layer index. The specific configurations of $\mathcal{L}$ and $\mathcal{C}$ are adapted to the architecture and depth of each model. For TinyBERT-6L, we utilize all layers, setting $\mathcal{L} = \{1, 2, 3, 4, 5, 6\}$ with checkpoints $\mathcal{C} = \{(16, 1), (32, 2), (64, 3), (256, 4), (512, 5),$

$(768, 6)\}$. For BERT-base, we select distributed layers $\mathcal{L} = \{2, 4, 6, 8, 9, 10, 12\}$, associated with $\mathcal{C} = \{(16, 2), (32, 4), (64, 6), (128, 8), (256, 9), (512, 10), (768, 12)\}$. For the larger BGE-M3 model, we employ $\mathcal{L} = \{1, 4, 7, 11, 15, 19, 24\}$ and checkpoints $\mathcal{C} = \{(16, 1), (32, 4), (64, 7), (128, 11), (256, 15), (512, 19), (1024, 24)\}$. Finally, for Qwen3, the configuration is set to $\mathcal{L} = \{2, 6, 12, 16, 20, 24, 28\}$ with $\mathcal{C} = \{(16, 2), (32, 6), (64, 12), (128, 16), (256, 20), (512, 24), (1024, 28)\}$. A detailed ablation study on the impact of layer selection is provided in Section B.1.

B Additional ablation study

B.1 Impact of Layer Selection Strategy

Datasets	Rep. size	L_1 (Distributed)	L_2 (Deep)	L_3 (Shallow)
Banking77	16	**75.38**	73.87	71.53
	32	**86.48**	84.83	82.06
	64	90.74	**90.76**	89.54
	128	92.94	92.74	**92.99**
	256	**92.89**	92.35	92.06
	512	92.98	**93.41**	92.28
	1024	92.81	**93.02**	92.87
STS12	16	**62.96**	61.70	61.46
	32	**65.71**	64.39	64.16
	64	**66.61**	65.52	64.30
	128	**68.28**	68.09	67.41
	256	**70.93**	70.05	69.53
	512	**70.85**	70.31	69.49
	1024	**70.31**	70.28	69.73

Table 7: Ablation study on layer selection for MIPIC using the BGE-M3 backbone.

The results in Table 7 demonstrate that distributing MIPIC across the entire network depth ($L_1 = [1, 4, 7, 11, 15, 19, 24]$) is superior to focusing solely on deep ($L_2 = [18, 19, 20, 21, 22, 23, 24]$) or shallow ($L_3 = [1, 2, 3, 4, 5, 6, 7]$) layers. While deep layers contain the richest task semantics, restricting MIPIC to the tail end of the backbone (L_2) fails to provide early guidance for low-dimensional prefixes, leading to suboptimal alignment in the initial stages. Conversely, focusing only on the early stages (L_3) lacks the high-level semantic signals necessary to construct a robust "semantic bridge" to the final task. By selecting checkpoints that span from the earliest to the final layers, L_1 maintains a continuous and stable flow of task information. This global chaining ensures that even the most compressed representations are grounded in deep semantic knowledge through a step-by-step refinement process throughout the backbone.

B.2 Impact of Hard Top-k Token Selection

To investigate the necessity of our selective alignment strategy, we compare the standard MIPIC framework (employing Hard Top-k selection) against a variant where the student aligns using the full set of tokens ("Full Token Alignment") regardless of the dimension size. This ablation was conducted on the BGE-M3 backbone using Banking77 and STS12 benchmarks, as detailed in Table 8. The results clearly demonstrate the critical advantage of the Hard Top-k selection strategy, particularly in highly compressed regimes. At lower dimensions (16, 32, and 64), MIPIC significantly outperforms the full-token alignment baseline. This confirms our intuition that low-capacity representations suffer from information saturation when forced to encode the entire sequence topology. By filtering for only the most salient tokens, the Top-k strategy effectively reduces noise, allowing the limited dimension space to concentrate exclusively on the semantic skeleton of the input. Conversely, at the full capacity (1024 dimensions), the Full Token approach yields marginally higher performance. This is expected, as the high-dimensional space possesses sufficient bandwidth to encode the complete token granularity without saturation. However, MIPIC remains robustly competitive even at this scale, while establishing a decisive superiority where Matryoshka learning matters most: in the extreme compression scenarios.

Datasets	Rep. Size	MIPIC (Top-k CKA)	Full Token Alignment
Banking77	16	**75.38**	72.45
	32	**86.48**	83.42
	64	**90.74**	88.11
	128	**92.94**	90.56
	256	**92.89**	92.34
	512	**92.98**	92.95
	1024	92.81	**92.98**
STS12	16	**62.96**	60.67
	32	**65.71**	63.23
	64	**66.61**	64.63
	128	**68.28**	67.98
	256	**70.93**	70.78
	512	**70.85**	70.56
	1024	70.31	**70.62**

Table 8: Ablation study on the Hard Top-k selection strategy using the BGE-M3 backbone. We compare our adaptive Top-k approach against aligning the full token sequence at every dimension.

B.3 Impact of Components in SIA

To validate the individual contributions of the internal mechanisms within the Self-Distilled Intra-Relational Alignment (SIA) module, we conducted an ablation study on the BERT-base backbone using the STS12 and TweetEval benchmarks. The results are presented in Table 9. As observed, removing either Attention Matching or Top-k CKA leads to a degradation in performance, particularly in low-dimensional regimes (16 and 32), confirming that both components are essential for maintaining representation quality. However, the omission of Top-k CKA results in a significantly sharper performance drop compared to removing Attention Matching. This disparity highlights the fundamental role of Top-k CKA: while Attention Matching ensures that the student focuses on the same salient tokens as the teacher, it is the Top-k CKA mechanism that explicitly reorganizes the geometric structure of the information. By enforcing topological consistency between the teacher's full state and the student's truncated prefix, Top-k CKA effectively compresses the rich semantic landscape into the bottleneck. Without this structural reorganization, the lower-dimensional representation struggles to encapsulate the necessary information density, leading to the observed decline in effectiveness.

Datasets	Dim.	MIPIC	w/o Attn. Match	w/o Top-k CKA
STS12	16	**62.08**	61.54	60.11
	32	**64.87**	64.14	63.67
	64	**65.88**	65.98	64.78
	128	**66.32**	65.94	65.11
	256	**67.08**	66.91	66.57
	512	**67.55**	67.21	67.11
	768	**67.64**	67.61	67.53
TweetEval	16	**60.38**	60.12	59.42
	32	**62.30**	61.23	60.96
	64	**65.64**	65.44	64.58
	128	**68.24**	67.55	67.11
	256	**70.45**	70.02	69.87
	512	**71.58**	71.23	70.34
	768	**71.26**	71.24	70.97

Table 9: Ablation study on SIA components using BERT-base backbone. We compare the full MIPIC framework against variants removing Attention Matching ("w/o Attn. Match") and Top-k CKA alignment ("w/o Top-k CKA").

B.4 Training time analysis

We explicitly analyze the computational cost of our framework compared to the MRL and ESE baselines using the BERT-base backbone. The training throughput (measured in iterations per second and samples per second) is reported in Table 10.

As indicated in Table 10, MIPIC exhibits a lower training throughput compared to baselines. Specifically, our method operates at approximately 2.88 iterations/s, representing a reduction of roughly 45% compared

Method	Iterations / s	Throughput (sample/s)
MRL	7.61	121.76
ESE	5.24	83.84
MIPIC (Ours)	2.88	46.08

Table 10: Training efficiency comparison on the BERT-base backbone. MIPIC incurs higher training latency due to the computation of auxiliary alignment losses (SIA and PIC).

to ESE (5.24 it/s) and 62% compared to the standard MRL (7.61 it/s). This increased overhead is expected, as MIPIC requires additional forward passes and gradient computations for the multi-layer alignment objectives (SIA and PIC) involving CKA and InfoNCE calculations. However, it is crucial to distinguish between *training cost* and *inference latency*. The computational overhead shown above is strictly a **one-time training investment**. Structurally, MIPIC does not introduce any additional parameters or modules to the backbone encoder during inference. All auxiliary projectors (e.g., P_i in SIA, ϕ_i in PIC) are discarded after training. Consequently, a deployed MIPIC model possesses **zero additional inference latency** compared to a standard backbone model. Given that representation models are typically trained once but queried millions of times, we argue that the increased training time is a justifiable trade-off for the significant gains in representation quality and compression efficiency demonstrated in our main experiments.

C Additional Results for larger model

Table 11: Performance comparison on in-domain datasets using BGEM3 and Qwen 3 (0.6B) backbones. **Bold** indicates the best performance for a given representation size.

		BGEM3			Qwen					BGEM3			Qwen		
Dataset	**Dim**	MRL	ESE	MIPIC	MRL	ESE	MIPIC	**Dataset**	**Dim**	MRL	ESE	MIPIC	MRL	ESE	MIPIC
	16	73.06	72.07	**75.38**	55.02	56.94	**58.45**		16	61.35	61.53	**61.92**	61.71	62.92	**63.71**
	32	86.09	84.91	**86.48**	77.72	78.29	**78.51**		32	61.78	61.23	**62.45**	63.78	63.64	**64.35**
	64	90.53	90.67	**90.74**	84.31	83.44	**85.82**		64	62.14	63.12	**63.91**	64.57	64.35	**66.28**
Banking77	128	92.10	92.02	**92.94**	87.87	87.25	**88.84**	WiC	128	63.85	64.32	**64.57**	64.42	64.14	**65.79**
	256	92.71	92.06	**92.89**	89.62	89.67	**90.55**		256	64.21	64.28	**64.69**	65.00	64.07	**66.21**
	512	92.93	92.13	**92.98**	91.61	**91.42**	91.32		512	64.71	64.21	**65.04**	64.57	64.42	**66.42**
	1024	**93.11**	92.25	92.81	91.81	91.98	**92.16**		1024	65.38	64.42	**65.58**	63.28	64.14	**66.07**
	16	56.91	55.32	**58.08**	55.15	53.35	**57.96**		16	69.66	68.35	**69.89**	64.15	64.45	**66.37**
	32	61.72	59.95	**61.82**	61.83	59.11	**62.13**		32	71.03	69.48	**71.08**	68.32	67.66	**69.14**
	64	66.80	65.95	**67.43**	64.63	63.24	**65.41**		64	71.63	70.27	**71.92**	69.12	68.92	**69.89**
TweetEval	128	70.06	69.67	**70.98**	67.51	65.99	**68.35**	SICK	128	71.93	70.83	**72.50**	69.68	69.54	**69.79**
	256	72.66	71.90	**72.84**	70.23	67.21	**70.91**		256	72.46	71.08	**72.84**	69.45	69.74	**69.82**
	512	72.61	72.69	**72.98**	71.09	69.67	**71.83**		512	72.78	71.24	**73.14**	69.53	69.03	**69.98**
	1024	72.95	**74.13**	73.56	71.82	69.32	**71.94**		1024	72.77	71.23	**73.15**	69.33	69.09	**69.76**
	16	67.21	67.43	**67.65**	68.11	67.18	**69.69**		16	75.21	73.96	**76.81**	67.83	67.66	**68.44**
	32	67.36	67.65	**67.69**	68.34	67.24	**69.15**		32	77.71	76.13	**78.30**	71.82	70.55	**72.65**
	64	67.47	67.69	**67.73**	68.63	67.59	**69.57**		64	79.50	77.61	**80.30**	73.22	72.39	**74.36**
MRPC	128	67.39	67.47	**67.71**	69.04	67.53	**69.10**	STSB	128	80.59	78.55	**81.27**	75.17	74.82	**75.99**
	256	67.42	**67.88**	67.76	68.75	67.65	**70.68**		256	**82.44**	79.40	82.20	75.67	74.05	**75.92**
	512	67.71	67.59	**67.89**	70.45	67.88	70.21		512	81.28	80.94	**83.07**	76.07	74.03	**76.65**
	1024	67.59	67.53	**67.97**	**70.98**	68.05	70.73		1024	82.52	80.52	**83.39**	75.76	74.29	**76.85**

The expansion of our experiments to high-capacity models, specifically Qwen3-0.6B embedding and BGE-M3, confirms the scalability and robustness of the MIPIC framework. The results demonstrate that the synergy between Self-Distilled Intra-Relational Alignment (SIA) and Progressive Information Chaining (PIC) remains highly effective as model parameters and depth increase, transitioning successfully from standard encoders to large-scale architectures. Extensive evaluations across nearly all benchmarks indicate that MIPIC consistently outperforms established baselines such as MRL and ESE. This persistent superiority suggests that our global chaining strategy effectively captures and organizes the complex semantic structures inherent in larger models, ensuring that task-specific signals are preserved throughout the network's internal representation hierarchy regardless of the model size. Key observations from

Table 12: Performance comparison on out-of-domain datasets using BGEM3 and Qwen 3 (0.6B) backbones. **Bold** indicates the best performance for a given representation size.

		BGEM3			Qwen					BGEM3			Qwen		
Dataset	**Dim**	MRL	ESE	MIPIC	MRL	ESE	MIPIC	**Dataset**	**Dim**	MRL	ESE	MIPIC	MRL	ESE	MIPIC
	16	62.95	61.97	**63.87**	59.41	58.10	**62.96**		16	73.16	72.77	**74.56**	67.39	67.98	**68.17**
	32	67.02	64.78	**67.67**	63.70	61.15	**65.71**		32	74.67	75.54	**75.68**	70.88	70.12	**71.85**
	64	67.99	65.85	**68.51**	65.93	63.30	**66.61**		64	76.96	77.10	**78.13**	72.84	71.63	**74.41**
STS12	128	69.38	66.83	**69.90**	67.31	66.89	**68.28**	STS16	128	78.88	77.75	**79.26**	74.29	73.28	**75.85**
	256	70.82	67.73	**71.06**	68.48	69.68	**70.93**		256	80.91	78.48	**80.93**	75.00	74.71	**76.67**
	512	71.64	68.80	**71.76**	68.93	68.01	**70.85**		512	82.13	80.20	**82.21**	75.26	75.61	**77.31**
	1024	72.11	68.56	**72.45**	**70.74**	69.28	70.31		1024	**82.31**	79.97	82.10	74.60	75.80	**76.83**
	16	75.32	73.32	**76.04**	67.83	68.09	**69.23**		16	70.49	69.87	**72.42**	63.81	63.52	**65.62**
	32	77.20	75.60	**78.76**	70.71	70.58	**73.59**		32	73.43	72.06	**74.83**	67.81	66.95	**68.32**
	64	79.66	77.88	**80.02**	73.65	73.23	**76.94**		64	75.43	72.76	**76.64**	69.08	68.96	**69.69**
STS13	128	80.96	78.98	**81.15**	75.27	75.89	**77.35**	SickR	128	76.52	75.12	**77.31**	69.75	69.63	**69.87**
	256	82.42	80.30	**82.49**	77.09	76.16	**77.48**		256	77.03	76.41	**77.34**	69.63	69.89	**69.96**
	512	83.54	81.76	**83.68**	78.52	77.82	**78.63**		512	**77.56**	76.55	77.43	**69.76**	69.14	69.75
	1024	84.03	81.79	**84.97**	**79.51**	78.51	78.82		1024	**77.68**	76.48	77.45	69.56	**69.65**	69.53
	16	67.07	65.03	**68.33**	59.40	59.97	**60.85**		16	29.92	28.46	**30.83**	29.39	28.62	**31.35**
	32	69.43	67.28	**70.65**	61.20	62.14	**63.14**		32	36.94	33.94	**38.94**	39.93	39.61	**41.92**
	64	72.46	68.96	**72.73**	64.43	63.68	**65.43**		64	43.45	44.53	**46.08**	47.44	**48.42**	48.03
STS14	128	73.38	70.18	**73.83**	66.01	66.95	**67.76**	Emotion	128	49.84	50.05	**52.40**	54.75	51.51	**55.61**
	256	74.45	71.39	**74.75**	67.44	67.32	**68.94**		256	54.07	53.41	**57.33**	58.44	**59.33**	58.77
	512	**74.93**	72.45	72.28	**68.73**	68.57	68.06		512	58.73	55.59	**60.67**	62.81	**64.79**	61.52
	1024	75.33	72.35	**75.65**	69.59	69.14	**69.97**		1024	61.39	60.12	**62.45**	**66.42**	66.26	65.15
	16	74.57	75.44	**76.79**	67.39	68.85	**71.61**		16	80.62	80.47	**82.47**	80.04	80.49	**82.42**
	32	78.46	77.98	**79.05**	70.88	71.11	**73.85**		32	81.74	81.32	**84.24**	82.59	81.74	**83.44**
	64	81.39	80.14	**81.94**	72.84	72.42	**75.14**		64	83.58	81.60	**84.51**	83.67	82.32	**84.23**
STS15	128	82.29	80.46	**83.10**	74.29	74.29	**77.66**	SciTail	128	84.43	82.03	**84.76**	84.24	83.50	**84.69**
	256	82.90	81.10	**83.74**	75.00	75.25	**78.83**		256	84.19	82.03	**84.57**	84.36	83.55	**84.79**
	512	83.19	81.45	**84.22**	75.26	76.44	**79.04**		512	85.03	82.30	**85.08**	84.38	84.16	**84.65**
	1024	83.62	81.69	**84.45**	74.60	76.86	**79.12**		1024	**85.46**	82.69	85.13	84.28	84.21	**84.54**

the large-scale evaluation highlight the remarkable resilience of MIPIC under extreme dimensional compression and its enhanced generalization capabilities. The performance gap between MIPIC and other methods is most pronounced at extremely low dimensions, such as 16 and 32, where it effectively condenses vast knowledge into compact prefixes while maintaining high task awareness. Furthermore, on out-of-domain (OOD) datasets like STS12–16 and SciTail, MIPIC maintains a consistent accuracy lead over current state-of-the-art baselines. This confirms that the depth-wise semantic consolidation provided by PIC allows large backbones to retain a stable semantic bridge, ensuring that even the most compressed representations remain grounded in robust, generalized knowledge across diverse tasks and domains.

D Use of Large Language Models

We acknowledge the use of Large Language Models (LLMs) during the preparation of this manuscript. These tools were utilized strictly for stylistic refinement, grammatical editing, and improving the overall flow of the presentation. At no point were LLMs used to conceptualize research ideas, formulate hypotheses, or interpret experimental findings. The authors remain fully accountable for the integrity and accuracy of the final content.