PibE-MPP: A Play-it-by-Ear Masking Performance Plug-in for LLMs

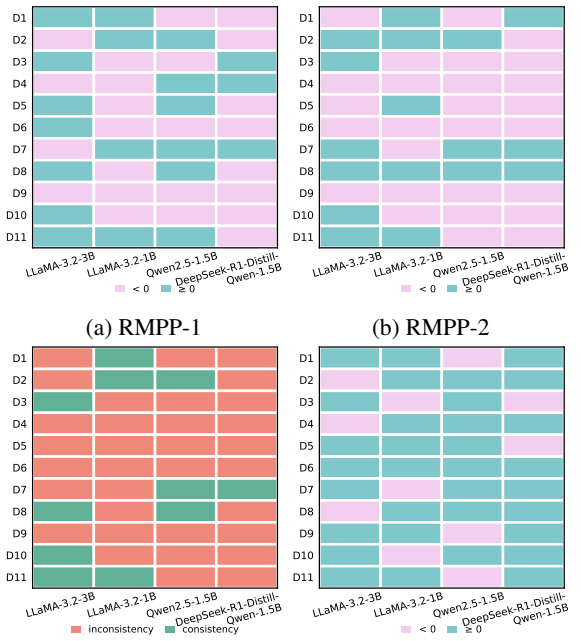
Mengwei Wang[1,*], **Simin Niu**[1,*], **Xun Liang**[1,†], **Yuefeng Ma**[2], **Sensen Zhang**[3], **Jiawei Yang**[1], **Shichao Song**[1], **Hanyu Wang**[1], **Huayi Lai**[1]
[1]School of Information, Renmin University of China
[2]School of Computer Science, Qufu Normal University
[3]College of Information Engineering, China Jiliang University
College of Modern Science and Technology

Abstract

Treating random masking as a performance plug-in for large language models (LLMs) offers three advantages: low coupling to the task, the model, and training resources. However, the critical drawback is that its gains are highly stochastic. Motivated by this, we propose **p**lay-**i**t-**b**y-**e**ar **m**asking **p**erformance **p**lug-in (PibE-MPP), which enables LLMs to adaptively select masking target combinations for each task, retaining these advantages and mitigating the drawback. Specifically, we pose two core questions: what are the masking targets, and what is the masking strategy, under 7 constraints obtained from these advantages and the drawback. For the first question, we select all attention heads in the last layer as masking targets by constructing a first-order Markov process with alternating hidden state and information fusion. The feasibility of this target is validated by random masking experiments. For the second question, we first construct a small yet interpretable candidate set by proposing a three-axis mapping and a mean-based criterion for fusion features of masking targets. We then propose an axis-variance minimization to select a compact masking-target combination, reducing sensitivity to outlier targets. Experiments on 6 LLMs (Qwen and LLaMA) and 24 datasets demonstrate PibE-MPP's effectiveness and generality, gain stability, and domain performance, and verify the necessity of its final module, providing empirical evidence of its transferability across tasks and models. The code is available at GitHub.

1 Introduction

Random masking is a widely adopted classic baseline in large language model (LLMs) research (Yao et al., 2024; Wu et al., 2025; Wang et al., 2025a; Kang et al., 2025). When *used as a performance plug-in*, it naturally exhibits three advantages: low coupling to the task (no impact on other tasks after completing a given task), the model (no architecture changes), and training resources (no training). Therefore, it is easy to transfer across tasks and models. However, its performance gains are highly stochastic and unpredictable (Fig. 1), the gains can fluctuate in signdirectly conflicting with the stability and reusability required by performance plug-ins.



(a) RMPP-1 (b) RMPP-2

(c) Two RMPP Consistency (d) PibE-MPP

Figure 1: Performance gain over baseline. D_i denotes the i-th dataset (see Table 1; same order). (a) and (b) Two random masking performance plug-in (RMPP-1, RMPP-2) gains are few and irregular whether fixing the model or the dataset. (c) RMPP consistency is also few and irregular whether fixing the model or the dataset. Thus, RMPP gains are highly stochastic. (d) PibE-MPP gains are more and stable whether fixing the model or the dataset.

Motivated by this, we propose **p**lay-**i**t-**b**y-**e**ar **m**asking **p**erformance **p**lug-in (PibE-MPP) that can adaptively generate masking target combina-

[*]Equal contribution
[†]Corresponding author

Findings of the Association for Computational Linguistics: ACL 2026, pages 2214–2225
July 2-7, 2026 ©2026 Association for Computational Linguistics

tions under a *given task×given model* condition, retaining the three advantages and delivering stable gains. Moreover, *from a methodological positioning perspective*, PibE-MPP is not intended to replace training-level improvement paradigms such as fine-tuning or other training-based methods (Hui et al., 2024; Yang et al., 2024a; Liu et al., 2025; DeepSeek-AI, 2025); instead, it is a *training-free* performance plug-in that can provide additional gains for models. Meanwhile, although PibE-MPP involves masking operations, our goal is *stable task performance improvement* rather than pruning-oriented compression and acceleration (Ashkboos et al., 2024; Le et al., 2025; He and Lin, 2025; Wang et al., 2025b), which in most cases is accompanied by performance degradation.

In PibE-MPP design, we pose two core questions: **what are the masking targets** and **what is the masking strategy**. Around these two questions, we further formalize the three low-coupling advantages and the drawback of random masking performance plug-in to 7 constraints. For the drawback, we distill two key constraints: masking decisions must be influenced by the target model and must be influenced by the target task. For the first question, based on token-wise decomposition (Oh and Schuler, 2023) and the generative characteristics of LLMs with causal attention, we construct a first-order Markov process with *information-fusion–hidden-state alternation*, thereby converging the masking targets to the last-layer attention heads, and we empirically validate the feasibility of this target via random masking performance plug-in.

For the second question, we model masking strategy as *masking target combination*, and adopt a two-stage strategy of candidate combination construction + combination selection to avoid expensive search over an exponential combination space. In the first stage, under the controlled condition of target task×target model, we propose a candidate combination construction method based on a three-axis mapping, yielding a small set of direction-complementary and interpretable combinations. In the second stage, we propose an axis-variance minimization combination selection method that selects the most compact combination across all axes and is less sensitive to outlier targets, thereby improving the stability of performance gains in a statistical sense. We conduct extensive experiments on PibE-MPP. The results show its effectiveness and generality, thereby demonstrating its transferability across multiple models and tasks.

Our main contributions are as follows.

- We pose two core questions and provide 7 constraints, thereby retaining the three advantages while overcoming a drawback.
- We treat all attention heads in the last layer as masking targets by designing a first-order Markov process with alternating hidden state and information fusion, and empirically validate the feasibility of this target.
- We propose candidate combination construction method based on three-axis mapping and combination selection method based on axis-variance minimization, to improve gain stability while satisfying all constraints.
- Extensive evaluation on 6 LLMs and 24 datasets confirms PibE-MPP's effectiveness, stable gains, and domain performance, and includes an ablation demonstrating the necessity of the final module.

2 Related Work

2.1 Large language models

Common types of LLMs include general-purpose LLMs and domain-specific LLMs. General-purpose LLMs are pretrained on large-scale, cross-domain corpora to achieve strong cross-task generalization (Yang et al., 2024a; Team, 2024). Domain-specific LLMs are usually initialized from a general-purpose LLM and further pretrained on domain corpora to acquire specialized capabilities (e.g., Qwen2.5-Math (Yang et al., 2024b)), often followed by post-training such as instruction tuning and preference optimization. Knowledge distillation is another common route that transfers capabilities via teacher-student training (e.g., DeepSeek-R1-Distill-Qwen-1.5B (DeepSeek-AI, 2025)). Overall, these training and transfer routes typically require corresponding training when the target model, domain, or setting changes. In contrast, PibE-MPP is a **training-free** performance plug-in that can be directly attached to existing models, making it complementary to rather than in conflict with existing training paradigms.

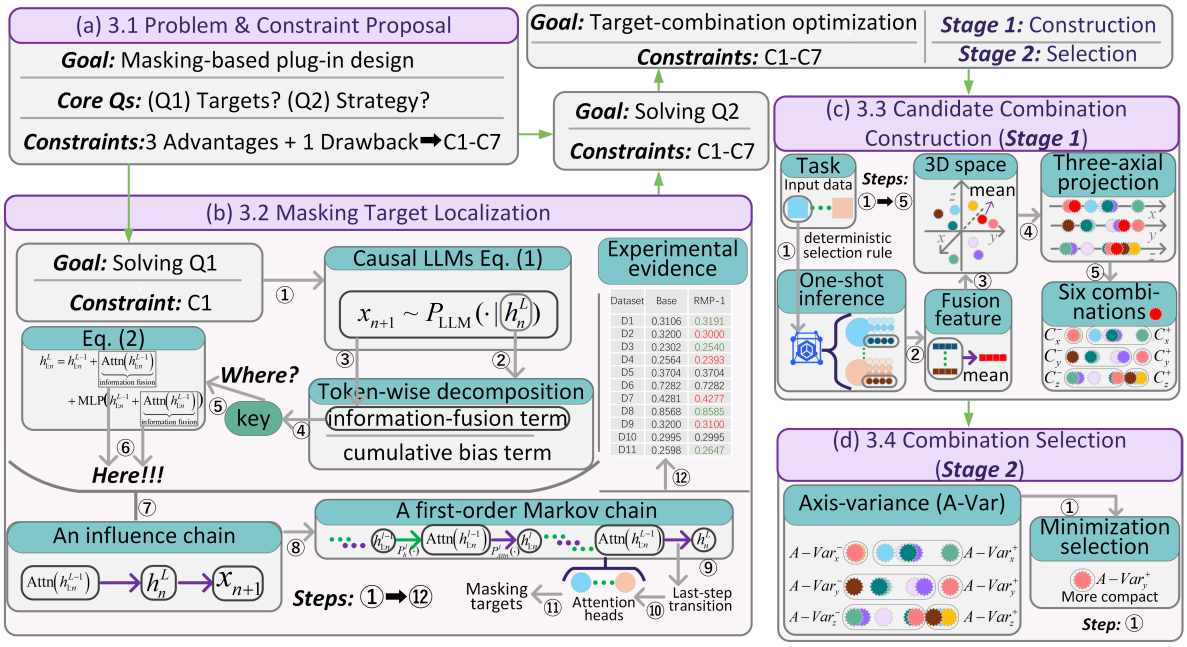


Figure 2: Overall framework. (a) The proposal of two core questions and their constraint set. (b) For core question (1): the localization logic of masking targets (all attention heads in the last layer), and experimental verification that masking targets can induce performance fluctuations, thereby demonstrating feasibility. (c) Stage 1 for solving core question (2): constructing diverse candidate combinations via many methods such as three-axial mapping. (d) Stage 2 for solving core question (2): selecting a more robust combination based on the variance criterion. Moreover, (c) and (d) jointly constitute the main usage pipeline of PibE-MPP.

2.2 LLMs pruning

LLMs pruning typically reduces inference cost by masking or removing a subset of parameters/structures. Its targets span weight-level masking (Frantar and Alistarh, 2023; Gu et al., 2025) and structured pruning (e.g., attention heads) (Xia et al., 2024; Hou et al., 2025; Yang et al., 2025; Guo et al., 2025). These pruning methods primarily aim at compression and speedup, treating performance degradation as a cost or constraint.

Although our PibE-MPP also performs *masking* on attention heads, it optimizes for **stable performance gains** rather than reducing parameters/compute. Therefore, a direct comparison between PibE-MPP and pruning methods can be potentially misleading due to objective mismatch. In addition, since the objective of random masking can be naturally defined as expected task performance improvement, we adopt it as a more aligned baseline and match its masking targets with PibE-MPP to ensure a fair comparison.

3 Method

We design the "**p**lay-**i**t-**b**y-**e**ar" **m**asking **p**erformance **p**lug-in (PibE-MPP). The overall framework of this paper is shown in Fig. 2, where the plug-in usage pipeline is illustrated in Fig. 2c and Fig. 2d.

3.1 Problem and constraint proposal

When we treat random masking as a performance plug-in, it offers low coupling to tasks (transferable across tasks), low coupling to models (applicable to different LLMs), and low coupling to training resources (reducing usage constraints). However, it suffers from a critical drawback. The performance gains are highly stochastic (as shown in Fig. 1).

For designing a masking performance plug-in for LLMs, we pose two core questions: **(1)** what are the masking targets (i.e., the *type* and the *scope* of targets); **(2)** what is the masking strategy (deciding which targets to mask and the selection mechanism). To ensure that the designed masking performance plug-in can inherit the three low-coupling advantages and address one severe drawback, we formalize them into the following constraints. We denote the constraints as **C1–C7** for brevity.

1. Advantage constraints:

Low coupling to the target model means that the method is weakly related to the specific structure of the target model (e.g., the LLaMA2 ar-

chitecture), ensuring generality across multiple model architectures.

C1: Do not rely on a specific model structure.

Low coupling to the target task means that applying the method to one task should not change its usability and effectiveness on other tasks.

C2: Parameters remain unchanged after use.

Low coupling to training resources means that avoiding large-scale training and data processing, which can be further divided into the following three constraints.

C3: Low coupling to the training dataset.

C4: Low coupling to the validation dataset.

C5: The additional computational overhead should be approximately the overhead of directly running inference on the target task (computational resources are constrained).

2. Drawback constraints:

Highly stochastic performance gains fundamentally arise because random masking decisions are not effectively constrained by the "given model and given task" condition. Therefore, we propose two constraints to explicitly introduce the "model/task" condition.

C6: Affected by the given model.

C7: Affected by the given task.

Notably, core question (1) is a general prerequisite for core question (2). Therefore, masking strategies are usually designed based on given targets. Furthermore, solving core question (1) determines the target type and scope under **C1**, while solving core question (2) needs to satisfy all the above constraints simultaneously.

3.2 Masking target localization

For the core question (1), we infer the type and scope of the masking targets from the perspective of the LLMs generation process. Specifically, based on causal-attention modeling and the transformer structure, we know that using only the hidden-layer information of the last token of the current input is sufficient to predict the next token.

Let the input be $X = \{x_1, \ldots, x_n\}$, the LLM has L layers, and the hidden state at layer l be h^l; then x_{n+1} can be expressed as:

$$\begin{aligned} & x_{n+1} = LLM(X) \\ \Rightarrow\ & x_{n+1} \sim P_{\mathrm{LLM}}(\cdot \mid x_{1:n}) \\ \Rightarrow\ & x_{n+1} \sim P_{\mathrm{LLM}}(\cdot \mid h_n^L) \end{aligned} \tag{1}$$

Based on Eq. (1), we know h_n^L is a key and further need to determine which masking targets directly influence h_n^L. To this end, based on token-wise decomposition (Oh and Schuler, 2023), the hidden state of the last token can be strictly decomposed into an information-fusion term (obtained by summing the additive contribution components of all tokens) and an accumulated bias term. The information-fusion term explicitly characterizes how information from each token is aggregated into the representation of the last token. Based on Eq. (1), the information-fusion term is a key for h_n^L.

Then, we can know that the update of all h at layer L in LLMs can be written as follows:

$$\begin{aligned} h_{1:n}^L = & h_{1:n}^{L-1} + \underbrace{\mathrm{Attn}\Big(h_{1:n}^{L-1}\Big)}_{\text{information fusion}} \\ & + \mathrm{MLP}\Big(h_{1:n}^{L-1} + \underbrace{\mathrm{Attn}\Big(h_{1:n}^{L-1}\Big)}_{\text{information fusion}}\Big) \end{aligned} \tag{2}$$

From Eq. (2), information fusion occurs only in the attention module. Therefore, the attention module is a key for h_n^L. Furthermore, with the layer index as the time step, we establish a first-order Markov chain with information-fusion (attention module)–hidden-state alternation:

$$\begin{aligned} & \textbf{hidden-state}\rightarrow\textbf{information-fusion:} \\ & \qquad Attn(h_{1:n}^{l-1}) \sim P_h^l\Big(\cdot \mid h_{1:n}^{l-1}\Big) \\ & \textbf{information-fusion}\rightarrow\textbf{hidden-state:} \\ & \qquad h_{1:n}^{l} \sim P_{Attn}^l\Big(\cdot \mid Attn(h_{1:n}^{l-1})\Big) \end{aligned} \tag{3}$$

where $l = 1, \ldots, L,$. Both $P_h^l(\cdot)$ and $P_{Attn}^l(\cdot)$ are deterministic transition kernels defined by the LLM. Therefore, Eq. (3) corresponds to a deterministic Markov process.

Based on the last-step transition property of a first-order Markov chain, the last-layer attention module can be viewed as a masking target directly related to the final predictive representation. Moreover, since information fusion happens in parallel across multiple heads and is aggregated via output projection, we further refine it to the last-layer attention heads as the selected masking targets. Furthermore, we apply random masking to the selected targets to verify that masking these targets can also induce performance fluctuations (e.g., the "Experimental evidence" in Fig. 2a), thereby demonstrating the feasibility of the chosen masking targets.

3.3 Candidate combination construction

For core question (2), we model "masking" as a target-combination optimization: under the premise of satisfying the 7 constraints, we seek target combinations that can bring stable performance improvements. Suppose the target model has N candidate targets (taking LLaMA-3.2-3B as an example, $N = 24$), then the size of the combination space can be expressed as follows:

$$\sum_{k=1}^{N} \binom{N}{k} = 2^N - 1 \tag{4}$$

Based on Eq. (4), the combination space size is $2^N - 1$, which grows exponentially with N. Direct exhaustive evaluation over the training/validation set would violate C1–C7. Therefore, we adopt a two-stage solution. We construct candidate combinations in this subsection, and design a selection method in the next subsection.

To satisfy C5–C7, we only run one forward inference of the target LLM using the one-shot sample of the target task (Brown et al., 2020; Min et al., 2022), thereby obtaining target activation states under the controlled condition of "target task $\times$ target model". For one-shot sample selection, we use the deterministic rule of "the first sample of the given task dataset" to ensure reproducibility.

In addition, a single target is computed as:

$$\mathrm{Attn}(h_{1:n}) = \mathrm{A}_{\mathrm{score}}(h_{1:n}\mathbf{W}_V) \tag{5}$$

where $\mathrm{A}_{\mathrm{score}} = \mathrm{softmax}(\frac{(h_{1:n}\mathbf{W}_Q)(h_{1:n}\mathbf{W}_K)^\top}{\sqrt{d_k}})$. W_Q, W_K, and W_V are trainable parameters.

From Eq. (5), the attention score matrix $\mathrm{A}_{\mathrm{score}}$ determines cross-token information routing and fusion strength. Furthermore, combining Eq. (1), we take the attention score distribution of the last token over all tokens as the target fusion feature.

Since the target fusion feature is usually high-dimensional and its dimensionality varies with the number of tokens in an input, we use 3D t-SNE (Guan et al., 2020; Li et al., 2021) to embed it into a three-dimensional space. This embedding preserves local neighborhood structure as much as possible, enables a unified analysis of variable-length high-dimensional features, and provides an intuitive visualization (e.g., the "3D space" view in Fig. 2c).

Furthermore, we introduce the commonly used "mean" concept in attention-head representation analysis (Kobyzev et al., 2025; Zhang et al., 2025; Jha and Reagen, 2025). Specifically, the mean point $\boldsymbol{\mu}$ in 3D space is obtained by embedding the arithmetic mean vector of all target fusion feature vectors under 3D t-SNE (e.g., the "Fusion feature" view and "3D space" view in Fig. 2c), providing comparability between the mean fusion feature and each target fusion feature under the same embedding mapping.

In 3D space, a simple and intuitive mapping is to project onto each axis, producing three sets of mapping results, each containing all targets and the mean target (e.g., the "Three-axial projection" in Fig. 2c). Then for each axis $r \in \{x, y, z\}$, using μ_r as the reference threshold, we construct within-axis combinations as follows:

$$\mathcal{C}_r^+ \triangleq \{\, i \mid z_{i,r} \geq \mu_r \,\}, \mathcal{C}_r^- \triangleq \{\, i \mid z_{i,r} < \mu_r \,\}. \tag{6}$$

Thus, we naturally obtain 6 candidate combinations $\{\mathcal{C}_x^+, \mathcal{C}_x^-, \mathcal{C}_y^+, \mathcal{C}_y^-, \mathcal{C}_z^+, \mathcal{C}_z^-\}$ (e.g., the "Six combinations" in Fig. 2c), ensuring directional diversity and intuitive interpretability of the candidate set.

Overall, the candidate combination construction method based on three-axis mapping satisfies the above 7 constraints.

3.4 Combination selection

For the candidate combinations, we need to solve how to select a combination that can bring stable performance gains. To this end, we propose a combination selection method based on axis-variance minimization. Specifically, the core motivation for introducing the variance criterion is that variance can characterize the dispersion of the candidate combination along an axis. When the variance of a combination along an axis is smaller, the selected targets are more "compact" in the embedding space. This makes the combination less likely to be dominated by individual outlier targets, reducing sensitivity to abnormal points. It also tends to exhibit more consistent behavior under different perturbations or stochastic factors and thus yields more stable performance gains.

Formally, the combination selection method based on axis-variance minimization can be expressed as follows:

$$\mathcal{C}_{r^\star}^{s^\star} \triangleq \arg\min_{\mathcal{C}_r^s} \frac{1}{|\mathcal{C}_r^s|} \sum_{i \in \mathcal{C}_r^s} \left(z_{i,r} - \bar{z}_{\mathcal{C}_r^s} \right)^2 \tag{7}$$

where $r \in \{x, y, z\}$ and $s \in \{+, -\}$, and $\bar{z}_{\mathcal{C}_r^s} \triangleq \frac{1}{|\mathcal{C}_r^s|} \sum_{i \in \mathcal{C}_r^s} z_{i,r}$.

Overall, this combination selection method is a purely statistical criterion and requires no additional training or validation, so it can avoid the resource overhead brought by extra inference on the validation dataset and reduce performance instability caused by distribution differences between the validation and test datasets. Meanwhile, the method is parameter-free and does not depend on specific LLMs. Therefore, the method satisfies C1–C5.

So far, core questions (1) and (2) have both been solved, and the solving process satisfies the above 7 constraints. Therefore, we can treat combination construction and combination selection as a performance plug-in. Further, we can observe a unique property of this plug-in. Under a fixed target model, when facing different target tasks, this plug-in will not directly reuse a fixed set of targets, but instead adaptively generates the corresponding masking target combination according to the specific target task. Therefore, we refer to this plug-in as the "**p**lay-**i**t-**b**y-**e**ar" **m**asking **p**erformance **p**lug-in (PibE-MPP).

4 Experiment

We conduct experiments to evaluate the effectiveness, stability, and generality of PibE-MPP across various LLMs and tasks. Our study is guided by the following four research questions (RQs).

RQ1: Effectiveness and generality? Can PibE-MPP improve the performance of various general-purpose LLMs on multiple tasks?

RQ2: Stability? Compared with random masking performance plug-in, can PibE-MPP obtain stable performance gains?

RQ3: Domain performance? Can PibE-MPP enable domain-specific LLMs to achieve specialized improvements on domain tasks?

RQ4: Necessity? Is the final module combination selection in PibE-MPP necessary?

4.1 Setting

Model: First, we select LLaMA and Qwen families. For the former, we select LLaMA-3.2-3B and LLaMA-3.2-1B (Liu et al., 2025). For the latter, we select Qwen2.5-1.5B (Team, 2024; Yang et al., 2024a) and DeepSeek-R1-Distill-Qwen-1.5B (DeepSeek-AI, 2025). DeepSeek-R1-Distill-Qwen-1.5B is a distilled model. These LLMs are used to verify the effectiveness and generality of PibE-MPP across multiple architectures and multiple learning paradigms of LLMs. Second, we select Qwen2.5-Coder-3B (Hui et al., 2024; Yang et al., 2024a) and Qwen2.5-Math-1.5B (Yang et al., 2024b) to verify whether PibE-MPP can achieve specialized improvements for domain-specific LLMs on domain tasks.

Dataset: In our experiments, we use a total of 24 datasets. Among them, 11 datasets are used to evaluate general-purpose LLMs, covering multiple domains such as mathematics and general knowledge. In addition, we use 10 datasets to evaluate a math LLM and 3 datasets to evaluate a code LLM. For brevity, we abbreviate dataset names in the main text. The full names and corresponding citations are provided in the appendix.

Baseline: For convenience, we simplify the experimental notation. Base denotes the underlying LLM without any plug-ins (general or domain-specific LLM). PibE-MPP and RMPP denote Base equipped with PibE-MPP and the random-masking plug-in, respectively. As discussed in Section 2, our main baseline is Base, and we additionally include RMPP, which masks only the last-layer attention heads.

Other setting: All masking is implemented by zeroing the attention-score matrix. We report single-run results since most benchmarks are multiple-choice and log-likelihood-based evaluation is deterministic. We use lm-evaluation-harness (Gao et al., 2024). Additional settings/analyses (fixed t-SNE hyperparameters, RMPP vs. PibE-MPP performance bounds comparison, and PibE-MPP low coupling) are provided in the appendix.

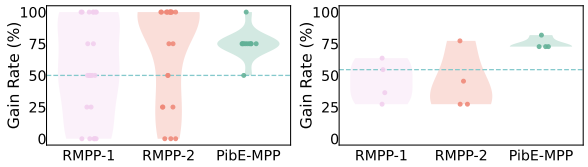
4.2 Effectiveness and generality (RQ1)

This section evaluates four general-purpose LLMs (including one distilled variant) from two model families on 11 datasets, using two baselines, where RMPP is run twice (RMPP-1/2). The results are summarized in Table 1.

From Table 1, compared with Base and RMPP-1/2, PibE-MPP shows an upward signal in most cells, indicating relatively consistent effectiveness and generality. Under fixed-task conditions, PibE-MPP outperforms Base and at least one RMPP on most datasets, demonstrating cross-task robustness. For example, on `xwinograd_zh`, it is no worse than Base and RMPP-1/2 across all four LLMs. Under fixed-model conditions, the "Sum-

	gaokao history	history	formal logic	gaokao mathqa	school_psych-ology_light	xwinograd zh	xnli bg	xwinograd en	gaokao physics	gaokao chemistry	high_school us_history
LLaMA-3.2-3B											
Base	31.06	32.00	23.02	25.64	37.04	72.82	42.81	85.68	32.00	29.95	25.98
RMPP-1	31.91	30.00	25.40	23.93	37.04	72.82	42.77	85.85	31.00	29.95	26.47
RMPP-2	28.94	32.00	23.02	23.93	35.19	71.43	44.10	85.85	28.50	30.43	25.98
PibE-MPP	31.91 ↑=↑	31.00 ↓↑↓	26.98 ↑↑↑	24.50 ↓↑↑	38.89 ↑↑↑	73.41 ↑↑↑	43.05 ↑↑↓	85.38 ↓↓↓	34.00 ↑↑↑	30.92 ↑↑↑	26.47 ↑=↑
Summary	vs Base (W,E,L)=(8,0,3), dmax=+3.96; vs RMPP-1 (8,2,1), dmax=+3.00; vs RMPP-2 (8,0,3), dmax=+5.50										
LLaMA-3.2-1B											
Base	22.13	14.00	23.02	29.63	27.78	64.48	39.36	81.20	26.50	28.99	25.49
RMPP-1	25.53	17.00	14.29	25.93	25.93	64.29	39.84	80.86	24.00	26.57	25.49
RMPP-2	25.53	17.00	16.67	24.79	27.78	63.29	38.63	81.42	23.50	25.12	25.98
PibE-MPP	22.98 ↑↓↓	18.00 ↑↑↑	20.63 ↓↑↑	30.48 ↑↑↑	27.78 =↑=	64.48 =↑↑	38.71 ↓↓↑	81.72 ↑↑↑	29.00 ↑↑↑	27.05 ↓↑↑	25.49 ==↓
Summary	vs Base (5,3,3), dmax=+4.00; vs RMPP-1 (8,1,2), dmax=+6.34; vs RMPP-2 (8,1,2), dmax=+5.69										
Qwen2.5-1.5B											
Base	66.38	20.00	26.98	33.62	42.59	75.20	36.35	81.63	42.50	35.75	24.02
RMPP-1	60.00	21.00	25.40	34.19	42.59	74.01	36.71	81.72	40.50	32.85	24.51
RMPP-2	62.98	22.00	24.60	32.76	38.89	75.00	36.67	82.32	40.50	29.47	23.53
PibE-MPP	64.68 ↓↑↑	22.00 ↑↑=	27.78 ↑↑↑	36.47 ↑↑↑	44.44 ↑↑↑	75.79 ↑↑↑	36.91 ↑↑↑	81.72 ↑=↓	41.50 ↓↑↑	38.16 ↑↑↑	23.53 ↓↓=
Summary	vs Base (8,0,3), dmax=+2.85; vs RMPP-1 (9,1,1), dmax=+5.31; vs RMPP-2 (8,2,1), dmax=+8.69										
DeepSeek-R1-Distill-Qwen-1.5B											
Base	26.38	20.00	26.98	27.92	22.22	59.72	33.57	64.43	30.50	29.47	27.45
RMPP-1	25.96	17.00	26.98	28.49	16.67	59.52	35.02	64.30	27.50	24.15	25.49
RMPP-2	27.23	15.00	26.19	25.93	20.37	58.33	34.62	64.90	26.00	25.12	26.47
PibE-MPP	30.21 ↑↑↑	20.00 =↑↑	26.19 ↓↓=	28.21 ↑↓↑	20.37 ↓↑=	59.72 =↑↑	34.34 ↑↓↓	64.69 ↑↑↓	31.00 ↑↑↑	30.43 ↑↑↑	27.94 ↑↑↑
Summary	vs Base (7,2,2), dmax=+3.83; vs RMPP-1 (8,0,3), dmax=+6.28; vs RMPP-2 (7,2,2), dmax=+5.31										

Table 1: Results (accuracy, %). For each PibE-MPP entry, we append three tiny markers (↑/↓/=) to indicate win/loss/tie against [Base, RMPP-1, RMPP-2]. Summary reports (win, equal, loss) counts, e.g., (8,0,3), of PibE-MPP over each baseline across 11 datasets, and dmax as the maximum improvement (percentage points).



(a) Fixed-dataset view (b) Fixed-model view

Figure 3: Gain rate (%) distribution comparison. The dashed line marks the reference: in (a) fixed-dataset, over half of the models achieve non-negative gains; in (b) fixed-model, over half of the datasets achieve non-negative gains.

mary" further shows that it has a relatively high win ratio and relatively high gains against each baseline. For example, the win counts against Base are 8/5/8/7, and the "non-negative" (win+tie) counts reach 8/8/8/9, respectively, where the maximum improvements are +3.96/+4.00/+2.85/+3.83.

Overall, these results jointly verify the effectiveness and cross-task/cross-model generality of PibE-MPP, and indicate that it has low coupling to tasks and LLMs, enabling painless transfer across multiple LLMs and multi-domain tasks.

4.3 Gain stability (RQ2)

This section focuses on gain stability, comparing the gain fluctuations of RMPP and PibE-MPP from two perspectives, "fixed-model" and "fixed-task", and verifying the stability advantage of PibE-MPP (ours) through more intuitive visualization results.

Under the view of a fixed task dataset D_i, we define the "gain rate" as follows:

$$\mathrm{GR}(D_i) = \frac{1}{|M|} \sum_{j=1}^{|M|} \mathbb{I}\big(P_{i,j}^{\text{ours}} - P_{i,j}^{\text{Base}} \geq 0\big) \quad (8)$$

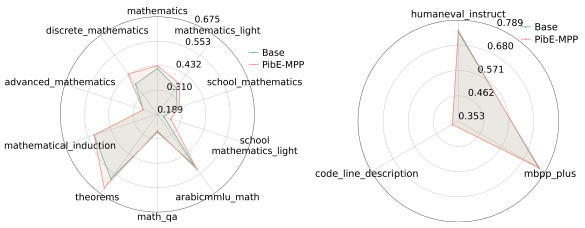
where M is the set of LLMs, and $|M|$ is the number of LLMs. $P_{i,j}^{\text{ours}}$ and $P_{i,j}^{\text{base}}$ denote the performance metric (e.g., accuracy) of the j-th LLM on task D_i with PibE-MPP (PibE-MPP) and Base, respectively. $\mathbb{I}(\cdot)$ is the indicator function.

Under the view of a fixed model M_j, we define the "gain rate" as:

$$\mathrm{GR}(M_j) = \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \mathbb{I}\big(P_{i,j}^{\text{ours}} - P_{i,j}^{\text{base}} \geq 0\big) \quad (9)$$

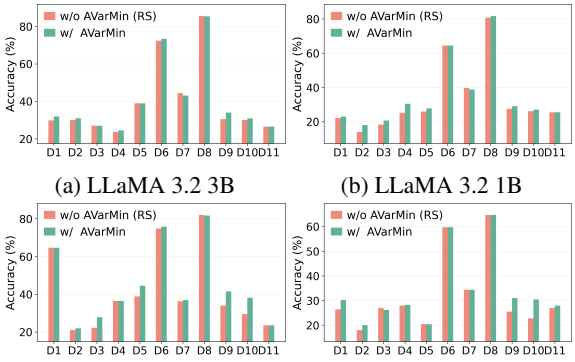
Based on Eq. (8) and Eq. (9), the stability differences between RMPP-1/2 and PibE-MPP can be observed in the gain rate distributions shown in Fig. 3.

Under the fixed-dataset view (Fig. 3a), the distributions of RMPP-1/2 are more dispersed and the distribution shapes of the two random trials are inconsistent, indicating that its cross-dataset gains are unstable and sensitive to random seeds; meanwhile, more values fall near or below the refer-



(a) Qwen2.5-Math-1.5B (b) Qwen2.5-Coder-3B

Figure 4: Performance on domain-specific LLMs.



(a) LLaMA 3.2 3B (b) LLaMA 3.2 1B

(c) Qwen2.5 1.5B (d) DeepSeek-R1-Distill-Qwen 1.5B

Figure 5: Ablation results of the final combination selection module (AVarMin). `w/ AVarMin` denotes using axis-variance minimization for combination selection; `w/o AVarMin (RS)` denotes removing this module and replacing it with random selection.

ence line, showing that it is difficult to cover non-negative gains for more models on more datasets. In contrast, the distribution of PibE-MPP is more concentrated and overall higher, with more values above the reference line, indicating that it can produce non-negative gains for more models on most datasets, and thus has better cross-dataset stability and overall gain level. Under the fixed-model view (Fig. 3b), RMPP-1/2 still exhibit large fluctuations and an overall lower level, indicating unstable cross-model gains and insufficient overall gains; whereas PibE-MPP still maintains higher concentration and overall level, indicating that it can also stably cover more datasets and obtain non-negative gains across different models.

Overall, PibE-MPP effectively alleviates the high randomness of performance gains, and better meets the stability and transferability requirements of performance plug-ins than RMPP.

4.4 Domain performance (RQ3)

We evaluate the impact of PibE-MPP on the domain performance of domain-specific LLMs. The results are shown in Fig. 4.

For the Qwen2.5-Math-1.5B (Fig. 4a), PibE-MPP expands outward or remains comparable on most tasks, showing a more consistent within-domain gain signal, reflecting further strengthening of math expertise. For the Qwen2.5-Coder-3B (Fig. 4b), the two curves are overall close, and are comparable or slightly outward on some tasks, indicating that this plug-in does not systematically weaken existing code expertise, and can still bring improvements on some tasks. Overall, PibE-MPP has the capability to enable domain LLMs to achieve specialized improvements (at least without systematic degradation).

4.5 Necessity (RQ4)

We only ablate the last module axis-variance minimization (`AVarMin`) of the PibE-MPP pipeline and replace it with random selection (RS), while keeping the rest of the pipeline and evaluation settings unchanged to ensure attributable comparisons. The results are shown in Fig. 5.

As shown in Fig. 5, across the four models, `w/ AVarMin` is overall better than (or no worse than) `w/o AVarMin (RS)` and shows clear advantages on multiple datasets; in contrast, random replacement often leads to accuracy drops of varying degrees and larger fluctuations across datasets, indicating that it relies more on "accidentally hitting" a better combination. This verifies the necessity of the last module for the PibE-MPP pipeline.

5 Conclusion

Inspired by random masking, we propose a lightweight performance plug-in, PibE-MPP, which inherits the three coupling advantages of random masking performance plug-in and addresses its severe drawback of highly stochastic gains. PibE-MPP enables given LLMs to adaptively generate a set of masking targets for a given dataset, and obtain stable performance improvements through a simple masking operation on the given LLMs, which has been verified in extensive experiments. PibE-MPP's transferability across multiple tasks and models is validated.

Limitations

We treat random masking, a classic baseline method, as a performance plug-in and find that it has three low-coupling advantages, but also suffers from a severe drawback that its performance gains are highly stochastic. To this end, we pro-

pose PibE-MPP, a "play-it-by-ear" masking performance plug-in, which inherits the three low-coupling advantages and also effectively mitigates the severe drawback. However, PibE-MPP still has the following limitations. To ensure reproducibility and stability, we adopt a one-shot deterministic selection rule and use 3D t-SNE with default parameters. The main goal of this paper is to design a lightweight performance plug-in that inherits the three low-coupling advantages and alleviates the highly stochastic gains, thus we do not further systematically study the sensitivity and correlation between this selection rule/hyperparameter settings and the performance of PibE-MPP. To the best of our knowledge, existing work rarely designs lightweight performance plug-ins systematically from the perspective of three low-coupling advantages + one severe drawback. This makes our work an early exploration of this research direction. Therefore, we do not pursue the theoretical optimum upper bound in the two steps of candidate combination construction and final combination selection, so these two modules still have room for further optimization.

Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities and the Research Funds of Renmin University of China 24XNKJ31, and National Social Science Foundation of China 62072463. Xun Liang is the corresponding author of this paper.

References

Saleh Ashkboos, Maximilian L. Croci, Marcelo Gennari Do Nascimento, and et al. 2024. Slicegpt: Compress large language models by deleting rows and columns. In *International Conference on Learning Representations (ICLR)*.

Jacob Austin, Augustus Odena, Maxwell Nye, and et al. Program synthesis with large language models.

Tom B. Brown, Benjamin Mann, Nick Ryder, and et al. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Mark Chen, Jerry Tworek, Heewoo Jun, and et al. 2021. Evaluating large language models trained on code.

Alexis Conneau, Ruty Rinott, Guillaume Lample, and et al. 2018. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.

Elias Frantar and Dan Alistarh. 2023. Sparsegpt: Massive language models can be accurately pruned in one-shot.

Leo Gao, Jonathan Tow, Baber Abbasi, and et al. 2024. The language model evaluation harness.

Tianteng Gu, Bei Liu, Bo Xiao, and et al. 2025. Denoiserotator: Enhance pruning robustness for llms via importance concentration. *arXiv preprint arXiv:2505.23049*.

Yue Guan, Jingwen Leng, Chao Li, and et al. 2020. How far does bert look at: Distance-based clustering and analysis of bert's attention. In *Proceedings of COLING*.

Jialong Guo, Xinghao Chen, Yehui Tang, and et al. 2025. Slimllm: Accurate structured pruning for large language models.

Jiujun He and Huazhen Lin. 2025. Olica: Efficient structured pruning of large language models without retraining. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 22580–22594. PMLR.

Dan Hendrycks, Collin Burns, Steven Basart, and et al. 2021. Measuring massive multitask language understanding. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Bairu Hou, Qibin Chen, Jianyu Wang, and et al. 2025. Instruction-following pruning for large language models.

Huang Huang, Fei Yu, Jianqing Zhu, and et al. 2024. Acegpt, localizing large language models in arabic. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8139–8163. Association for Computational Linguistics.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, and et al. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *arXiv preprint arXiv:2305.08322*.

Binyuan Hui, Jian Yang, Zeyu Cui, and et al. 2024. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*.

Nandan Kumar Jha and Brandon Reagen. 2025. Entropy-guided attention for private llms.

Seil Kang, Jinyeong Kim, Junhyeok Kim, and et al. 2025. See what you are told: Visual attention sink in large multimodal models. In *International Conference on Learning Representations (ICLR)*.

Ivan Kobyzev, Abbas Ghaddar, Dingtao Hu, and et al. 2025. Integral transformer: Denoising attention, not too much not too little. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*.

Fajri Koto, Haonan Li, Sara Shatnawi, and et al. 2024. Arabicmmlu: Assessing massive multitask language understanding in arabic. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 5622–5640. Association for Computational Linguistics.

Qi Le, Enmao Diao, Ziyan Wang, and et al. 2025. Probe pruning: Accelerating llms through dynamic pruning via model-probing.

Jian Li, Xing Wang, Zhaopeng Tu, and et al. 2021. On the diversity of multi-head attention. *Neurocomputing*, 454:14–24.

Zechun Liu, Changsheng Zhao, Igor Fedorov, and et al. 2025. Spinquant: Llm quantization with learned rotations.

Sewon Min, Xinxi Lyu, Ari Holtzman, and et al. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, and et al. 2023. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 15991–16111. Association for Computational Linguistics.

Byung-Doh Oh and William Schuler. 2023. Token-wise decomposition of autoregressive language model hidden states for analyzing model predictions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10105–10117, Toronto, Canada. Association for Computational Linguistics.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, and et al. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Trans. Mach. Learn. Res.*, 2023.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Alexey Tikhonov and Max Ryabinin. 2021. It's all in the heads: Using attention heads as a baseline for cross-lingual transfer in commonsense reasoning. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, Findings of ACL, pages 3534–3546. Association for Computational Linguistics.

Xuehao Wang, Liyuan Wang, Binghuai Lin, and et al. 2025a. Headmap: Locating and enhancing knowledge circuits in llms. In *International Conference on Learning Representations (ICLR)*.

Yuxin Wang, MingHua Ma, Zekun Wang, and et al. 2025b. CFSP: An efficient structured pruning framework for LLMs with coarse-to-fine activation information. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 9311–9328. Association for Computational Linguistics.

Wenhao Wu, Yizhong Wang, Guangxuan Xiao, and et al. 2025. Retrieval head mechanistically explains long-context factuality. In *International Conference on Learning Representations (ICLR)*.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and et al. 2024. Sheared llama: Accelerating language model pre-training via structured pruning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

An Yang, Baosong Yang, Binyuan Hui, and et al. 2024a. Qwen2 technical report.

An Yang, Beichen Zhang, Binyuan Hui, and et al. 2024b. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement.

Mingzhe Yang, Sihao Lin, Li Changlin, and et al. 2025. Let LLM tell what to prune and how much to prune. In *International Conference on Machine Learning (ICML)*.

Yunzhi Yao, Ningyu Zhang, Zekun Xi, and et al. 2024. Knowledge circuits in pretrained transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Arda Yüksel, Abdullatif Köksal, Lütfi Kerem Senel, and et al. 2024. Turkishmmlu: Measuring massive multitask language understanding in turkish. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 7035–7055. Association for Computational Linguistics.

Jue Zhang, Qingwei Lin, Saravan Rajmohan, and et al. 2025. From reasoning to answer: Empirical, attention-based and mechanistic insights into distilled DeepSeek r1 models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, and et al. 2024. Agieval: A human-centric benchmark for evaluating foundation models. In *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 2299–2314. Association for Computational Linguistics.

A Dataset Name and Citation

Due to the need to keep the description and representation in the main text concise and intuitive, we simplify the names of all the datasets used to different degrees, and the correspondence between their full names and abbreviations is shown in Table 2. We list the citations for each dataset in the table. All datasets can be used through the lm-evaluation-harness (Gao et al., 2024).

B Supplementary Details of Default Settings and Experimental Analysis

We use t-SNE from `sklearn.manifold` with the default settings: `n_components=3`, `perplexity=5`, and `random_state=42`. Since the core objective of this paper is to design the overall PibE-MPP to inherit three low-coupling advantages and effectively solve one severe defect, we did not conduct additional research exploration on this parameter setting. However, in all experiments we defaulted to this set of parameters, and the experiments cover multiple models and multiple datasets, and the good robustness of the experimental results also indirectly indicates that this parameter as a basic default setting is feasible.

In the practical implementation of the deterministic selection rule for the one-shot example, we follow the lm-evaluation-harness evaluation pipeline and use the first encountered sample as the fixed one-shot example. Notably, this sample does not necessarily align with the first sample in the original dataset. Similar to the t-SNE parameter settings, we apply this deterministic selection method consistently across all experiments. The overall results show that PibE-MPP maintains stable performance gains under this setting. Furthermore, extensive experiments validate the reasonableness of this deterministic rule and demonstrate the robustness of the method to the choice of the one-shot example. As a result, the generalizability and reproducibility of PibE-MPP in the one-shot setting can be ensured.

All experiments were conducted on an RTX 3080 GPU and an Intel(R) Core(TM) i7-10700K CPU. The software environment includes Python 3.9.20, PyTorch with CUDA 12.1, Transformers 4.47.0, and lm-evaluation-harness 0.4.9.

C Theoretical Performance Bound Comparison Between PibE-MPP and RMPP

Essentially, both PibE-MPP and RMPP select a masking target combination under masking operations. The search scope of RMPP is the complete search space constructed by all masking targets. In this paper, all masking targets are all attention heads in the last layer, thus the size of the combination space can be given by Eq. (4), i.e., there are $2^N - 1$ non-empty combinations in total, growing exponentially. In contrast, PibE-MPP can be regarded as first constructing candidate combinations from this exponential space to achieve search-space compression, and then performing combination selection on the candidate set. Therefore, the finally selected combination is a subset of the candidate combinations, and the candidate set itself is a subset of the original search space (i.e., the search space of RMPP).

Therefore, if we continuously perform RMPP, there is theoretically a possibility of hitting the final combination selected by PibE-MPP, and its hit probability can be written as follows:

$$p_{\text{hit}} \approx \frac{1}{2^N - 1} \tag{10}$$

In theory, the performance upper bounds for RMPP and PibE-MPP satisfy:

$$\text{Perf}^{\text{sup}}_{\text{RMPP}} \geq \text{Perf}^{\text{sup}}_{\text{PibE-MPP}} \tag{11}$$

In theory, the performance lower bounds for RMPP and PibE-MPP satisfy:

$$\text{Perf}^{\text{inf}}_{\text{RMPP}} \leq \text{Perf}^{\text{inf}}_{\text{PibE-MPP}} \tag{12}$$

On the other hand, if we further assume that each point in the search space has a one-to-one correspondence with a performance value, then the performance fluctuation range of RMPP covers the entire search space, which is also why RMPP produces highly stochastic performance gains, while the result of PibE-MPP is deterministic, thus it will not produce performance fluctuations due to different execution times.

D Low-Coupling Analysis of PibE-MPP

Among the three low-coupling design constraints (task-, model-, and compute-resource coupling), the first two have been evidenced in the Method

Full name	Abbr. 1	Abbr. 2	Citation
agieval_gaokao_history	gaokao_history	D1	(Zhong et al., 2024)
turkishmmlu_history	history	D2	(Yüksel et al., 2024)
arabic_leaderboard_arabic_mmlu_formal_logic	formal_logic	D3	(Hendrycks et al., 2021; Huang et al., 2024)
agieval_gaokao_mathqa	gaokao_mathqa	D4	(Zhong et al., 2024)
arabic_leaderboard_arabic_mmlu_high_school_psychology_light	school_psychology_light	D5	(Hendrycks et al., 2021; Huang et al., 2024)
xwinograd_zh	xwinograd_zh	D6	(Muennighoff et al., 2023; Tikhonov and Ryabinin, 2021)
xnli_bg	xnli_bg	D7	(Conneau et al., 2018)
xwinograd_en	xwinograd_en	D8	(Muennighoff et al., 2023; Tikhonov and Ryabinin, 2021)
agieval_gaokao_physics	gaokao_physics	D9	(Zhong et al., 2024)
agieval_gaokao_chemistry	gaokao_chemistry	D10	(Zhong et al., 2024)
arabic_leaderboard_arabic_mmlu_high_school_us_history	high_school_us_history	D11	(Hendrycks et al., 2021; Huang et al., 2024)
arabic_leaderboard_arabic_mmlu_elementary_mathematics	mathematics	–	(Hendrycks et al., 2021; Huang et al., 2024)
arabic_leaderboard_arabic_mmlu_elementary_mathematics_light	mathematics_light	–	(Hendrycks et al., 2021; Huang et al., 2024)
arabic_leaderboard_arabic_mmlu_high_school_mathematics	school_mathematics	–	(Hendrycks et al., 2021; Huang et al., 2024)
arabic_leaderboard_arabic_mmlu_high_school_mathematics_light	school_mathematics_light	–	(Hendrycks et al., 2021; Huang et al., 2024)
arabicmmlu_math_primary_school	arabicmmlu_math	–	(Koto et al., 2024)
bigbench_elementary_math_qa_multiple_choice	math_qa	–	(Srivastava et al., 2023)
bigbench_identify_math_theorems_multiple_choice	theorems	–	(Srivastava et al., 2023)
bigbench_mathematical_induction_multiple_choice	mathematical_induction	–	(Srivastava et al., 2023)
ceval-valid_advanced_mathematics	advanced_mathematics	–	(Huang et al.)
ceval-valid_discrete_mathematics	discrete_mathematics	–	(Huang et al.)
humaneval_instruct	humaneval_instruct	–	(Chen et al., 2021; Gao et al., 2024)
mbpp_plus	mbpp_plus	–	(Austin et al.; Gao et al., 2024)
bigbench_code_line_description_multiple_choice	code_line_description	–	(Srivastava et al., 2023)

Table 2: Full dataset names, abbreviations, and citations. We report two abbreviations: **Abbr. 1** and **Abbr. 2** are short names used in tables/figures (or – if unused).

and Experiments sections. Here we further analyze the coupling between PibE-MPP and compute resources. We consider three types of resources: training data, validation data, and hardware resources.

First, PibE-MPP does not rely on additional training or validation data. Therefore, it is weakly coupled to data resources.

Second, regarding hardware resources, the candidate-composition construction stage typically involves one-shot sample selection, a one-shot forward inference (with attention score matrices returned), feature extraction, t-SNE embedding, 3-axis mapping, and mean-based candidate construction, etc. Among these steps, the dominant cost is clearly the one-shot forward inference that returns attention score matrices, while the remaining steps are lightweight post-processing.

In the composition selection stage, we only compute the intra-axis variance for six candidate compositions and pick the one with the minimum variance. This is simple statistical computation with negligible overhead. During deployment, PibE-MPP mainly zeroes out the attention score matrices on the selected masking targets, which is also an operator-level lightweight computation.

Overall, the hardware resource consumption of PibE-MPP is upper-bounded by a one-shot forward inference with attention score matrices returned, indicating low coupling to compute resources. Compared with methods that require additional training, PibE-MPP is more advantageous in terms of hardware resource consumption.