

HiVe: Beyond Static Prompts for Multitask Learning via Hierarchy-based Vertical Mixture-of-Experts

Hyeonjik Bae Minyeol Kim Susik Yoon

Computer Science and Engineering

Korea University, Seoul, Korea

{bhg4060,minyeol1315,susik}@korea.ac.kr

Abstract

As large language models (LLMs) continue to scale, parameter-efficient fine-tuning (PEFT) has become a practical alternative to full-parameter adaptation. Prompt tuning is effective, but existing approaches either use flat prompt structures or hierarchical structures with fixed prompt composition, limiting adaptive prompt specialization. To address this limitation, we propose HiVe, a prompt tuning framework that models prompts at multiple levels and enables input-dependent specialization. HiVe constructs a prompt hierarchy by leveraging inter-task relationships during training, and employs a vertical mixture-of-experts (V-MoE) mechanism at inference time to compose prompts up to the level of specialization required for each input. Experiments show that HiVe consistently outperforms strong prompt tuning baselines across diverse tasks.

1 Introduction

The prohibitive cost of full fine-tuning large language models (LLMs) has motivated the development of parameter-efficient fine-tuning (PEFT) that updates a small subset of parameters (Wang et al., 2025). Specifically, prompt tuning (PT) learns soft prompts while keeping the backbone model frozen (Lester et al., 2021). The single shared prompt used in early PT approaches has proven ineffective in multi-task and multi-domain scenarios, thereby failing to capture diverse data distributions (Kim et al., 2024; Dun et al., 2025). Recent mixture-of-experts (MoE)-based PT approaches employ input-dependent prompt selection. For example, SMOp (Choi et al., 2023) sparsely routes inputs to short prompts, while PT-MoE (Li et al., 2025) decomposes prompts into low-rank components and combines them through routing.

However, the fundamental limitations of existing approaches lie in their *flat prompt space*, treating all prompts as experts at the same level. As illustrated

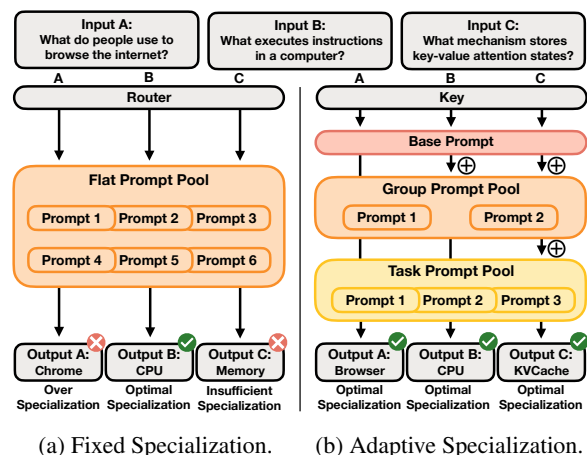


Figure 1: Fixed vs. adaptive prompt specialization. Unlike flat MoE approaches (left), HiVe dynamically constructs and uses the appropriate level of prompts (right).

in Figure 1, each input requires a diverse set of specializations to produce the correct output; some inputs benefit from general shared knowledge, while others require more task-specific information. Existing flat MoE-based approaches jointly encode both general and specialized knowledge in a few prompts within the same parameter space, making them inadequate for handling varying levels of prompt specialization across inputs.

Prior work has explored multi-level or hierarchical prompting for organizing shared and task-specific knowledge (Wang et al., 2022b; Chen et al., 2023; Liu et al., 2023), but typically applies a fixed task-dependent path or prompt composition, rather than selecting the hierarchy depth for each input. We identify that the core challenges of realizing adaptive specialization within a hierarchical prompt space are twofold: (i) automatically constructing the prompt hierarchy from learned task relationships at training time, and (ii) dynamically utilizing these disentangled prompts based on input characteristics at inference time. To our knowledge, no existing PT method explicitly addresses both requirements simultaneously.

To this end, we propose HiVe (Hierarchy-based Vertical Mixture-of-Experts)¹, a framework that treats a learned prompt hierarchy as a routing space for dynamically selecting the level of specialization required by each input. HiVe organizes prompts into a hierarchy of progressively specialized representations and employs an input-dependent, key-based prompt routing mechanism.

Specifically, at training time, HiVe constructs a three-level hierarchy consisting of base, group, and task prompts. However, constructing this hierarchy effectively is challenging, as incorrect grouping can introduce interference between tasks and degrade performance. To mitigate this, we induce a group-level hierarchy based on the stability of learned prompt relationships, allowing semantically related tasks to form stable groups during training. Furthermore, we adopt a residual prompt decomposition strategy that extracts shared information into higher-level prompts while preserving task-specific information at lower levels, thereby reducing both intra- and inter-level redundancy and interference.

At inference time, HiVe employs a key-based vertical MoE (V-MoE) mechanism that adaptively routes inputs across hierarchical specialization levels. Unlike conventional MoE approaches that route among parallel experts at the same abstraction level, V-MoE determines how deeply to traverse the prompt hierarchy for each input and progressively composes prompts down to the selected level. This enables the model to balance shared and task-specific knowledge according to the specialization required by each input.

We evaluate HiVe on Machine Reading for Question Answering (MRQA) and summarization benchmarks under both in-domain and out-of-domain settings. Experimental results demonstrate that our hierarchical, adaptive prompt specialization consistently improves both in-domain performance and out-of-domain generalization across diverse tasks over PT baselines. Remarkably, HiVe with a limited prompt parameter budget performs competitively with resource-heavy alternatives like full fine-tuning ($\sim 13,000\times$ params) and low-rank adaptation ($\sim 3.2\times$ params).

In brief, the primary highlights of HiVe are:

- **Emergent prompt hierarchy:** A data-driven prompt hierarchy is induced through stability-aware clustering, while residual decomposition

separates shared and task-specific representations across hierarchy levels, reducing redundancy and interference.

- **Vertical MoE routing:** A key-based hierarchical routing mechanism that dynamically determines the appropriate level of prompt specialization for each input by composing prompts down to the selected level.
- **Domain-robust generalization:** HiVe achieves the best results among PT methods, outperforming existing baselines in F1 on 12 MRQA datasets and ROUGE-L on 11 summarization datasets under both in- and out-of-domain settings, while maintaining performance parity with resource-heavy alternatives.

2 Related Work

2.1 MoE Prompt Tuning

MoE has been widely adopted to scale model capacity through sparse expert activation (Shazeer et al., 2017). Recent work has incorporated MoE into parameter-efficient fine-tuning (PEFT) to enable conditional computation over lightweight parameters. AdaMix (Wang et al., 2022a) and HydraLoRA (Tian et al., 2024) selectively activate adapter or LoRA experts (Hu et al., 2022). Prior methods such as ATTEMPT (Asai et al., 2022), SMOPT (Choi et al., 2023), and PT-MoE (Li et al., 2025) perform input-dependent prompt composition or routing. However, these approaches rely on flat prompt structures in which all prompts are treated at the same level of specialization, limiting their ability to adaptively model varying levels of specialization across inputs.

2.2 Task Grouping

Task relatedness plays a critical role in multi-task learning, particularly in NLP settings (Bingel and Søgaard, 2017). However, jointly learning loosely related tasks can result in interference and lead to negative transfer (Ni et al., 2023). Prior work has learned task groupings either by representing tasks as sparse combinations of shared latent components (Kumar and Daumé III, 2012) or by estimating gradient-based task affinity (Fifty et al., 2021). More recently, task vectors have been used to dynamically group source tasks based on their relevance to a target task in multi-task prompt tuning (Zhang et al., 2025). Building on these approaches, we construct hierarchical prompts based on task relationships learned during training.

¹Source code is available at <https://github.com/HyeonJikBae/HiVe>.

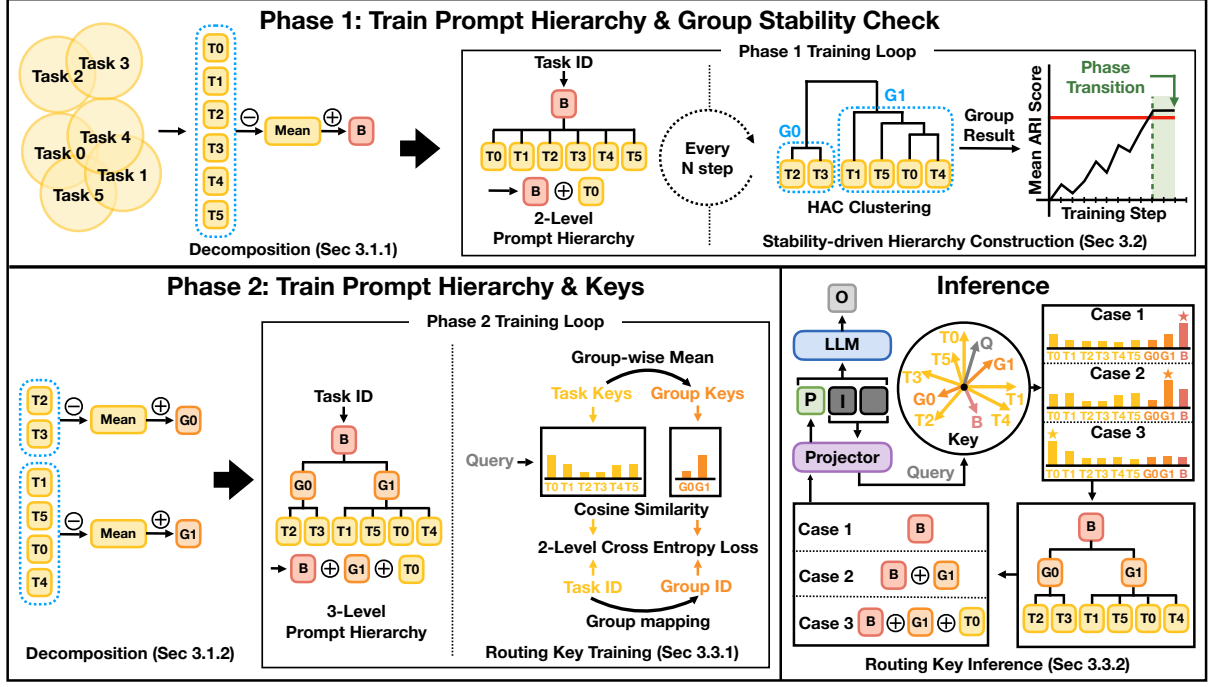


Figure 2: Overview of HiVe, consisting of two training phases followed by inference: (1) stability-driven hierarchy induction, (2) joint optimization of hierarchical prompts and routing keys, and (3) input-dependent hierarchical prompt selection through vertical mixture-of-experts (V-MoE) routing.

3 Methods

Figure 2 illustrates the overall framework of HiVe, rooted in a V-MoE mechanism with a *base-group-task* prompt hierarchy. At inference time, HiVe matches the input query against routing keys, selects the appropriate specialization level, and constructs the final prompts along the hierarchical path. At training time, after the initialization step described in Appendix C, HiVe operates in two training phases. In Phase 1, the model learns a base prompt and task prompts, forming a two-level hierarchy, and clusters tasks based on their task prompts. In Phase 2, group prompts are introduced to complete the three-level hierarchy, and routing keys are jointly optimized to direct each input query to the most appropriate prompt. Full training details are provided in Algorithm 1. The following sections introduce the three core components of HiVe—prompt decomposition, hierarchy construction, and V-MoE—in detail.

3.1 Residual Prompt Decomposition

We apply residual-based prompt decomposition before training to prevent information concentration in lower-level prompts, allowing higher-level prompts to retain general task-solving capability.

Algorithm 1 Training of HiVe

Require: Training data $\mathcal{X}$, template embeddings T , task embeddings E , low-rank dimension r
Ensure: Constructed prompt and key loss
1: $(P_B, P_T, W) \leftarrow \text{INITIALIZEPROMPT}(T, E, r)$
2: */* Sec. 3.1.1 */*
3: $(P_B, P_T) \leftarrow \text{DECOMPROMPT}(P_B, P_T)$
4: **for** each training step **do**
5: $\mathcal{L}_{key} \leftarrow 0$ and $(x, t) \sim \mathcal{X}$ *// x: input, t: task ID*
6: **if** Phase 1 **then**
7: $\tilde{P} \leftarrow \text{CONSTRUCTPROMPT}(P_B, P_T, W, t)$
8: */* Sec. 3.2 */*
9: **if** step mod $N = 0$ **then**
10: $c \leftarrow \text{UPDATEGROUPING}(P_T)$
11: $S \leftarrow \text{COMPUTEMEANARI}(c)$
12: **if** $\text{ISSTABLEGROUPING}(S)$ **then**
13: */* Sec. 3.1.2 */*
14: $(P_G, P_T) \leftarrow \text{DECOMPROMPT}(P_T, c)$
15: $\mathcal{K} \leftarrow \text{INITIALIZEKEY}(E, W)$
16: Phase1 $\rightarrow$ Phase2
17: **end if**
18: **end if**
19: **else**
20: */* Sec. 3.3.1 */*
21: $\mathcal{L}_{key} \leftarrow \text{TRAINKEY}(\mathcal{K}, t, c)$
22: $\tilde{P} \leftarrow \text{CONSTRUCTPROMPT}(P_B, P_G, P_T, W, t, c)$
23: **end if**
24: **end for**
25: **return** $\tilde{P}, \mathcal{L}_{key}$

3.1.1 Base-Task Decomposition

The initial prompt space consists of a base prompt $P_B \in \mathbb{R}^{l \times r}$ and task prompts $P_T = \{P_T^{(\tau)}\}_{\tau=1}^n$, where each task prompt $P_T^{(\tau)} \in \mathbb{R}^{l \times r}$ corresponds to an individual task. Here, n , l , and r denote the number of tasks, prompt length, and low-rank dimension used in the prompt space, respectively.

$P_T^{(\tau)}$ is initialized from task-specific data samples, inherently entangling representations shared across tasks with task-specific representations:

$$P_T^{(\tau)} = P_{sh} + P_{sp}^{(\tau)}, \quad (1)$$

where P_{sh} and $P_{sp}^{(\tau)}$ denote the shared and task-specific representations, respectively. To extract P_{sh} , we estimate it as the global mean of P_T :

$$P_{sh} = \frac{1}{n} \sum_{\tau=1}^n P_T^{(\tau)}. \quad (2)$$

Then, P_{sh} is transferred from P_T to P_B :

$$P_T^{(\tau)} \leftarrow P_T^{(\tau)} - P_{sh}, \quad P_B \leftarrow P_B + P_{sh}. \quad (3)$$

3.1.2 Group-Task Decomposition

Once clustering assignments c are finalized via stability-based clustering (Section 3.2), group prompts $P_G = \{P_G^{(g)}\}_{g=1}^m$ are constructed based on c , where each group prompt $P_G^{(g)} \in \mathbb{R}^{l \times r}$ and m denotes the number of groups.

To extract group-shared representations $P_{sh}^{(g)}$, we estimate them as the mean representation of task prompts in the same group:

$$P_{sh}^{(g)} = \frac{1}{|\mathcal{T}_g|} \sum_{\tau \in \mathcal{T}_g} P_T^{(\tau)}, \quad (4)$$

where $\mathcal{T}_g = \{\tau \mid c(\tau) = g\}$ denotes the set of task indices assigned to group g . Then, $P_{sh}^{(g)}$ is transferred from P_T to P_G :

$$P_T^{(\tau)} \leftarrow P_T^{(\tau)} - P_{sh}^{(c(\tau))}, \quad P_G^{(g)} \leftarrow P_{sh}^{(g)}. \quad (5)$$

Through these two decompositions, P_B , P_G , and P_T are encouraged to encode representations at their respective levels of granularity.

3.2 Stability-driven Hierarchy Construction

During Phase 1, prompts are trained as follows:

$$\tilde{P}_{\text{phase1}} = (P_B + P_T^{(t)})W, \quad (6)$$

where t denotes the ground-truth task ID and $W \in \mathbb{R}^{r \times d}$ denotes the projection matrix. As training progresses, P_T increasingly encodes task characteristics, serving as the basis for task grouping.

To measure task similarity, the distance matrix $\mathbf{D}$ is computed via cosine dissimilarity between mean-pooled task prompt vectors $\mathbf{v}_i$ and $\mathbf{v}_j$:

$$D_{i,j} = 1 - \frac{\mathbf{v}_i \cdot \mathbf{v}_j}{\|\mathbf{v}_i\| \|\mathbf{v}_j\|}. \quad (7)$$

Based on $\mathbf{D}$, tasks are grouped via Hierarchical Agglomerative Clustering (HAC), producing clustering assignment c_s at training step s :

$$c_s = \text{HAC}(\mathbf{D}). \quad (8)$$

The number of clusters is selected based on the silhouette score over cluster sizes from 2 to n . Singleton clusters are excluded from P_G construction.

Grouping too early may result in inaccurate clustering due to immature prompts, while grouping too late reduces training time for hierarchical learning. To finalize c at the appropriate stage, we evaluate clustering stability via an ARI-based metric (Hubert and Arabie, 1985). Every N steps, clustering is performed and c_s is compared against the previous K results to compute MeanARI_s:

$$\text{MeanARI}_s = \frac{1}{K} \sum_{k=1}^K \text{ARI}(c_s, c_{s-Nk}). \quad (9)$$

Once MeanARI_s exceeds the threshold η for R consecutive checks, c is finalized and Phase 2 begins.

3.3 Vertical Mixture-of-Experts (V-MoE)

V-MoE dynamically selects the appropriate level of prompt specialization within the prompt hierarchy for each input. Unlike conventional softmax-based MoE routers, which struggle to model hierarchical relationships, V-MoE employs a key-based routing mechanism that naturally encodes such relationships through centroid derivation.

3.3.1 Routing Key Training

In Phase 2, we introduce a routing key set $\mathcal{K} = \{k_B, k_G^{(g)}, k_T^{(\tau)}\}$, where each key lies in the same low-rank space $\mathbb{R}^r$ as the prompts. Task keys k_T are explicitly optimized, and higher-level keys are derived as centroids:

$$k_B = \frac{1}{n} \sum_{\tau=1}^n k_T^{(\tau)}, \quad k_G^{(g)} = \frac{1}{|\mathcal{T}_g|} \sum_{\tau \in \mathcal{T}_g} k_T^{(\tau)}. \quad (10)$$

This allows k_B and k_G to capture shared semantics while reducing bias toward individual task representations. The goal of key training is to attract each input query to its task key while repelling others, using derived group keys to preserve intra-group proximity. To this end, the mean-pooled input $x \in \mathbb{R}^d$ is projected via W to obtain query $q = xW^\top \in \mathbb{R}^r$, which is matched against the routing keys via cosine similarity:

$$s_T^{(\tau)} = \cos(q, k_T^{(\tau)}), \quad s_G^{(g)} = \cos(q, k_G^{(g)}). \quad (11)$$

The routing keys are optimized via a two-level cross-entropy loss with ground-truth supervision:

$$\mathcal{L}_{key} = \text{CE}(s_T, t) + \text{CE}(s_G, c(t)). \quad (12)$$

Independently, the prompt is constructed using the full ground-truth hierarchy:

$$\tilde{P}_{\text{phase2}} = (P_B + P_G^{(c(t))} + P_T^{(t)})W. \quad (13)$$

This decoupled training ensures that all hierarchy levels are jointly utilized during training, allowing each level to learn appropriate representations while preventing potential knowledge contamination from routing errors.

3.3.2 Routing Key Inference

Without requiring task ID, the model selects the routing key k^* with the highest cosine similarity to q among all keys in $\mathcal{K}$:

$$k^* = \arg \max_{k \in \mathcal{K}} \cos(q, k), \quad (14)$$

where k^* determines the hierarchical level, and the prompt is composed by aggregating prompts down to the selected level, ranging from P_B alone to the full prompt hierarchy $P_B + P_G^{(g)} + P_T^{(\tau)}$:

$$\tilde{P}_{\text{inf}} = \begin{cases} P_B, & k^* = k_B \\ P_B + P_G^{(g^*)}, & k^* = k_G^{(g^*)} \\ P_B + P_G^{(c(\tau^*))} + P_T^{(\tau^*)}, & k^* = k_T^{(\tau^*)} \end{cases} \quad (15)$$

After constructing a prompt at an appropriate specialization level, $\tilde{P}_{\text{inf}}$ is projected through W to obtain the final prompt representation $\tilde{P}_{\text{inf}}W$, which is prepended to the input sequence.

4 Experiments

4.1 Experimental Setup

Datasets and Tasks. HiVe is evaluated across question answering and summarization tasks. For question answering, we use the MRQA benchmark (Fisch et al., 2019), consisting of 6 in-domain and 6 out-of-domain datasets. For summarization, we construct a diverse benchmark composed of 6 in-domain and 5 out-of-domain datasets. For in-domain settings, models are trained on the training splits and periodically evaluated on the validation splits every few training steps. The checkpoint with the best validation performance is selected and evaluated directly on the out-of-domain datasets without further training. Detailed dataset descriptions are provided in Appendix A.

Baselines and Model. We compare HiVe with representative adaptation methods, including full fine-tuning (FT), parameter-efficient fine-tuning (PEFT) methods such as LoRA (Hu et al., 2022) and HydraLoRA (Tian et al., 2024), and prompt-based methods such as Prompt Tuning (PT) (Lester et al., 2021), DPT (Xiao et al., 2023), SMoP (Choi et al., 2023), and PT-MoE (Li et al., 2025). An adapted version of HiPro (Liu et al., 2023) is evaluated only in-domain, as it requires task IDs at inference. Baselines are selected to support both in- and out-of-domain evaluation under a unified input and backbone architecture. All methods use Llama-3.2-1B-Instruct as the backbone. For fair comparison, all prompt-based methods are configured to use a comparable number of trainable parameters (approximately 87K), while LoRA-based baselines are adjusted to use similar budgets whenever possible. Detailed hyperparameter settings are provided in Appendix D.2.

Evaluation Metrics. Evaluation follows the standard protocols for each benchmark. For question answering, we report F1 scores in the main results and provide Exact Match (EM) scores in Appendix E.1. For summarization, we report ROUGE-L scores. All results are averaged over three runs with different random seeds.

4.2 Experimental Results

4.2.1 Overall Performance

MRQA. Table 1 presents the main results on the MRQA benchmark. HiVe achieved the best performance among prompt-based methods in both in-domain and out-of-domain (OOD) settings. In the in-domain setting, it achieved an average F1 score of 75.4, compared with FT (75.2) while updating only approximately 0.007% of the model parameters. It also achieved the best performance across all in-domain tasks, improving the average F1 score over SMoP and PT-MoE by 2.1 and 2.6 points, respectively. Notably, HiVe also outperforms the hierarchical baseline HiPro-adapted by 1.0 F1 points. These gains persisted in OOD settings, indicating improved robustness under domain shifts. HiVe outperformed FT by 2.7 F1 points on average and achieved the best performance on four out of six OOD datasets. In addition, it improved the average F1 score over SMoP and PT-MoE by 1.4 and 2.2 points, respectively. These results suggest that HiVe effectively adapts prompt specialization levels across tasks and domains.

Method	# of Params	In-domain							Out-of-domain						
		SQ	News	Tri	Srch	HP	NQ	Avg	BSQ	DP	DRC	RC	RE	TB	Avg
FT	1.2B	87.3 _{0.9}	60.4 _{0.7}	71.7 _{1.1}	81.8 _{0.5}	75.7 _{0.9}	74.0 _{0.6}	75.2 _{0.2}	73.5 _{2.4}	48.6 _{0.8}	49.0 _{1.4}	47.0 _{0.8}	85.6 _{0.8}	56.8 _{2.6}	60.1 _{1.1}
LoRA	106K	87.3 _{0.4}	59.2 _{0.6}	74.8 _{0.3}	79.4 _{1.0}	74.4 _{0.5}	73.9 _{0.3}	74.9 _{0.3}	77.0 _{0.7}	47.5 _{1.5}	49.8 _{0.3}	50.4 _{0.4}	85.3 _{0.5}	60.0 _{0.9}	61.7 _{0.7}
HydraLoRA	278K	87.8 _{0.6}	59.0 _{0.5}	74.9 _{0.4}	79.8 _{1.5}	74.9 _{0.4}	74.3 _{0.2}	75.2 _{0.4}	76.6 _{0.5}	48.6 _{2.6}	49.7 _{0.3}	51.0 _{1.3}	85.3 _{0.7}	58.8 _{1.4}	61.7 _{0.3}
HiPro-adapted 75K / 90K		88.8 _{0.2}	57.4 _{0.3}	72.5 _{0.2}	77.6 _{0.2}	75.4 _{0.3}	74.6 _{0.4}	74.4 _{0.2}	-	-	-	-	-	-	-
PT	81K	87.1 _{0.6}	56.5 _{1.7}	71.9 _{3.9}	73.5 _{1.4}	72.5 _{0.7}	73.1 _{1.1}	72.5 _{1.0}	76.6 _{0.4}	48.9 _{0.6}	46.9 _{0.3}	50.4 _{0.2}	86.0 _{0.3}	56.2 _{2.3}	60.8 _{0.2}
DPT	87K	86.8 _{0.3}	55.8 _{3.7}	72.5 _{1.7}	77.2 _{2.2}	73.1 _{0.8}	73.3 _{0.9}	73.1 _{1.5}	76.8 _{0.5}	51.3 _{0.6}	48.6 _{0.1}	50.2 _{0.2}	85.5 _{0.6}	58.5 _{0.9}	61.8 _{0.2}
SMoP	86K	87.5 _{0.7}	55.4 _{3.0}	72.0 _{1.1}	78.3 _{1.1}	73.1 _{0.6}	73.5 _{0.8}	73.3 _{0.6}	77.1 _{1.3}	51.0 _{1.1}	47.8 _{1.4}	50.1 _{1.9}	85.9 _{0.4}	56.5 _{2.3}	61.4 _{0.7}
PT-MoE	87K	87.0 _{0.7}	57.4 _{0.7}	72.0 _{1.7}	75.2 _{1.4}	73.2 _{0.5}	72.1 _{1.3}	72.8 _{0.2}	76.2 _{0.6}	45.7 _{0.9}	49.2 _{1.0}	50.2 _{0.9}	85.5 _{0.5}	56.6 _{0.7}	60.6 _{0.4}
HiVe	87K	87.7 _{0.8}	59.2 _{0.9}	74.9 _{1.7}	80.5 _{0.5}	75.2 _{0.6}	74.9 _{0.5}	75.4 _{0.3}	78.3 _{0.9}	53.9 _{0.2}	48.9 _{0.4}	50.1 _{0.8}	86.0 _{1.0}	59.8 _{0.9}	62.8 _{0.3}

Table 1: MRQA F1 comparison on in- and out-of-domain datasets with mean and standard deviation.

Method	# of Params	In-domain							Out-of-domain					
		Wiki	Red	XS	CNN	Giga	Media	Avg	Dialog	Sci	AESLC	WH	MN	Avg
FT	1.2B	32.0 _{0.7}	20.0 _{0.3}	26.6 _{0.4}	23.0 _{1.5}	39.4 _{0.5}	14.9 _{1.1}	26.0 _{0.6}	22.0 _{1.6}	18.5 _{2.0}	14.8 _{0.4}	22.0 _{1.6}	10.7 _{1.2}	17.6 _{0.5}
LoRA	106K	30.2 _{0.5}	20.4 _{0.4}	25.3 _{0.5}	23.9 _{0.4}	38.1 _{1.1}	14.0 _{0.5}	25.3 _{0.4}	14.8 _{1.3}	19.0 _{1.3}	14.6 _{1.3}	16.9 _{3.0}	10.6 _{0.5}	15.2 _{0.4}
HydraLoRA	278K	30.4 _{0.4}	20.3 _{0.3}	25.7 _{0.4}	23.9 _{0.4}	38.6 _{0.9}	14.3 _{0.6}	25.5 _{0.4}	15.9 _{0.7}	18.1 _{0.4}	14.6 _{0.6}	20.2 _{0.5}	10.2 _{0.4}	15.8 _{0.4}
HiPro-adapted	75K / 90K	29.5 _{0.3}	20.6 _{0.3}	25.2 _{0.3}	25.0 _{0.2}	38.5 _{0.3}	16.0 _{0.6}	25.8 _{0.3}	—	—	—	—	—	—
PT	81K	30.0 _{0.4}	20.2 _{0.3}	25.0 _{0.4}	21.6 _{0.6}	38.0 _{0.4}	16.9 _{0.4}	25.3 _{0.4}	18.8 _{1.5}	18.6 _{1.1}	13.1 _{0.8}	19.9 _{0.4}	8.8 _{0.5}	15.8 _{0.4}
DPT	87K	30.3 _{0.3}	20.0 _{0.4}	25.3 _{0.3}	21.9 _{0.4}	38.0 _{0.4}	16.9 _{0.4}	25.4 _{0.3}	18.6 _{0.8}	18.1 _{0.9}	14.1 _{0.8}	20.8 _{0.6}	8.9 _{0.4}	16.1 _{0.4}
SMoP	86K	30.4 _{0.4}	19.8 _{0.6}	24.7 _{0.3}	23.2 _{0.5}	37.9 _{0.4}	15.8 _{0.4}	25.3 _{0.3}	18.1 _{0.6}	20.2 _{1.1}	14.1 _{0.4}	19.0 _{0.5}	9.1 _{0.4}	16.1 _{0.3}
PT-MoE	87K	29.2 _{0.4}	20.5 _{0.4}	25.0 _{0.4}	20.3 _{0.6}	38.4 _{0.4}	17.0 _{0.4}	25.1 _{0.5}	18.8 _{0.4}	18.3 _{0.6}	14.3 _{1.0}	17.9 _{1.3}	8.1 _{0.6}	15.5 _{0.7}
HiVe	87K	30.5 _{0.3}	20.6 _{0.7}	25.9 _{0.2}	24.0 _{0.5}	38.8 _{0.1}	16.7 _{2.5}	26.1 _{0.6}	17.5 _{1.6}	18.7 _{2.0}	14.4 _{0.1}	20.9 _{1.0}	12.3 _{0.6}	16.8 _{0.3}

Table 2: Summarization ROUGE-L comparison on in- and out-of-domain datasets with mean and standard deviation.

Summarization. Table 2 presents the ROUGE-L results on the summarization benchmark. Similar trends were observed on summarization tasks. In the in-domain setting, HiVe achieved an average ROUGE-L score of 26.1, compared with FT (26.0). It also achieved the best performance on five out of six in-domain tasks, improving the average ROUGE-L score over SMoP and PT-MoE by 0.8 and 1.0 points, respectively. Relative to HiPro-adapted, HiVe yields a 0.3-point higher average ROUGE-L score. These results indicate that the improvements are maintained across diverse in-domain summarization datasets. In OOD settings, HiVe improved the average ROUGE-L score over SMoP and PT-MoE by 0.7 and 1.3 points, respectively, with large gains observed on the Multi-News dataset, the most challenging task in our benchmark. The OOD results further demonstrate stable performance under distribution shifts. These results suggest that HiVe performs effectively across both extractive and generative tasks.

4.2.2 Ablation Study

Table 3 presents ablation studies on the core components of HiVe, including residual decomposition, hierarchical structure, and V-MoE routing. Additional ablation results are provided in Appendix F.

Residual Decomposition. Removing residual decomposition (*w/o Residual Decomposition*) de-

Model	MRQA		Summ.	
	In	Out	In	Out
HiVe	75.4 _{0.3}	62.8 _{0.3}	26.1 _{0.6}	16.8 _{0.3}
<i>w/o Residual Decomp.</i>	74.9 _{0.5} (-0.5)	61.8 _{0.2} (-1.0)	26.0 _{0.3} (-0.1)	14.4 _{0.4} (-2.4)
<i>w/o Hierarchy</i>				
Base-level only	73.6 _{0.7} (-1.8)	62.1 _{0.2} (-0.7)	25.4 _{0.1} (-0.7)	16.0 _{0.4} (-0.8)
Group-level only	73.7 _{0.7} (-1.7)	61.1 _{0.8} (-1.7)	25.5 _{0.1} (-0.6)	15.2 _{1.0} (-1.6)
Task-level only	74.4 _{0.6} (-1.0)	61.6 _{0.4} (-1.2)	26.1 _{0.2} (0.0)	14.5 _{0.6} (-2.3)
<i>w/o V-MoE</i>				
Base-key only	71.1 _{0.1} (-4.3)	63.1 _{0.2} (+0.3)	20.5 _{0.4} (-5.6)	15.0 _{0.1} (-1.8)
Group-key only	74.5 _{0.5} (-0.9)	62.6 _{0.1} (-0.2)	20.9 _{0.8} (-5.2)	14.8 _{0.1} (-2.0)
Task-key only	75.1 _{0.2} (-0.3)	62.7 _{0.1} (-0.1)	26.1 _{0.3} (0.0)	16.0 _{0.2} (-0.8)

Table 3: Ablation study of HiVe on MRQA and summarization. The evaluation covers the three core components: residual decomposition, hierarchical structure, and V-MoE routing. Performance changes relative to the full model are shown in parentheses.

grades performance in both in-domain and out-of-domain datasets, with larger drops in OOD settings. This suggests that, without decomposition, shared and task-specific representations become entangled, making it difficult to preserve shared representations across hierarchy levels. Consequently, each level may fail to capture its intended role.

Field	Case 1	Case 2	Case 3
Context	Comets are small, icy objects that have very elliptical orbits around the Sun. Their orbits carry them from the outer solar system to the inner solar system, close to the Sun. Early in Earth’s history, comets may have brought <u>water</u> and other substances to Earth during collisions. Comet tails form the outer layers of ice melt...	<u>The Stock Market crash in New York</u> led people to hoard their money ; as consumption fell , the American economy steadily contracted , 1929 - 32 . Given the close economic links between the two countries , the collapse quickly affected Canada . Added to the woes of the prairies were those of Ontario...	On June 4, 2014, the NFL announced that the practice of branding Super Bowl games with Roman numerals, a practice established at Super Bowl V, would be temporarily suspended, and that the game would be named using <u>Arabic numerals</u> as Super Bowl 50 as opposed to Super Bowl L. The use of Roman numerals...
Question	In the early solar system, comets that struck Earth may have brought in what?	How did the great depression start in canada?	What type of numeral did the latest Super Bowl use to designate the game number?
Answer	water	The Stock Market crash in New York	Arabic numerals
Base	<u>water</u>	stock market crash	Roman
Base + Group	water and other substances	<u>The Stock Market crash in New York</u>	Roman numerals
Base + Group + Task	water and other substances	The Stock Market crash in New York led people to hoard their money ; as...	<u>Arabic numerals</u>

Table 4: Case study of progressively specialized prompt compositions in HiVe. Underlined predictions indicate the specialization level selected by HiVe. The yellow-highlighted spans denote the ground-truth answers.

Hierarchical Architecture. Variants using only a single hierarchy level (*Base-level only*, *Group-level only*, and *Task-level only*) consistently underperform the full model. In particular, *Task-level only* remains relatively strong in in-domain settings but shows larger degradation under domain shifts, whereas *Base-level only* and *Group-level only* exhibit lower overall performance. These results suggest that the proposed Base–Group–Task hierarchy effectively balances shared and task-specific representations across hierarchy levels.

Vertical Mixture of Experts. Static routing variants using only a single key type (*Base-key only*, *Group-key only*, and *Task-key only*) generally degrade performance across most settings, particularly on summarization tasks. Although *Task-key only* maintains relatively strong performance, its robustness under domain shifts is reduced. In contrast, *Base-key only* slightly improves performance on MRQA out-of-domain tasks. These findings suggest that the optimal level of specialization may vary depending on the input, supporting the need for multi-level dynamic routing.

4.3 In-Depth Analysis

4.3.1 Case Study

We qualitatively analyze the effectiveness of V-MoE routing by comparing predictions generated with progressively expanded hierarchy levels for the same input. As shown in Table 4, different hierarchy levels produce different degrees of semantic specificity. In some cases, the Base-level predictor generates concise and generalized answers, while the Task-level predictor captures finer contextual

details. The Group-level predictor often exhibits an intermediate level of specialization, avoiding both overly generic and excessively verbose predictions. Notably, HiVe tends to select different hierarchy levels depending on the input characteristics, producing the most appropriate prediction for each case. For example, in Case 1, only the Base-level predictor correctly outputs the concise answer “water,” whereas the Group- and Task-level predictors include unnecessary contextual spans. In Case 2, the Group-level predictor produces the most appropriate level of specificity, while the Base-level predictor is overly generic and the Task-level predictor generates an excessively verbose response. Finally, in Case 3, only the Task-level predictor successfully captures the subtle distinction between Roman numerals and Arabic numerals.

4.3.2 Grouping Analysis

Figure 3 shows that task grouping gradually stabilizes during training. The mean ARI score consistently increases and eventually satisfies the stability criterion for both MRQA and summarization. To avoid unreliable early-stage group formation, stability checks are activated only after the warm-up stage, and measurement begins only after the rolling window is fully populated. The HAC visualizations further show that semantically related tasks form coherent groups, such as Tri and Srch in MRQA, and Giga and XS in summarization. Although the overall grouping structure remains stable, minor variations are occasionally observed across different hyperparameter settings.

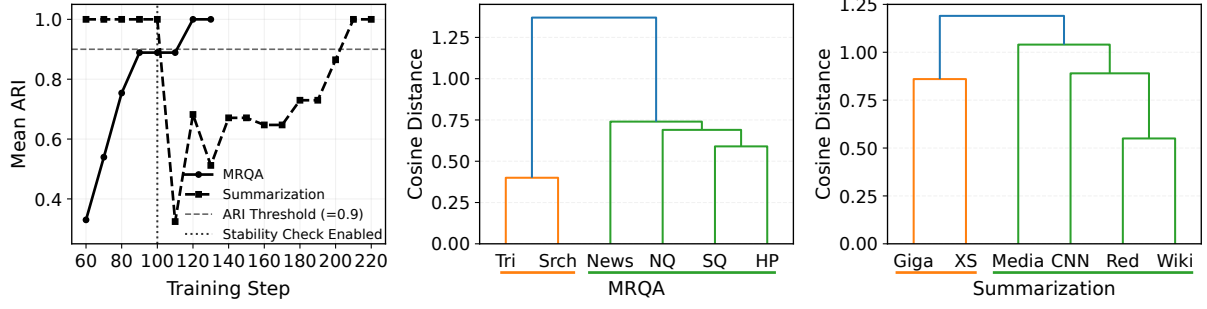


Figure 3: Grouping stability during training. The left figure shows the mean ARI across training steps, and the right figure shows HAC-based grouping results, where underlined task names indicate the same group.

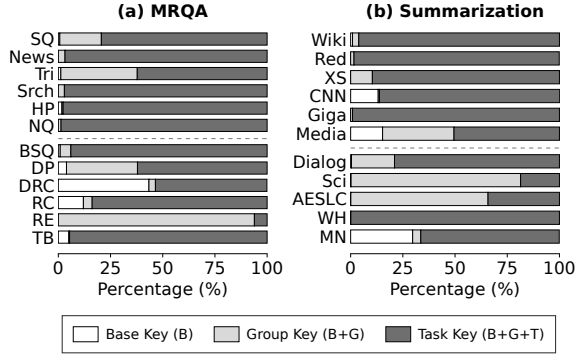


Figure 4: Routing key level selection rates across datasets. Different tasks exhibit distinct preferences for hierarchical routing levels.

4.3.3 Routing Analysis

Figure 4 presents the routing key selection distribution across hierarchy levels. Overall, different tasks exhibit distinct routing preferences across hierarchy levels. For example, RE strongly favors Group-level routing, while DRC shows increased reliance on Base-level routing. In contrast, several tasks more frequently utilize Task-level routing, indicating that the preferred level of specialization varies depending on the task and domain characteristics. These results suggest that using a fixed specialization level may be suboptimal for multi-task adaptation, motivating adaptive routing over hierarchical prompts.

4.3.4 Model Hyperparameter Sensitivity

Figure 5 shows the effects of three main hyperparameters used in HiVe.

Prompt Length. We analyze the effect of prompt length (20, 40, 60, and 80). Overall, the most stable performance is observed at shorter prompt lengths (20–40), while longer lengths tend to degrade performance. In particular, both in-domain and out-of-domain performance noticeably decrease at length 60. Although partial recovery

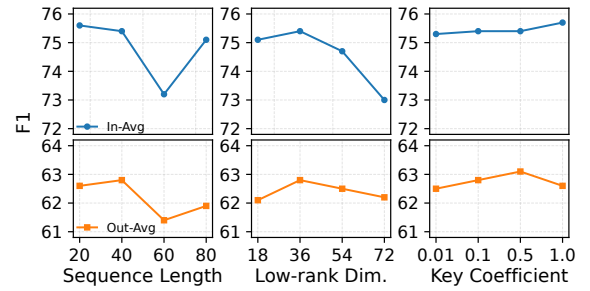


Figure 5: Hyperparameter sensitivity analysis of HiVe on MRQA across different configurations. From left to right, the figures illustrate the effects of prompt length, low-rank dimension, and key loss coefficient on in-domain and out-of-domain performance.

is observed at length 80, the overall performance remains lower than that of shorter prompt lengths. These results suggest that large prompt spaces do not necessarily yield better results.

Low-Rank Dimension. We analyze the effect of low-rank dimensionality (18, 36, 54, and 72). Performance consistently peaks at dimension 36 across both in-domain and out-of-domain settings. Larger dimensions gradually degrade performance, with a more pronounced decline in the in-domain setting. These results suggest that increasing the low-rank dimension beyond a moderate size does not provide additional performance gains.

Key Coefficient. We also evaluate the effect of the key loss coefficient λ on MRQA, which controls the strength of routing supervision. Small coefficients lead to insufficient routing optimization, while excessively large coefficients occasionally degrade overall generalization performance, possibly because the shared projection matrix W used by both prompts and routing keys becomes more biased toward routing-specific features. Overall, moderate coefficients (0.1–0.5) provide the most stable performance.

Setting	Step	Part.	In	OOD
Default	220	A	26.5	16.6
$K=3$	200	A	26.4	16.7
$K=7$	230	A	26.5	17.0
$N=5$	185	A	26.5	16.2
$N=20$	280	A	26.3	16.0
$\eta=0.7$	190	A	26.4	16.2
$\eta=0.8$	210	A	26.5	16.6
$R=1$	100	B	25.6	16.1
$R=3$	230	A	26.5	17.0

Table 5: Grouping hyperparameter sensitivity on summarization using a single seed. Partitions A and B denote {Giga, XS}/{Red, Wiki, CNN, Media} and {Red, Wiki, CNN}/{Giga, XS, Media}, respectively.

Benchmark	Seed	Step	Part.	In	OOD
MRQA	1	130	A	75.3	62.7
MRQA	2	110	A	75.7	62.6
MRQA	3	160	A	75.2	63.2
Summ.	1	220	B	26.5	16.6
Summ.	2	280	B	26.4	16.7
Summ.	3	110	C	25.4	17.1

Table 6: Seed sensitivity of task grouping. Partitions A, B, and C denote {Tri, Srch}/{SQ, News, HP, NQ}, {Giga, XS}/{Red, Wiki, CNN, Media}, and {CNN, Giga, Media}/{Wiki, Red, XS}, respectively.

4.3.5 Grouping Sensitivity

We further analyze the sensitivity of task grouping to its hyperparameters and random initialization.

Hyperparameter Sensitivity. Table 5 reports the effects of the grouping hyperparameters on summarization by varying the window size K , check interval N , ARI threshold η , and patience R . Across the tested ranges, these hyperparameters mainly affect when grouping is finalized, while the resulting partition and downstream performance remain largely stable. The main exception is $R=1$, which finalizes grouping earlier, produces a different partition, and lowers in-domain performance. This suggests that grouping should be finalized only after stability is confirmed across multiple checks.

Random-Seed Sensitivity. Table 6 evaluates the sensitivity of task grouping to random seeds. On MRQA, all seeds recover the same partition and exhibit similar in-domain and OOD performance. On summarization, two seeds yield the same partition, whereas one seed produces a different grouping with lower in-domain but slightly higher OOD performance. Thus, seed-dependent partition variation does not consistently improve or degrade performance in either the in-domain or OOD setting.

5 Conclusion

We proposed HiVe, a hierarchical prompt tuning framework that adaptively controls the level of prompt specialization for each input through vertical mixture-of-experts (V-MoE) routing. By combining stability-driven task grouping and residual prompt decomposition, HiVe effectively disentangles shared and task-specific knowledge across hierarchical levels. Beyond parameter sharing, the learned hierarchy also serves as an input-dependent routing space for controlling the degree of specialization required by each input. Experimental results on question answering and summarization benchmarks demonstrate that HiVe consistently outperforms strong prompt tuning baselines in both in-domain and out-of-domain settings while maintaining parameter efficiency. Overall, our findings suggest that dynamically controlling prompt specialization provides an effective direction for multitask learning. Future work could explore soft or top- k routing as alternatives to the current hard-routing scheme for more flexible specialization across hierarchical levels.

Limitations

HiVe relies on stability-driven task grouping to construct hierarchical prompts. Since the group structure emerges from learned task prompt representations, it can be influenced by factors such as random seeds, prompt length, low-rank dimensionality, and grouping hyperparameters. As a result, some configurations may finalize grouping prematurely, leading to different group structures and reduced downstream performance. Furthermore, HiVe is evaluated on within-task distribution shifts, where the task objective remains unchanged while the input distribution varies, while generalization across task types remains outside the scope of this study. Adaptive routing can also mitigate over- or under-specialization for unseen inputs, but cannot capture patterns beyond the learned hierarchy, leaving gaps under substantial distribution shifts.

Acknowledgments

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korea government(MSIT) (No. RS-2026-25507543 (40%), No. IITP-2026-RS-2025-02304828 (40%), No. IITP-2026-RS-2020-II201819 (10%), and No. RS-2026-25585074 (10%)).

References

- Akari Asai, Mohammadreza Salehi, Matthew Peters, and Hannaneh Hajishirzi. 2022. [ATTEMPT: Parameter-efficient multi-task tuning via attentional mixtures of soft prompts](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6655–6672, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Joachim Bingel and Anders Søgaard. 2017. [Identifying beneficial task relations for multi-task learning in deep neural networks](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 164–169, Valencia, Spain. Association for Computational Linguistics.
- Isabel Cachola, Kyle Lo, Arman Cohan, and Daniel Weld. 2020. [TLDR: Extreme summarization of scientific documents](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4766–4777, Online. Association for Computational Linguistics.
- Guoxin Chen, Yiming Qian, Bowen Wang, and Liangzhi Li. 2023. [MPrompt: Exploring multi-level prompt tuning for machine reading comprehension](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5163–5175, Singapore. Association for Computational Linguistics.
- Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021. [DialogSum: A real-life scenario dialogue summarization dataset](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5062–5074, Online. Association for Computational Linguistics.
- Joon-Young Choi, Junho Kim, Jun-Hyung Park, Wing-Lam Mok, and SangKeun Lee. 2023. [SMoP: Towards efficient and effective prompt tuning with sparse mixture-of-prompts](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14306–14316, Singapore. Association for Computational Linguistics.
- Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. [DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs](#). 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 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.
- Chen Dun, Mirian Hipolito Garcia, Guoqing Zheng, Ahmed H. Awadallah, Robert Sim, and Anastasios Kyrillidis. 2025. [Sweeping heterogeneity with smart mops: Mixture of prompts for LLM task adaptation](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 16426–16434, Philadelphia, USA. AAAI Press.
- Matthew Dunn, Levent Sagun, Mike Higgins, V. Ugur Guney, Volkan Cirik, and Kyunghyun Cho. 2017. [SearchQA: A new q&a dataset augmented with context from a search engine](#). *arXiv preprint arXiv:1704.05179*.
- Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. [Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.
- Chris Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. 2021. [Efficiently identifying task groupings for multi-task learning](#). In *Advances in Neural Information Processing Systems*, volume 34, pages 27503–27516, Online. Curran Associates, Inc.
- Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. [MRQA 2019 shared task: Evaluating generalization in reading comprehension](#). In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 1–13, Hong Kong, China. Association for Computational Linguistics.
- David Graff and Christopher Cieri. 2003. [English gigaword](#). LDC Catalog No. LDC2003T05.
- Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. [Teaching machines to read and comprehend](#). In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Gold Wang, Lu Wang, and Weizhu Chen. 2022. [LoRa: Low-rank adaptation of large language models](#). In *Proceedings of the 10th International Conference on Learning Representations (ICLR)*, Online. OpenReview.net.
- Lawrence Hubert and Phipps Arabie. 1985. [Comparing partitions](#). *Journal of Classification*, 2(1):193–218.
- Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. [TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.
- Aniruddha Kembhavi, Minjoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. [Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension](#). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4999–5007, Honolulu, Hawaii. IEEE.

- Doyoung Kim, Susik Yoon, Dongmin Park, Youngjun Lee, Hwanjun Song, Jihwan Bang, and Jae-Gil Lee. 2024. [One size fits all for semantic shifts: Adaptive prompt tuning for continual learning](#). In *Proceedings of the 41st International Conference on Machine Learning*, Vienna, Austria. JMLR.org.
- Mahnaz Koupaee and William Yang Wang. 2018. [WikiHow: A large scale text summarization dataset](#). *arXiv preprint arXiv:1810.09305*.
- Abhishek Kumar and Hal Daumé III. 2012. [Learning task grouping and overlap in multi-task learning](#). In *Proceedings of the 29th International Conference on Machine Learning*, pages 1383–1390, Edinburgh, Scotland, UK. PMLR.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. [Natural questions: A benchmark for question answering research](#). *Transactions of the Association for Computational Linguistics*, 7:452–466.
- Faisal Ladhak, Esin Durmus, Claire Cardie, and Kathleen McKeown. 2020. [WikiLingua: A new benchmark dataset for cross-lingual abstractive summarization](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4034–4048, Online. Association for Computational Linguistics.
- Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. [RACE: Large-scale Reading comprehension dataset from examinations](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.
- Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. [The power of scale for parameter-efficient prompt tuning](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. [Zero-shot relation extraction via reading comprehension](#). In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada. Association for Computational Linguistics.
- Zongqian Li, Yixuan Su, and Nigel Collier. 2025. [PT-MoE: An efficient finetuning framework for integrating mixture-of-experts into prompt tuning](#). In *Advances in Neural Information Processing Systems*, volume 38, pages 83240–83261, San Diego, California. Curran Associates, Inc.
- Yajing Liu, Yuning Lu, Hao Liu, Yaozu An, Zhuoran Xu, Zhuokun Yao, Baofeng Zhang, Zhiwei Xiong, and Chenguang Gui. 2023. [Hierarchical prompt learning for multi-task learning](#). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10888–10898, Vancouver, Canada. IEEE.
- Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. [Don’t give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.
- Jingwei Ni, Zhijing Jin, Qian Wang, Mrinmaya Sachan, and Markus Leippold. 2023. [When does aggregating multiple skills with multi-task learning work? a case study in financial NLP](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7465–7488, Toronto, Canada. Association for Computational Linguistics.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. [SQuAD: 100,000+ questions for machine comprehension of text](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.
- Amrita Saha, Rahul Aralikkatte, Mitesh M. Khapra, and Karthik Sankaranarayanan. 2018. [DuoRC: Towards complex language understanding with paraphrased reading comprehension](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1693, Melbourne, Australia. Association for Computational Linguistics.
- Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. [Outrageously large neural networks: The sparsely-gated mixture-of-experts layer](#). In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, Toulon, France. OpenReview.net.
- Chunlin Tian, Zhan Shi, Zhijiang Guo, Li Li, and Chengzhong Xu. 2024. [HydraLoRa: An asymmetric LoRa architecture for efficient fine-tuning](#). In *Advances in Neural Information Processing Systems*, volume 37, Vancouver, Canada. Curran Associates, Inc.
- Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordani, Philip Bachman, and Kaheer Suleman. 2017. [NewsQA: A machine comprehension dataset](#). In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200, Vancouver, Canada. Association for Computational Linguistics.
- George Tsatsaronis, Michael Schroeder, Georgios Paliouras, Yannis Almirantis, Ion Androutsopoulos, Eric Gaussier, Patrick Gallinari, Thierry Artieres,

- Michael R. Alvers, Matthias Zschunke, and Axel-Cyrille Ngonga Ngomo. 2012. [BioASQ: A challenge on large-scale biomedical semantic indexing and question answering](#). In *Proceedings of the AAAI Fall Symposium on Information Retrieval and Knowledge Discovery in Biomedical Text*, pages 92–98, Arlington, Virginia. AAAI Press.
- Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. 2017. [TL;DR: Mining Reddit to learn automatic summarization](#). In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 59–63, Copenhagen, Denmark. Association for Computational Linguistics.
- Luping Wang, Sheng Chen, Linnan Jiang, Shu Pan, Runze Cai, Sen Yang, and Fei Yang. 2025. [Parameter-efficient fine-tuning in large language models: A survey of methodologies](#). *Artificial Intelligence Review*, 58(8):227.
- Yaqing Wang, Sahaj Agarwal, Subhabrata Mukherjee, Xiaodong Liu, Jing Gao, Ahmed Hassan Awadallah, and Jianfeng Gao. 2022a. [AdaMix: Mixture-of-adaptations for parameter-efficient model tuning](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5744–5760, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Zihan Wang, Peiyi Wang, Tianyu Liu, Binghuai Lin, Yunbo Cao, Zhifang Sui, and Houfeng Wang. 2022b. [HPT: Hierarchy-aware prompt tuning for hierarchical text classification](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3740–3751, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Yao Xiao, Lu Xu, Jiayi Li, Wei Lu, and Xiaoli Li. 2023. [Decomposed prompt tuning via low-rank reparameterization](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13335–13347, Singapore. Association for Computational Linguistics.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. [HotpotQA: A dataset for diverse, explainable multi-hop question answering](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.
- Peiyi Zhang, Richong Zhang, Zhijie Nie, and Ziqiao Wang. 2025. [Dynamic task vector grouping for efficient multi-task prompt tuning](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 26805–26821, Vienna, Austria. Association for Computational Linguistics.
- Rui Zhang and Joel Tetreault. 2019. [This email could save your life: Introducing the task of email subject line generation](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 446–456, Florence, Italy. Association for Computational Linguistics.
- Chenguang Zhu, Yang Liu, Jie Mei, and Michael Zeng. 2021. [MediaSum: A large-scale media interview dataset for dialogue summarization](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5927–5934, Online. Association for Computational Linguistics.

Appendix

A Dataset Details

Table 7 summarizes the datasets used across question answering and summarization tasks, divided into in-domain and out-of-domain splits. For summarization, the numbers of training, validation, and test samples are capped at 40K, 10K, and 3K per task, respectively.

Split	Dataset	Abbrev.
Question Answering		
In	SQuAD (Rajpurkar et al., 2016)	SQ
In	NewsQA (Trischler et al., 2017)	News
In	TriviaQA (Joshi et al., 2017)	Tri
In	SearchQA (Dunn et al., 2017)	Srch
In	HotpotQA (Yang et al., 2018)	HP
In	Natural Questions (Kwiatkowski et al., 2019)	NQ
Out	BioASQ (Tsatsaronis et al., 2012)	BSQ
Out	DROP (Dua et al., 2019)	DP
Out	DuoRC (Saha et al., 2018)	DRC
Out	RACE (Lai et al., 2017)	RC
Out	RelationExtraction (Levy et al., 2017)	RE
Out	TextbookQA (Kembhavi et al., 2017)	TB
Summarization		
In	WikiLingua (Ladhak et al., 2020)	Wiki
In	Reddit TIFU (Völske et al., 2017)	Red
In	XSum (Narayan et al., 2018)	XS
In	CNN/DailyMail (Hermann et al., 2015)	CNN
In	Gigaword (Graff and Cieri, 2003)	Giga
In	MediaSum (Zhu et al., 2021)	Media
Out	DialogSum (Chen et al., 2021)	Dialog
Out	SciTLDR (Cachola et al., 2020)	Sci
Out	AESLC (Zhang and Tetreault, 2019)	AESLC
Out	WikiHow (Koupaee and Wang, 2018)	WH
Out	Multi-News (Fabbri et al., 2019)	MN

Table 7: Datasets used in our experiments.

B Input Templates

All inputs are formatted using a unified conversational template based on the Llama-3.2-1B-Instruct chat format. For question answering tasks, inputs are formatted as:

```
<|start_header_id|>user<|end_header_id|>
Context: {context}
Question: {question}
<|eot_id|>
<|start_header_id|>assistant
<|end_header_id|>
Answer:
```

For summarization tasks, the following template is used:

```
<|start_header_id|>user<|end_header_id|>
Document: {document}
<|eot_id|>
<|start_header_id|>assistant
<|end_header_id|>
Summary:
```

During training, the target sequence is appended after the assistant prefix, while during inference it is excluded.

C Parameter Initialization

To improve optimization stability, soft prompts are initialized from natural language templates rather than random vectors. The base prompt P_B and projection matrix W are initialized using generic task-independent instructions to reduce task-specific bias:

```
[ "Process input and infer task.",
  "Select relevant information.",
  "Model relationships in the input.",
  "Maintain consistency with the input.",
  "Generate an appropriate response." ]
```

Given the averaged template embedding matrix $E \in \mathbb{R}^{l \times d}$, SVD is applied:

$$E = U \Sigma V^\top.$$

The low-rank initialization is constructed using the truncated SVD components, where r denotes the target low-rank dimension:

$$P_B = U_r \Sigma_r^{1/2}, \quad W = \Sigma_r^{1/2} V_r^\top.$$

Task prompts are initialized from task-specific template embeddings projected into the shared low-rank space. For each task τ , the task-specific embedding matrix $E_\tau \in \mathbb{R}^{l \times d}$ is extracted from the frozen backbone and projected via W :

$$P_T^{(\tau)} = E_\tau W^\top.$$

Prior to Phase 2, the task-specific embedding matrix is mean-pooled to obtain $\bar{E}_\tau \in \mathbb{R}^d$, and the task key is initialized as:

$$k_T^{(\tau)} = \bar{E}_\tau W^\top.$$

D Implementation Details

D.1 Experiment Settings

Table 8 summarizes the training and generation settings used across question answering and summarization experiments.

Setting	MRQA	Summarization
GPU	1× RTX PRO 6000 Blackwell	
Batch size	16	
Gradient accumulation steps	16	
Max sequence length	512	768
Max new tokens	100	128
Training steps	1000	
Warm-up steps	100	
Evaluation interval	Every 100 steps	
Number of beams	1	
Do sample	False	

Table 8: Training and generation settings used for MRQA and Summarization experiments. For HiVe, evaluation is performed only during Phase 2.

Method	Hyperparameter Settings
FT	learning rate=3e-5
LoRA	learning rate=3e-4; $r = 1$; $\alpha = 16$; target modules={q_proj, v_proj}
HydraLoRA	learning rate=3e-4; $r = 1$; $\alpha = 16$; expert branches=2; target modules={q_proj, v_proj}
PT	learning rate=3e-3; prompt length=40; initialization=template
DPT	learning rate=3e-3; prompt length=40; low-rank dim=42; initialization=template
SMoP	learning rate=3e-3; total prompts=40; prompts=2; initialization=template
PT-MoE	learning rate=3e-3; prompt length=40; prompts=2; low-rank dim=39; initialization=template
HiPro-adapted	learning rate=3e-3; Stage A/B training steps=200/800; prompt length=40; low-rank dim=36
HiVe	learning rate=3e-3; prompt length=40; base/group/task prompts=1/2/6; low-rank dim=36; key coefficient=0.1; ARI threshold=0.9; check interval=10; stability window=5; stability patience=2

Table 9: Main hyperparameter settings used for baseline methods and HiVe.

Method	# of Params	In-domain							Out-of-domain						
		SQ	News	Tri	Srch	HP	NQ	Avg	BSQ	DP	DRC	RC	RE	TB	Avg
FT	1.2B	79.3 _{1.3}	44.9 _{0.6}	65.6 _{1.8}	76.0 _{0.6}	59.9 _{0.7}	60.9 _{0.4}	64.4 _{0.4}	58.1 _{4.0}	40.0 _{1.3}	40.1 _{2.2}	34.6 _{0.8}	74.8 _{1.8}	48.7 _{2.3}	49.4 _{1.7}
LoRA	106K	79.0 _{0.6}	43.0 _{0.7}	69.2 _{0.5}	72.6 _{1.2}	58.3 _{0.4}	60.2 _{0.2}	63.7 _{0.4}	61.7 _{0.1}	38.6 _{1.3}	40.3 _{0.3}	39.3 _{1.1}	73.2 _{1.0}	51.5 _{1.3}	50.8 _{0.7}
HydraLoRA	278K	79.8 _{0.9}	42.5 _{1.0}	69.3 _{0.6}	73.1 _{1.8}	58.9 _{0.5}	60.8 _{0.2}	64.0 _{0.5}	60.8 _{2.4}	39.9 _{1.8}	40.4 _{0.9}	39.2 _{1.3}	73.5 _{0.6}	49.2 _{2.2}	50.5 _{0.7}
PT	81K	78.6 _{1.0}	39.6 _{2.0}	62.5 _{6.9}	61.3 _{5.2}	56.2 _{0.6}	59.3 _{0.8}	59.6 _{2.0}	59.5 _{2.8}	39.6 _{1.5}	37.5 _{0.2}	38.5 _{0.2}	74.9 _{0.8}	46.3 _{3.8}	49.4 _{0.2}
DPT	87K	78.4 _{0.3}	38.3 _{4.0}	64.6 _{3.7}	68.7 _{3.9}	57.0 _{1.0}	59.4 _{1.1}	61.1 _{2.3}	61.0 _{4.0}	42.0 _{0.6}	39.3 _{0.3}	37.5 _{0.9}	73.4 _{1.5}	49.9 _{0.5}	50.5 _{1.2}
SMoP	86K	79.4 _{1.2}	38.5 _{3.6}	63.4 _{2.7}	71.4 _{1.3}	56.9 _{0.8}	59.9 _{1.2}	61.6 _{0.8}	61.0 _{3.2}	41.8 _{1.0}	38.1 _{2.1}	37.6 _{2.8}	74.4 _{1.0}	47.3 _{2.3}	50.0 _{1.2}
PT-MoE	87K	78.5 _{0.9}	40.6 _{0.8}	63.2 _{4.3}	66.5 _{3.1}	57.0 _{0.2}	58.5 _{1.2}	60.7 _{0.6}	59.9 _{1.7}	36.5 _{0.8}	40.5 _{0.9}	38.2 _{0.4}	74.0 _{1.0}	48.2 _{0.7}	49.6 _{0.3}
HiVe	87K	79.8 _{1.1}	42.8 _{1.0}	67.9 _{4.2}	73.6 _{0.8}	59.2 _{0.2}	61.9 _{0.3}	64.2 _{0.5}	64.0 _{2.0}	44.6 _{0.2}	39.6 _{0.7}	38.0 _{1.4}	74.6 _{2.2}	52.1 _{1.2}	52.1 _{0.7}

Table 10: MRQA EM comparison on in-domain and out-of-domain datasets with mean and standard deviation.

Method	# of Params	In-domain							Out-of-domain						
		SQ	News	Tri	Srch	HP	NQ	Avg	BSQ	DP	DRC	RC	RE	TB	Avg
SMoP	172K	94.8	68.6	85.4	88.8	82.3	80.0	83.3	82.1	71.3	57.7	65.8	87.9	72.2	72.8
PT-MoE	171K	94.3	68.0	85.7	88.3	80.8	80.4	82.9	81.8	67.9	56.0	63.0	88.2	74.3	71.9
HiVe	157K	95.4	68.0	85.8	89.8	82.6	82.3	84.2	81.4	74.7	58.2	65.2	88.3	74.6	73.7

Table 11: MRQA F1 comparison on in-domain and out-of-domain datasets with Llama-3.1-8B-Instruct.

Method	# of Params	In-domain							Out-of-domain					
		Wiki	Red	XS	CNN	Giga	Media	Avg	Dialog	Sci	AESLC	WH	MN	Avg
SMoP	172K	<u>36.3</u>	<u>23.5</u>	<u>33.1</u>	23.2	<u>41.8</u>	16.9	<u>29.1</u>	<u>18.7</u>	17.9	<u>17.8</u>	25.6	10.3	18.1
PT-MoE	171K	35.5	23.0	32.7	<u>23.5</u>	41.4	<u>17.2</u>	28.9	18.1	<u>19.7</u>	17.9	24.8	<u>10.4</u>	<u>18.2</u>
HiVe	159K	37.4	24.1	34.0	26.9	42.4	18.7	30.6	19.8	20.6	15.5	<u>25.0</u>	13.4	18.9

Table 12: Summarization ROUGE-L comparison on in-domain and out-of-domain datasets with Llama-3.1-8B-Instruct.

D.2 Baseline Hyperparameter Settings

Table 9 summarizes the main hyperparameter settings used in our experiments. For fair comparison, learning rates followed common settings for each method family, prompt lengths were fixed across methods, and low-rank dimensions were adjusted for comparable parameter budgets. For MoE-based baselines, the number of prompts matched the number of group prompts in HiVe.

E Additional Experiments

E.1 MRQA EM Results

Table 10 presents detailed MRQA exact match (EM) results across in- and out-of-domain datasets.

Method	Prompt Length	Relative FLOPs	Latency (ms)
PT	40	1.1624×	38.17
DPT	40	1.1624×	44.78
SMoP	20	1.0797×	40.98
PT-MoE	40	1.1624×	38.43
HiVe	40	1.1624×	38.38

Table 13: Inference efficiency comparison with prompt-tuning baselines.

E.2 Additional Backbone Experiments

Tables 11 and 12 present additional results using Llama-3.1-8B-Instruct, showing that HiVe achieves higher average performance on both benchmarks while using fewer parameters.

Method	In-domain							Out-of-domain						
	SQ	News	Tri	Srch	HP	NQ	Avg	BSQ	DP	DRC	RC	RE	TB	Avg
HiVe	87.7 _{0.8}	59.2 _{0.9}	74.9 _{1.7}	80.5 _{0.5}	75.2 _{0.6}	74.9 _{0.5}	75.4 _{0.3}	78.3 _{0.9}	53.9 _{0.2}	48.9 _{0.4}	50.1 _{0.8}	86.0 _{1.0}	59.8 _{0.9}	62.8 _{0.3}
<i>w/o Residual Decomp.</i>	87.4 _{0.3}	58.7 _{2.0}	74.0 _{1.4}	80.2 _{0.2}	74.8 _{0.7}	74.4 _{0.5}	74.9 _{0.5}	76.5 _{1.4}	51.8 _{0.6}	48.0 _{1.1}	49.7 _{1.4}	85.9 _{0.6}	59.2 _{0.7}	61.8 _{0.2}
<i>w/o Hierarchical Structure</i>														
Base-level only	87.0 _{0.2}	57.0 _{1.4}	73.8 _{0.8}	77.3 _{1.3}	73.2 _{0.3}	72.9 _{0.7}	73.6 _{0.7}	78.0 _{0.4}	50.9 _{1.1}	48.7 _{0.8}	50.2 _{0.8}	86.3 _{0.3}	58.7 _{1.3}	62.1 _{0.2}
Group-level only	86.7 _{0.9}	59.1 _{0.6}	72.7 _{1.3}	77.7 _{1.0}	73.2 _{0.6}	72.9 _{0.3}	73.7 _{0.7}	77.1 _{0.5}	49.5 _{0.5}	49.1 _{0.8}	49.7 _{1.2}	85.7 _{0.6}	55.8 _{2.7}	61.1 _{0.8}
Task-level only	88.2 _{0.4}	57.5 _{1.3}	73.1 _{1.9}	78.8 _{1.4}	74.5 _{0.5}	74.5 _{0.3}	74.4 _{0.6}	76.9 _{0.1}	53.3 _{0.4}	46.6 _{1.0}	47.9 _{0.8}	86.5 _{0.2}	58.7 _{1.4}	61.6 _{0.4}
<i>w/o Vertical Mixture-of-Experts</i>														
Base-key only	85.2 _{0.5}	56.7 _{0.3}	76.8 _{0.3}	67.5 _{1.0}	69.9 _{1.5}	70.4 _{0.2}	71.1 _{0.1}	80.3 _{0.8}	53.3 _{0.3}	48.6 _{0.1}	50.5 _{0.4}	87.2 _{0.2}	58.4 _{0.2}	63.1 _{0.2}
Group-key only	86.4 _{0.3}	59.6 _{0.3}	78.9 _{0.8}	77.3 _{2.7}	72.5 _{0.3}	72.2 _{0.1}	74.5 _{0.5}	78.4 _{0.3}	52.5 _{0.5}	49.3 _{0.3}	51.5 _{0.0}	86.7 _{0.6}	57.1 _{0.1}	62.6 _{0.1}
Task-key only	88.3 _{0.6}	58.7 _{0.2}	72.9 _{2.3}	80.7 _{0.4}	75.4 _{0.5}	74.7 _{0.1}	75.1 _{0.2}	78.5 _{0.1}	53.2 _{0.1}	48.3 _{0.1}	49.9 _{0.2}	86.9 _{0.1}	59.4 _{0.2}	62.7 _{0.1}

Table 14: Detailed ablation study of HiVe on in-domain and out-of-domain MRQA datasets.

Method	In-domain							Out-of-domain					
	Wiki	Red	XS	CNN	Giga	Media	Avg	Dialog	Sci	AESLC	WH	MN	Avg
HiVe	30.5 _{0.3}	20.6 _{0.7}	25.9 _{0.2}	24.0 _{0.5}	38.8 _{0.1}	16.7 _{2.5}	26.1 _{0.6}	17.5 _{1.6}	18.7 _{2.0}	14.4 _{0.1}	20.9 _{1.0}	12.3 _{0.6}	16.8 _{0.3}
w/o Residual Decomp.	30.7 _{0.1}	20.9 _{0.2}	26.2 _{0.4}	24.4 _{0.6}	38.6 _{0.4}	15.5 _{0.8}	26.0 _{0.3}	16.1 _{0.9}	15.1 _{0.4}	10.3 _{0.1}	17.0 _{1.4}	13.3 _{0.8}	14.4 _{0.4}
w/o Hierarchical Structure													
Base-level only	29.3 _{0.8}	20.3 _{0.5}	25.1 _{0.2}	21.7 _{0.6}	38.2 _{0.8}	17.6 _{0.4}	25.4 _{0.1}	18.9 _{0.1}	18.9 _{1.3}	13.9 _{0.5}	19.1 _{0.1}	9.2 _{1.1}	16.0 _{0.4}
Group-level only	29.7 _{0.1}	20.4 _{0.1}	24.8 _{0.6}	23.2 _{0.6}	38.2 _{0.6}	16.9 _{0.1}	25.5 _{0.1}	18.8 _{0.3}	17.7 _{1.8}	11.6 _{0.8}	19.6 _{3.0}	8.3 _{1.1}	15.2 _{1.0}
Task-level only	31.1 _{0.3}	20.8 _{0.4}	26.1 _{0.2}	24.7 _{1.2}	39.2 _{0.1}	14.8 _{1.0}	26.1 _{0.2}	17.0 _{0.5}	15.4 _{0.2}	10.1 _{1.0}	16.0 _{1.7}	14.2 _{0.1}	14.5 _{0.6}
w/o Vertical MoE													
Base-key only	17.8 _{0.6}	15.9 _{0.4}	21.0 _{1.1}	21.5 _{0.4}	28.1 _{1.3}	18.5 _{0.4}	20.5 _{0.4}	20.0 _{0.1}	18.8 _{0.6}	14.5 _{0.5}	13.0 _{0.4}	8.9 _{0.4}	15.0 _{0.1}
Group-key only	18.4 _{2.1}	17.1 _{1.8}	23.0 _{0.4}	19.7 _{1.5}	27.6 _{1.0}	19.2 _{1.4}	20.9 _{0.8}	19.1 _{0.5}	19.6 _{2.1}	14.1 _{0.6}	13.6 _{3.0}	7.7 _{1.1}	14.8 _{0.1}
Task-key only	31.1 _{0.4}	20.7 _{0.6}	26.1 _{0.2}	24.2 _{0.6}	39.0 _{0.1}	15.5 _{0.9}	26.1 _{0.3}	16.3 _{1.3}	15.3 _{0.9}	14.0 _{1.9}	21.0 _{1.3}	13.3 _{0.4}	16.0 _{0.2}

Table 15: Detailed ablation study of HiVe on in-domain and out-of-domain summarization datasets.

E.3 Efficiency Analysis

We additionally compare relative FLOPs and inference latency to evaluate the efficiency of HiVe, as shown in Table 13. Overall, HiVe achieves inference efficiency comparable to existing prompt-tuning baselines.

F Detailed Ablation Results

F.1 Per-Task Ablation Results

Tables 14 and 15 present full ablation results on MRQA and summarization benchmarks.

F.2 Effect of Residual Decomposition

Figure 6 compares the hierarchical clustering structures learned with and without residual decomposition on MRQA tasks. Both variants produce similar high-level grouping structures, indicating that the underlying semantic relationships between tasks are preserved. In contrast, the variant with residual decomposition exhibits clearer separation between task groups, suggesting that residual decomposition more effectively removes shared prompt components while preserving task-specific structure.

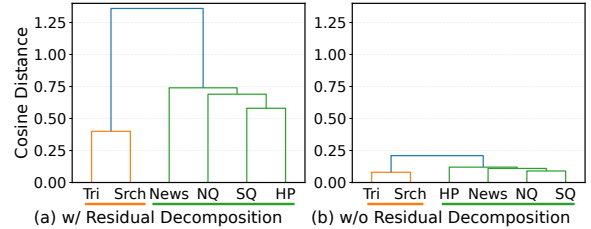


Figure 6: Hierarchical clustering structures with and without residual decomposition on MRQA tasks. Residual decomposition yields clearer task separation while preserving overall grouping structure.

F.3 Effect of Grouping Strategy

We further investigate whether the improvement comes from introducing a group level or from the proposed stability-driven grouping strategy. Table 16 compares HiVe with a two-level Base+Task variant and a Step-0 static variant, which fixes task groups using initial data representations before prompt training. Stability-driven grouping performs better than both alternatives, suggesting that grouping tasks based on stabilized task-prompt representations is more effective than either omitting the group level or fixing the grouping before inter-task relationships are learned.

Variant	In-domain	OOD
Base+Task	74.8 _{0.3}	61.9 _{0.4}
Step-0 static grouping	74.4 _{0.5}	61.6 _{0.6}
Stability-driven grouping	75.4_{0.3}	62.8_{0.3}

Table 16: Effect of grouping strategy on MRQA. The default variant is shaded in gray.

Component	Variant	In	OOD
Clustering	K-means	75.1 _{0.3}	62.8 _{0.1}
	HAC	75.4 _{0.3}	62.8 _{0.3}
Upper-level keys	Independently learned	71.7 _{0.7}	62.5 _{0.4}
	Centroid-derived	75.4_{0.3}	62.8_{0.3}
Matching metric	Negative Euclidean	74.0 _{0.5}	62.3 _{0.2}
	Cosine	75.4_{0.3}	62.8_{0.3}

Table 17: Ablation of key design choices on MRQA. Default variants are shaded in gray.

F.4 Ablation of Key Design Choices

Table 17 evaluates alternative choices for task clustering, upper-level key construction, and query-key matching on MRQA. HAC and K-means yield comparable performance, suggesting that grouping based on learned task-prompt representations is robust to the clustering algorithm. In contrast, independently learning the Base and Group keys causes most inputs to match the Base key, substantially reducing in-domain performance. Replacing cosine similarity with negative Euclidean distance also degrades both in-domain and OOD results. These findings support centroid-derived upper-level keys and cosine-based matching.