EXEMPLAR-DERIVED SOFT PROMPTS FOR EFFICIENT AND EFFECTIVE DOMAIN ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Adapting foundation models to new domains is challenging and computationally expensive. While parameter-efficient fine-tuning (PEFT) methods allow models to acquire domain-specific skills, they require updating deployed models and incur additional training and deployment overhead. In contrast, In-context Learning (ICL) avoids model updates and improves over off-the-shelf models by leveraging similar exemplars, but it often fails to achieve competitive accuracy on specialised tasks. Motivated by the success of in-context exemplars and the need for fine-tuning-level adaptation, we propose Multi-Head Attention-based Exemplar Soft Prompting (MHA-ESP), which uses an attention mechanism to learn soft prompts from retrieved exemplars, with multiple attention heads controlling prompt generation. Across multiple benchmarks and model scales, MHA-ESP performs on par with Low-Rank Adaptation (LoRA), outperforms standard ICL by an average of 18.85 points, and reduces inference cost by up to 10×GFLOPs, enabling efficient, high-accuracy domain adaptation without updating the foundation model.

1 INTRODUCTION

Fine-tuning is the standard approach for adapting foundation models to a domain by acquiring the skills needed to solve a specific task. In practice, foundation models are often adapted via Parameter-Efficient Fine-Tuning (PEFT), such as LoRA Hu et al. (2021), which updates only a small subset of the model's weights to save training cost while preserving performance Han et al. (2024). However, both full fine-tuning and PEFT have two key drawbacks. First, they require modifying the foundation model itself: in multi-tenant settings where a single backbone must serve users with distinct tasks (Tasks A, B, and C in Figure 1), the server is forced to maintain and route to a separate adapter for every user-task pair Wen & Chaudhuri (2024). Second, the upfront training cost remains high: adapting a massive model demands substantial computation resources and data, an expense that grows with model size Hu et al. (2021).

As an alternative, In-Context Learning (ICL) has emerged as a training-free "plug-and-play" adaptation strategy Xie et al. (2021); Min et al. (2022). Rather than updating model parameters through gradient descent, ICL conditions the model at inference time by providing a small set of task-specific exemplars directly in the input prompt, as shown in Figure 1. Pioneering work such as Brown et al. (2020); Wei et al. (2022) demonstrated that models exhibit strong in-context learning when presented with exemplars in natural-language form. Remarkably, in certain settings, ICL has even outperformed fine-tuning-based methods Mallen et al. (2022); Mosbach et al. (2023); Pingua et al. (2025).

However, ICL does not excel in all tasks. It performs poorly in tasks with out-of-distribution data where the model must learn a new skill Yu et al. (2024); Gupta et al. (2024). We found this to be a core limitation of ICL in our experiments as well: simply providing a few exemplars in context as text may not allow the model to acquire the required skill, especially when the skill requires reasoning beyond the data in its original training distribution.

Our goal is to develop a new exemplar-based method that has the advantages of ICL (no model modification, low upfront cost) but also performs well on skills or tasks for which the model was not trained. We investigate a new method called *Multi-Head Attention Exemplar Soft Prompt* (MHA-ESP). The idea behind it is to efficiently translate the exemplars into a soft-prompt-based representation, a format the frozen foundation model can more readily exploit. By paying a small up-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

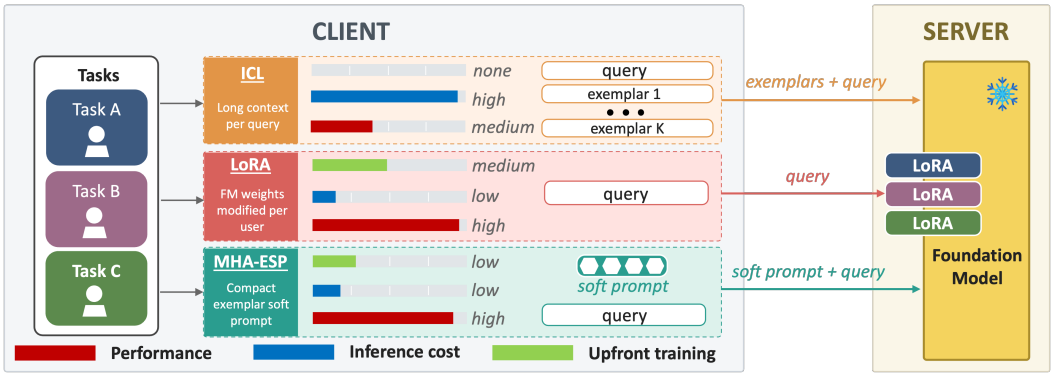


Figure 1: Comparison of domain adaptation strategies. ICL sends exemplars with query, incurring a high inference cost. LoRA modifies foundation model weights and requires per-user adapters on the SERVER side. MHA-ESP (ours) encodes exemplars into a compact soft prompt, achieving high performance with low inference cost while keeping trainable parameters locally on the USER side.

front training cost, it learns an auxiliary model that produces such task-conditioned embeddings and prepends them to the query. The approach preserves the deployment simplicity of ICL —no extra server-side parameters need to be maintained, as demonstrated in Figure 1, since the soft prompt is built before it is sent to the foundation model—and introduces only a lightweight training overhead for the soft-prompt encoder on the user side.

A key insight from our findings is that learning soft prompts from in-context exemplars is crucial to effective adaptation. In other words, simply learning a static soft prompt Liu et al. (2021); Lester et al. (2021); Liu et al. (2024) for a new task is not enough, which we find incurs a noticeable performance degradation. Across a wide range of foundation models, specialised tasks (e.g., molecular design, captioning, QA), and domains (e.g., chemistry, economics, math), our experiments show that the proposed exemplar-conditioned prompting consistently outperforms both the standard text-based retrieval and prompt-tuning baselines and closes the gap with fine-tuning baselines like LoRA. Furthermore, MHA-ESP 's design provides (i) gains in inference efficiency by reducing quadratic token overhead, (ii) stable and consistent results with the property of exemplar order-invariance embodied in the design.

In this work, we make the following contributions:

- We address fundamental limitations in ICL, which include unsatisfactory performance, high inference cost, and exemplar-order variance in domain adaptations.
- We propose an exemplar-order-invariant multi-head-attention architecture for soft-prompt generation in MHA-ESP, controlled by varying the attention heads.
- Our experiments show on-par performance with LoRA and an average 19-point gain over ICL across benchmarks and models, while cutting inference cost by $10\times$ in GFLOPs.

2 Related Works

Domain Adaptation via In-Context Learning. In-Context Learning (ICL) has emerged as the most widely used training-free approach for adapting foundation models to new domains. Instead of updating model parameters, ICL conditions the model at inference time on a small set of exemplars that demonstrate the target domain Brown et al. (2020); Xie et al. (2021); Min et al. (2022). This paradigm has demonstrated strong effectiveness across settings: in chain-of-thought (CoT) prompting where exemplars annotated with intermediate reasoning steps elicit multi-step reasoning behavior Wei et al. (2022); Kojima et al. (2022); in retrieval-augmented generation for grounding in external knowledge Lewis et al. (2020); Guu et al. (2020); Borgeaud et al. (2022); and in agent-based frameworks Yao et al. (2023); Shinn et al. (2023); Schick et al. (2023). Across these settings, ICL consistently improves over zero-shot prompting without modifying model weights.

Despite its advantages, ICL faces three fundamental limitations. First, because attention scales quadratically with sequence length Vaswani et al. (2017), appending more exemplars or retrieving context into the prompt incurs substantial inference cost. Second, ICL exhibits exemplar-order variance; permuting the input demonstrations can shift accuracy by more than 10 points on standard classification benchmarks Lu et al. (2022); Pezeshkpour & Hruschka (2023); Zheng et al. (2023). Third, ICL still underperforms on out-of-distribution or skill-acquisition domains where exemplars themselves failed to surface semantically sufficient context Yu et al. (2024); Gupta et al. (2024); Joren et al. (2024). In contrast, MHA-ESP with a small upfront training cost retains the adaptation signal of in-context exemplars but encodes them into a small, order-invariant set of soft-prompt vectors, reducing effective context length by an order of magnitude while delivering better performance than text-based ICL.

Domain Adaptation via Fine-Tuning. When training data and computational resources are available, fine-tuning typically yields the strongest domain-adaptation performance. However, full fine-tuning is prohibitive in both memory and compute Hu et al. (2021). Parameter-efficient fine-tuning methods have emerged to address this by freezing the backbone and updating a subset of parameters, generally referred to as the adapter modules Houlsby et al. (2019); Pfeiffer et al. (2021); Li & Liang (2021); He et al. (2021); Hu et al. (2021); Zhang et al. (2023). Among these, LoRA Hu et al. (2021) has become the standard for domain adaptation.

However, a practical limitation is that the adaptation resides *on the server side*: the serving infrastructure must store and load a separate adapter for each domain, leading to significant memory and management overhead when scaling to many heterogeneous domains. Recent work has therefore focused on multi-tenant LoRA serving: FLoRA Wen & Chaudhuri (2024) lets each example in a minibatch carry its own low-rank weights; S-LoRA Sheng et al. (2024) servers thousands of concurrent adapters via unified paging; dLoRA Wu et al. (2024) dynamically merges/unmerges adapters with the base model and migrates requests across replicas. These systems reduce serving costs but do not address the underlying constraint of the foundation model provider hosting the adapters. MHA-ESP addresses this by shifting adaptation to the user side, where each domain-specific request trains a lightweight attention-based module (a few million parameters) to generate soft prompts. The resulting soft prompt is prepended to each query at inference time, eliminating the need for maintaining per-domain adapters.

Domain Adaptation via Soft Prompting. Soft-prompt methods occupy a middle ground: like PEFT, they have a small training overhead, but like ICL, they operate purely at the input level, leaving model weights and the serving stack untouched. Prompt tuning Lester et al. (2021) and P-Tuning Liu et al. (2021) showed that prepending a few trainable vectors suffices to adapt very large models, making them well-suited for the multi-tenant setting Jain et al. (2024). Prior work either learns a static prompt per domain Lester et al. (2021); Liu et al. (2024), which lacks instance-specificity, or conditions soft prompts on individual queries Wu et al. (2022); Jain et al. (2024).

However, even these adaptive methods overlook the rich signal provided by similar exemplars—the key driver behind the success of ICL. MHA-ESP is explicitly designed to recover this signal directly: it retrieves exemplars relevant to the current query and learns to encode them into a query-specific soft prompt. By leveraging multi-head attention, MHA-ESP enables structured interactions between the query and retrieved exemplars, while maintaining order invariance in exemplar aggregation. In addition, the number of attention heads provides a simple and interpretable hyperparameter that controls the expressiveness of the resulting soft prompts, allowing the method to be flexibly applied across tasks and foundation models.

3 METHODOLOGY

The objective of domain adaptation is to adapt a foundation model f_θ to a new task with data $\mathbb{D} = \{(x_i, y_i)\}_{i=1}^N$ such that $y = f_\theta(x)$. It formally involves updating the parameters of the model θ to maximize the likelihood of the response y, i.e., $\max_\theta \mathbb{E}_{(x,y)\sim\mathbb{D}}[\log p_\theta(y|x)]$.

Domain adaptation with in-context exemplars. In domains where parametric updates are not feasible, either due to limited training data or computationally expensive training, adaptation can be achieved by providing a small set of domain-specific samples $\{(x_k, y_k)\}_{k=1}^K \subset \mathbb{D}$ as in-context exemplars. In this setting, the model prediction takes the form $y = f_\theta(\{(x_k, y_k)\}_{k=1}^K, x)$. This adap-

tation can be further enhanced by leveraging a relevance or similarity function $\Phi(x, x')$. Specifically, for each input x, the top-K most relevant samples in $\mathbb{D}$ are retrieved as in-context exemplars—i.e., $(x_k, y_k) = \arg\max_{(x',y')\in\mathbb{D}\setminus\{(x,y),(x_1,y_1),\ldots,(x_{k-1},y_{k-1})\}} \Phi(x, x')$, for $1 \leq k \leq K$.

Representing in-context exemplars. While approaches based on ICL typically represent exemplars in text form, we instead consider soft prompts as an alternative representation. To this end, we define an encoding function $g(\mathbb{D}_K|x) = \boldsymbol{Z}$ that embeds $\mathbb{D}_K$, the top-K in-domain samples retrieved for a given x, into a soft prompt $\boldsymbol{Z} \in \mathbb{R}^{d\times m}$, where m denotes this virtual prompt's length and d is the embedding dimension of the foundation model. Similar to prompt tuning, the learned soft prompt is prepended to the word embeddings of x, thereby enabling adaptation without updating θ.

3.1 Multi-Headed Attention Exemplar Soft Prompt

In this paper, we propose a multi-headed attention-based function for encoding exemplars into soft prompts (MHA-ESP), where each head projects the top-K in-context exemplars into the embedding space of the language model as a single soft-prompt vector $\boldsymbol{z} \in \mathcal{R}^d$. In particular, performing attention with H heads results in the soft prompt $\boldsymbol{Z}_{MHA} = [\boldsymbol{z}^{(1)}, \ldots \boldsymbol{z}^{(i)} \ldots, \boldsymbol{z}^{(H)}]$ of length $m = H$, where the output from the ith head is

$$\boldsymbol{z}^{(i)} = \textsc{Attention}(\boldsymbol{q}^i, \mathbf{K}^i, \mathbf{V}^i) = \text{softmax}(\frac{\boldsymbol{q}^i \boldsymbol{K}^i}{\sqrt{d}})\boldsymbol{V}^i;$$

with $\boldsymbol{K}^i = [\boldsymbol{k}^i_1, ..\boldsymbol{k}^i_k .., \boldsymbol{k}^i_K]$ and $\boldsymbol{V}^i = [\boldsymbol{v}^i_1, ..\boldsymbol{v}^i_k .., \boldsymbol{v}^i_K]$, where $\boldsymbol{q}^i = \boldsymbol{W}^Q_i E_x$ is the attention query corresponding to a given x, $\boldsymbol{k}^i_k = \boldsymbol{W}^K_i E_{x_k\oplus y_k}$ and $\boldsymbol{v}^i_k = \boldsymbol{W}^V_i E_{x_k\oplus y_k}$ are the keys and values corresponding to $(x_k, y_k) \in \mathbb{D}_K$, respectively. $E_x \in \mathcal{R}^{d'}$ represents the dense representation of x obtained by a sentence-embedding model with hidden dimension d'; $\oplus$ denotes the concatenation operator; and $\boldsymbol{W}^Q_i \in \mathcal{R}^{d\times d'}$, $\boldsymbol{W}^K_i \in \mathcal{R}^{d\times d'}$, and $\boldsymbol{W}^V_i \in \mathcal{R}^{d\times d'}$ represent the attention query, key, and value weights associated with head i, respectively. Overall, we optimize the following objective:

$$\max_{\varphi} \mathbb{E}_{(x,y)\sim\mathbb{D},\, \mathbb{D}_K=\Phi(x,\cdot)} \big[\log p(y|g_\varphi(\mathbb{D}_K|x), x)\big],$$

such that $g_\varphi(\mathbb{D}_K|x) = \boldsymbol{Z}_{MHA}$, where φ represents the parameters of the encoding function g.

Lower inference cost. Prior work Cheng et al. (2024) interprets the encoding function $g(\cdot)$ as a context-compression module achieving a compression ratio of $\frac{|\mathbb{D}_K|}{m}$. It reduces the tokenised length of the context $\mathbb{D}_K$ from $|\mathbb{D}_K|$ to m, thereby lowering overall inference cost during deployment. ICL without any compression yields a compression ratio of 1. In contrast, MHA-ESP achieves a higher and configurable compression ratio of $\frac{|\mathbb{D}_K|}{H}$ ($m = H$). Configurability comes from tailoring the soft-prompt size to the domain by varying the number of heads to adjust its representational capacity.

Order Invariance. We argue that the soft prompts generated by MHA-ESP are invariant to the order of retrieved exemplars. This property follows from the use of scaled dot-product attention, which depends only on the set of input tokens rather than their order Vaswani et al. (2017). Specifically, the inputs are exemplar embeddings $[E_{x_1\oplus y_1}, \ldots, E_{x_K\oplus y_K}]$ without positional encoding; therefore, permuting them does not alter the aggregated representation produced by each head. As a result, the soft prompt remains identical regardless of exemplar ordering. A formal proof is given in Appendix A.

4 Experiments, Results, and Discussion

4.1 Experimental Setup

Benchmarks. In this work, we focus on domains or tasks where foundation models exhibit weak zero-shot performance, but can benefit from in-context exemplars. Such exemplars provide cues in the form of analogies to similar samples, domain-specific formats (e.g., SMILES strings for molecules), specialized vocabulary, or templates for step-by-step reasoning. We therefore benchmark across two groups of tasks. The first group involves limited-data molecular-property-prediction tasks: (a) *BACE* [train/test split: 1,413/100]—binary classification of whether a molecule inhibits BACE1; (b) *BBBP* [train/test split: 1,950/100]—prediction of blood–brain barrier penetration (Yes/No); (c) *ClinTox* [train/test split: 1,384/100]—classification of molecules as clinically

Table 1: Comparison with ICL and LoRA across benchmarks. Hyperparameters are tuned via sweeps: MHA-ESP ($H \in \{1, 2, 4, 8\}$). Underlined indicates the best performance. **Bold** indicates the best improvement relative to LoRA for a given foundation model.

Methods	*BACE*	*BBBP*	*ClinTox*	*Design*	*Captioning*	*MMLU-Pro*	$\Delta^{\text{avg}}_{\text{LoRA}}$
Qwen3-4B							
LoRA	12.32	42.84	81.00	81.31	31.74	50.68	-
Off-the-Shelf	0.0	10.88	0.0	37.75	0.82	35.22	-42.18
ICL(fixed)	53.09	33.93	9.72	71.51	5.63	36.82	-22.82
ICL	75.87	69.33	44.24	71.66	5.67	38.24	-4.36
MHA-ESP	76.27	82.49	96.32	73.63	23.59	45.96	**13.46**
Llama3.2-3B-Instruct							
LoRA	0.0	54.77	93.06	80.63	31.82	42.09	-
Off-the-Shelf	0.0	0.0	0.0	17.38	0.0	25.45	-56.74
ICL(fixed)	48.86	36.39	36.23	65.71	4.74	28.67	-19.03
ICL	49.89	46.15	37.90	65.44	5.72	28.94	-16.69
MHA-ESP	67.62	87.61	94.44	80.73	32.79	36.43	**13.63**
Mistral-7B-Instruct							
LoRA	24.72	85.24	78.32	85.22	36.98	26.13	-
Off-the-Shelf	18.05	0.0	0.0	48.12	0.0	25.50	-52.31
ICL(fixed)	58.63	42.33	34.01	71.11	3.51	26.55	-20.93
ICL	77.65	66.62	59.79	71.11	3.53	26.48	-8.64
MHA-ESP	76.26	87.01	94.42	78.11	34.54	35.17	**9.98**

trial toxic vs. non-toxic Guo et al. (2023). The second group consists of medium-scale tasks on molecular reasoning and general language understanding: (d) *ChEBI-Design* [train/test split: 26,407/100]—generating new molecules from their descriptions and (e) *Captioning* [train/test split: 26,407/100]—generating text-based description that describes the molecule Edwards et al. (2022; 2021). (f) *MMLU-Pro* [train/test split: 6,011/6,019] – enhanced multi-task language understanding across 14 disciplines Wang et al. (2024). Additional performance comparison on the math-reasoning benchmark can be found in Appendix C.

Performance Metrics. For yes/no classification tasks, we report the *geometric mean* of the True-Positive and True-Negative rates, as it avoids the biases of accuracy or F1 under skewed label distributions. We call this quantity the "*Effective Accuracy*". For ChEBI-Design, we report Molecular ACCess System (*MACCS*), which measures similarity over molecular structural fingerprints; for Captioning, we report *BLEU-4* to measure phrase-level precision—critical for chemically accurate descriptors. For MMLU-Pro, we report *accuracy* to measure whether the model selects the correct option. To quantify improvement of a method relative to any given baseline, we compute $\Delta^{\text{avg}}_{\text{baseline}} = \sqrt[n]{\prod_{i=1}^{n}(100 + o_i^{method} - o_i^{baseline})} - 100$ where o_i is the score of the method on task i, and n is the total number of benchmark tasks.

Models. We evaluate three families of foundation models (θ) with various sizes: the *Qwen3-4B* Yang et al. (2025), *Llama-3.2-3B-Instruct*, and *Mistral-7B-Instruct-v0.3* Jiang et al. (2023). For smaller FM, refer to Appendix D. To encode in-context exemplars (E_x), we employ domain-specific embedding models like *ChemBERTa2* Ahmad et al. (2022) for property prediction tasks and *Qwen3-Embedding-0.6B* Zhang et al. (2025) for encoding natural language.

Retrieval Functions (Φ). We assume that a domain-specific retrieval function is provided. For molecular benchmarks, we retrieve the top-K relevant exemplars using Tanimoto Similarity Tanimoto (1958); Guo et al. (2023), which computes scaffold-level similarity between SMILES representations. For others, retrieval is performed via cosine similarity. Additional experimental details can be found in Appendix F.

4.2 Comparison with Domain Adaptation Baselines

Baselines. Since our goal is to achieve the performance of adapter-style methods with the scalability of in-context learning (ICL), we include *LoRA* Hu et al. (2021) as a representative PEFT baseline

Table 2: Comparison with soft-prompt-based baselines across benchmarks. Hyperparameters are tuned via sweeps: Prompt Tuning (virtual tokens $\in \{1, 5, 10\}$), MHA-ESP ($H \in \{1, 2, 4, 8\}$). Results are averaged over 3 random seeds. Underlined indicates the best performance. **Bold** indicates the best average performance across benchmark tasks.

Methods	*BACE*	*BBBP*	*ClinTox*	*Design*	*Captioning*	*MMLU-Pro*	*Average*
Qwen3-4B							
PT	0.0	11.10	0.0	39.96	1.13	35.27	14.58
IDPG	62.54	87.14	94.36	72.81	17.89	45.66	63.4
U-ESP	45.50	68.69	72.54	67.89	15.71	44.60	52.49
P-ESP	60.77	73.13	53.22	70.24	14.61	44.63	52.77
MHA-ESP	76.27	82.49	96.32	73.63	23.59	45.96	**66.38**
Llama3.2-3B-Instruct							
PT	13.61	36.89	61.88	38.24	3.64	14.12	28.06
IDPG	35.28	87.14	94.87	74.39	24.48	35.58	58.62
U-ESP	33.31	71.40	68.22	71.31	20.84	36.06	50.19
P-ESP	47.78	71.63	78.94	72.89	20.60	11.93	50.63
MHA-ESP	67.62	87.61	94.44	80.73	32.79	36.43	**66.60**
Mistral-7B-Instruct							
PT	54.97	33.97	0.0	33.88	5.10	22.34	25.04
IDPG	56.89	85.14	94.36	65.47	19.40	33.62	59.15
U-ESP	71.43	69.46	74.35	67.69	18.93	33.32	55.86
P-ESP	65.46	69.51	81.02	71.11	19.58	33.47	56.69
MHA-ESP	76.26	87.01	94.42	78.11	34.54	35.17	**67.59**

that adapts the model via server-side updates for each user request. In addition, we evaluate two representative variants of ICL: *fixed ICL*, where a shared set of exemplars is used across all queries, and *adaptive ICL*, where relevant exemplars are retrieved per query from the training set. For brevity, we refer to *adaptive ICL* simply as ICL in the remainder of the paper.

As shown in the Table 1, MHA-ESP achieves the highest overall average improvement of 12.34 over LoRA, with gains computed as the geometric mean across tasks and models. We next explain the performance gaps method by method.

Comparison with LoRA. Most of the gains of MHA-ESP over LoRA arise on limited-data benchmarks (BACE, BBBP, and ClinTox), where LoRA often overfits, collapsing to predictions dominated by a single class. On medium-scale tasks (ChEBI and MMLU-Pro), MHA-ESP does not consistently outperform LoRA but remains competitive, significantly narrowing the gap compared to ICL: while ICL incurs an average drop of -17.22 relative to LoRA, MHA-ESP reduces this gap to -2.99 (geometric mean across models and tasks). From a systems perspective, serving multiple domain adaptation requests with LoRA requires maintaining separate adapter parameters per task (e.g., $\sim$1B parameters in total for Mistral-7B across benchmarks in Table 3), whereas MHA-ESP requires no server-side parameter storage and only modest user-side overhead (less than 100M parameters on average). This makes MHA-ESP a more scalable alternative for multi-domain deployment.

Comparison with ICL. The performance improvement of MHA-ESP over ICL stems from its flexible representational capacity: by varying the number of attention heads—each with its own set of weights—the model can specialise in capturing different types of dependencies across exemplars. Figure 2 further shows that across $K = 1 \ldots 10$, ICL incurs substantially higher inference cost (in FLOPs) than MHA-ESP , due to its longer context length and the quadratic scaling of attention with respect to K (refer to Appendix G for a complete efficiency evaluation in terms of memory and inference time across tasks and models). Lastly, from Table 4, we observe that ICL also exhibits non-zero variance when the in-context exemplars are permuted. This sensitivity is expected because it concatenates exemplars as text whose order directly influences the model's input. In Appendix A.1 we found that this sensitivity is even worse for smaller-scale foundation models.

Findings. While MHA-ESP introduces a one-time, lightweight training step, it can be performed locally by each user, enabling scalable deployment across many tasks without server-side overhead as in LoRA, while improving performance where standard ICL is insufficient.

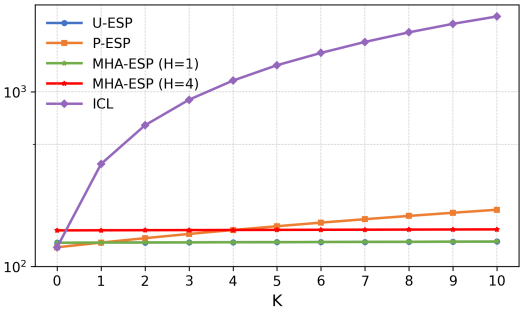


Figure 2: Inference compute in FLOPs with K.

Table 4: Order-(in)variance analysis for *Qwen3-4B* with $K = 5$: standard deviation in performance when exemplar order is randomized across 5 seeded shuffles. (Lower is better; zero means order-invariant). **Bold** shows the lowest standard deviation achieved.

Tasks	ICL	U-ESP	P-ESP	Ours
BACE	3.18	2.80	1.12	**0.0**
BBBP	2.25	3.02	1.27	**0.0**
ClinTox	8.52	0.12	4.91	**0.0**

4.3 Comparison with Soft-Prompting Baselines

Baselines. Since our method is soft-prompt-based, we benchmark against other soft-prompting approaches to identify the most effective design for domain adaptation. We first compare against standard soft-prompting baselines: (a) *Prompt Tuning (*PT *)*Lester et al. (2021), which learns a fixed, question-independent prompt shared across the entire task, and (b) *Instance-Dependent Prompt Generation (*IDPG *)*Wu et al. (2022), which conditions the soft prompt on the input question. Next, we evaluate against baselines that derive soft prompts specifically from retrieved exemplars. We adapt the architecture originally proposed in Cheng et al. (2024) to create two distinct variants: (c) *Unified-ESP (*U-ESP *)*, a many-to-one mapping where the entire set of exemplars is represented with a single "unified" soft prompt embedding, and (d) *Pointwise-ESP (*P-ESP *)*, a one-to-one mapping where each individual exemplar is encoded into its own distinct "point" or soft prompt embedding.

Table 3: Trainable parameter distribution in a client–server deployment when adapting a foundation model to all benchmark tasks using LoRA versus MHA-ESP .

Model + Adaptation via		Server-side	Client-side
Qwen3-4B	LoRA	792.72M	0
	MHA-ESP	0	53.92M
Llama3-3B	LoRA	583.56M	0
	MHA-ESP	0	69.31M
Mistral-7B	LoRA	1.0B	0
	MHA-ESP	0	87.02M

Formally, given a set of K exemplars, U-ESP is defined as: $\boldsymbol{Z} \in \mathcal{R}^{d\times 1} = \text{MLP}(E_{x_1 \oplus y_1 \oplus \ldots \oplus x_K \oplus y_K})$. In contrast, P-ESP is defined as: $\boldsymbol{Z} \in \mathcal{R}^{d\times K} = \text{MLP}(E_{x_1 \oplus y_1}) \oplus \ldots \oplus \text{MLP}(E_{x_K \oplus y_K})$. Both methods employ a single-layer MLP to project the soft prompts from the encoder's hidden space into the foundation model's embedding space. Finally, MHA-ESP utilises a many-to-many mapping to capture complex inter-exemplar dependencies while generating the final soft prompt representation. For a detailed illustration and differences between ESP-variants, refer to Appendix B.

Discussion. From Table 2, we observe that MHA-ESP consistently outperforms both PT and IDPG . This performance gap can be attributed to how soft prompts are constructed: PT learns a fixed, question-independent prompt shared across the task, while IDPG conditions its soft prompt on the input question but does not leverage related training exemplars. In contrast, MHA-ESP conditions its soft prompts on retrieved exemplars most relevant to the query, enabling better utilisation of task-specific information.

MHA-ESP also outperforms U-ESP and P-ESP across nearly all configurations. The gap relative to U-ESP likely arises from collapsing all exemplars into a single vector, leading to potentially lossy representations. While P-ESP improves over U-ESP by encoding each exemplar independently, all exemplars are processed with identical weights. In contrast, MHA-ESP assigns distinct weights to each head and computes representations via attention over the full set of exemplars, enabling head-specific representations. These differences extend to efficiency and robustness: Figure 2 reports inference compute (in FLOPs) as a function of K, using *ChemBERTa-2* as the encoder and *Qwen3-4B* as the foundation model, where MHA-ESP with a single head matches the cost of U-ESP and

Table 5: Effective accuracy of MHA-ESP against ICL and U-ESP under varying K. Results are averaged over 3 random seeds. **Bold** indicates the best performance. Underlined indicates the value of K that achieves the best effective accuracy for a given method.

Benchmarks	ICL			U-ESP			MHA-ESP		
	K=1	*K=5*	*K=10*	*K=1*	*K=5*	*K=10*	*K=1*	*K=5*	*K=10*
Qwen3-4B									
BACE	58.50	75.87	75.87	61.35	45.35	34.03	59.89	**76.27**	52.42
BBBP	67.50	69.33	68.01	73.51	68.69	70.43	72.12	**82.49**	80.43
ClinTox	30.94	44.24	54.48	47.64	72.54	65.16	63.19	**96.32**	89.44
Llama3.2-3B-Instruct									
BACE	51.71	49.89	36.36	39.77	33.31	16.29	52.37	**67.62**	54.73
BBBP	35.81	46.15	52.08	69.80	71.40	15.81	71.78	**87.61**	86.80
ClinTox	38.59	37.90	47.27	67.35	68.22	72.54	64.95	**94.44**	89.33

remains more efficient than P-ESP for head counts ¡K. Both baselines are also sensitive to exemplar ordering (Table 4): U-ESP 's joint encoding makes its representation depend on input order via encoding models's positional encodings, while P-ESP 's independent encoding determines the soft prompt's vector arrangement, producing downstream variance.

Findings. In domain adaptation tasks, MHA-ESP is the most effective soft-prompting-based method, delivering consistent performance gain across models and benchmarks.

5 Further Studies

Performance Sensitivity to Top-K Retrieval. In this section, we examine how task performance varies with the number of retrieved exemplars (K), focusing on context saturation Vladika & Matthes (2025), the point at which adding more context introduces noise and causes performance to plateau or decline. As shown in Table 5, increasing K generally improves effective accuracy for both ICL ($K = 1 \rightarrow 5 \rightarrow 10$) and MHA-ESP ($K = 1 \rightarrow 5$), reflecting the benefit of richer context. A key finding is that MHA-ESP at $K = 5$ generally outperforms ICL and U-ESP at their respective best-performing Ks, indicating that it is more effective at extracting and representing information that is sufficient to answer the questions Joren et al. (2024). Moreover, it achieves this performance with fewer FLOPs than ICL (See Figure 2). In table 5, we further observe that MHA-ESP reaches context saturation earlier than ICL: effective accuracy peaks around $K = 5$, whereas ICL generally continues to improve up to $K = 10$. Beyond these points, additional exemplars degrade effective accuracy, likely due to the inclusion of less relevant or noisy samples, consistent with findings of Liu et al. (2023); Zhao (2023); Hsieh et al. (2024). A corresponding analysis showcasing that MHA-ESP is more robust to noisy retrieval than ICL can be found in Appendix E.

Findings. MHA-ESP enables more effective domain adaptation with fewer exemplars over ICL.

Design Space of MHA-ESP . We study the design space of MHA-ESP along three axes: (1) the number of attention heads H, (2) the number of stacked multi-head attention (MHA) layers, and (3) the number of tokens used to represent each retrieved exemplar in the encoder-only language model. For (2), increasing the number of MHA layers corresponds to stacking multiple self-attention blocks in the computation of $g_\varphi(\mathbb{D}_K \mid x)$, which applies self-attention iteratively over intermediate soft-prompt representations. For (3), we adopt the sliding-window chunking strategy Hearst (1997) to obtain multiple tokens for representing a single exemplar.

We evaluate MHA-ESP on molecular-property-prediction tasks (BACE, BBBP, and ClinTox) using Mistral-7B-Instruct as the foundation model, with results summarised in Figure 3. As shown there, effective accuracy improves with an increasing number of attention heads, indicating that with more heads, available contextual information is more effectively exploited as capacity is distributed across heads. Additional results are reported in Appendix I.

Now, we fix the number of attention heads and vary both the number of tokens per exemplar and the number of stacked MHA layers. We observe that representing each exemplar with a single token

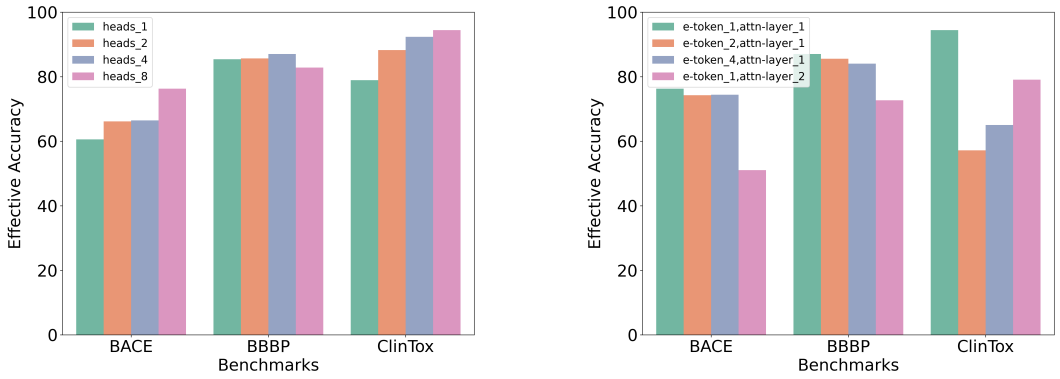


Figure 3: We fix the number of retrieved exemplars to five ($K = 5$) and use Mistral-7B-Instruct as the foundation model. The reported performance evaluates the effect of varying: (left) the number of attention heads, (right) the number of tokens pooled from exemplars (e-tokens) and the number of self-attention layers (attn-layers) in the MHA-ESP framework.

and using a single MHA layer achieves the best performance. While increasing attention depth or exemplar tokenization increases model expressiveness, we do not observe performance gain.

Findings. For the considered tasks, increasing the number of attention heads is more effective than increasing attention depth or per-exemplar tokenization.

Table 6: Comparison of Off-the-shelf, ICL, and MHA-ESP methods across three models on in-domain and out-of-domain settings. Each cell reports mean ± standard deviation across three random in-domain/out-of-domain splits. **Bold** indicates the best performance per model.

Method	**Qwen3-4B**		**Llama3.2-3B-Instruct**		**Mistral-7B-Instruct**	
	in-domain	*out-of-domain*	*in-domain*	*out-of-domain*	*in-domain*	*out-of-domain*
Off-the-shelf	35.60±2.76	38.24±2.37	25.37±2.87	27.95±2.52	25.53±3.15	28.81±2.62
ICL	37.90±2.35	40.75±1.79	28.74±3.25	31.80±2.81	26.49±2.18	29.09±1.68
MHA-ESP	**43.74±1.29**	**44.09±2.32**	**34.46±2.04**	**34.00±0.82**	**32.97±2.14**	**32.13±1.15**

Out-of-domain generalization. In this section, we test out-of-domain generalisation of learned MHA-ESP modules. To serve the purpose, we consider the QA benchmark MMLU-Pro, which spans 14 subject domains. We partition these domains into two disjoint groups: 7 domains are used to train MHA-ESP (treated as *in-domain* at test time), and the remaining 7 are held out entirely from training (*out-of-domain*). To reduce sensitivity to the particular partition, we evaluate over three random splits. In Table 6, we observe that MHA-ESP consistently outperforms ICL across all foundation models in both in-domain and out-of-domain settings. This suggests that its learned multi-headed attention generalises effectively beyond training distributions, enabling robust adaptation to unseen domains.

Refer to Appendix H and I for additional experiments.

6 Conclusion

In this paper, we proposed MHA-ESP , a soft-prompt-based method inspired by ICL, to use in-context exemplars to improve domain adaptation. MHA-ESP represents exemplars with soft prompts using an order-invariant architecture that avoids deployment overhead while maintaining low inference cost with stable performance. The number of attention heads serves as a tunable parameter, allowing efficient use of exemplar information and flexible control over soft-prompt generation. Compared to retrieval-augmented ICL and adapter-based fine-tuning methods, MHA-ESP offers a better balance between effectiveness, efficiency, and deployment-friendliness for domain adaptation.

AI USE STATEMENT

In this work, we used generative AI tools solely for editing the manuscript to improve grammar and readability. We have not used generative AI tools for any required disclosure tasks (such as model development, hypothesis generation, method implementation, or data analysis), and all other required disclosure categories are not applicable to this work. We have reviewed all AI-assisted edits for clarity and accuracy. We take full responsibility for the final content of this work, including all text, claims, and artifacts.

REPRODUCIBILITY STATEMENT

We have taken several measures to ensure the reproducibility of our theoretical and empirical findings. All core claims and theoretical guarantees, including the formal proof of exemplar order-invariance, are completely detailed in Appendix A. Full experimental settings, data splits, and model architectures are described in Section 4.1, with comprehensive hyperparameter selections and training details provided in Appendix F. All empirical results reported in our tables represent the average performance across 3 random seeds. Additionally, detailed information regarding computational hardware, execution time, and memory overhead can be found in Appendix G.

ETHICS STATEMENT

This work complies fully with the conference Code of Ethics. Our research focuses on fundamental methodology and does not involve human subjects, crowdsourcing, or personal data. As such, it presents no direct privacy, security, or safety risks, nor does it require Institutional Review Board (IRB) approval. A broader discussion regarding the potential positive and negative societal impacts of this work, along with relevant deployment considerations, is provided in Appendix J.

REFERENCES

Walid Ahmad, Elana Simon, Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta-2: Towards chemical foundation models. *arXiv preprint arXiv:2209.01712*, 2022.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pp. 2206–2240. PMLR, 2022.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Xin Cheng, Xun Wang, Xingxing Zhang, Tao Ge, Si-Qing Chen, Furu Wei, Huishuai Zhang, and Dongyan Zhao. xrag: Extreme context compression for retrieval-augmented generation with one token. *Advances in Neural Information Processing Systems*, 37:109487–109516, 2024.

Carl Edwards, ChengXiang Zhai, and Heng Ji. Text2mol: Cross-modal molecule retrieval with natural language queries. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 595–607, 2021.

Carl Edwards, Tuan Lai, Kevin Ros, Garrett Honke, Kyunghyun Cho, and Heng Ji. Translation between molecules and natural language. *arXiv preprint arXiv:2204.11817*, 2022.

Taicheng Guo, Bozhao Nan, Zhenwen Liang, Zhichun Guo, Nitesh Chawla, Olaf Wiest, Xiangliang Zhang, et al. What can large language models do in chemistry? a comprehensive benchmark on eight tasks. *Advances in neural information processing systems*, 36:59662–59688, 2023.

Shailja Gupta, Rajesh Ranjan, and Surya Narayan Singh. A comprehensive survey of retrieval-augmented generation (rag): Evolution, current landscape and future directions. *arXiv preprint arXiv:2410.12837*, 2024.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *International conference on machine learning*, pp. 3929–3938. PMLR, 2020.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. *arXiv preprint arXiv:2110.04366*, 2021.

Marti A Hearst. Text tiling: Segmenting text into multi-paragraph subtopic passages. *Computational linguistics*, 23(1):33–64, 1997.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *ICML*, 2019.

Cheng-Yu Hsieh, Yung-Sung Chuang, Chun-Liang Li, Zifeng Wang, Long T Le, Abhishek Kumar, James Glass, Alexander Ratner, Chen-Yu Lee, Ranjay Krishna, et al. Found in the middle: Calibrating positional attention bias improves long context utilization. *arXiv preprint arXiv:2406.16008*, 2024.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Abhinav Jain, Swarat Chaudhuri, Thomas Reps, and Chris Jermaine. Prompt tuning strikes back: Customizing foundation models with low-rank prompt adaptation. *Advances in Neural Information Processing Systems*, 37:47297–47316, 2024.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. arXiv preprint arXiv:2310.06825, 2023. URL `https://arxiv.org/abs/2310.06825`.

Hailey Joren, Jianyi Zhang, Chun-Sung Ferng, Da-Cheng Juan, Ankur Taly, and Cyrus Rashtchian. Sufficient context: A new lens on retrieval augmented generation systems. *arXiv preprint arXiv:2411.06037*, 2024.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *NeurIPS*, 2022.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *ACL*, 2021.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*, 2023.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*, 2021.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. Gpt understands, too. *AI Open*, 5:208–215, 2024.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8086–8098, 2022.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. *arXiv preprint arXiv:2212.10511*, 2022.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022.

Marius Mosbach, Tiago Pimentel, Shauli Ravfogel, Dietrich Klakow, and Yanai Elazar. Few-shot fine-tuning vs. in-context learning: A fair comparison and evaluation. *arXiv preprint arXiv:2305.16938*, 2023.

Pouya Pezeshkpour and Estevam Hruschka. Large language models sensitivity to the order of options in multiple-choice questions. *arXiv preprint arXiv:2308.11483*, 2023.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. Adapterfusion: Non-destructive task composition for transfer learning. In *EACL*, 2021.

Bhagyajit Pingua, Adyakanta Sahoo, Meenakshi Kandpal, Deepak Murmu, Jyotirmayee Rautaray, Rabindra Kumar Barik, and Manob Jyoti Saikia. Medical llms: fine-tuning vs. retrieval-augmented generation. *Bioengineering*, 12(7):687, 2025.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *NeurIPS*, 2023.

Jiayi Sheng, Luna Lyu, Jikai Jin, Tanglin Xia, Alex Gu, James Zou, and Pan Lu. Solving inequality proofs with large language models. *Advances in Neural Information Processing Systems*, 38, 2026.

Ying Sheng, Shiyi Cao, Dacheng Li, Coleman Hooper, Nicholas Demmel, Jeff Mo, Ronald Vermeulen, Ion Stoica, et al. S-LoRA: Serving thousands of concurrent LoRA adapters. *MLSys*, 2024.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *NeurIPS*, 2023.

Taffee T Tanimoto. An elementary mathematical theory of classification and prediction. 1958.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Juraj Vladika and Florian Matthes. On the influence of context size and model choice in retrieval-augmented generation systems. *arXiv preprint arXiv:2502.14759*, 2025.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024.

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

Yeming Wen and Swarat Chaudhuri. Batched low-rank adaptation of foundation models. In *International Conference on Learning Representations*, volume 2024, pp. 23423–23438, 2024.

Bingyang Wu, Ruidong Zhu, Zili Zhang, Peng Sun, Xuanzhe Liu, and Xin Jin. dLoRA: Dynamically orchestrating requests and adapters for LoRA LLM serving. In *OSDI*, 2024.

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Rui Hou, Yuxiao Dong, VG Vydiswaran, and Hao Ma. Idpg: An instance-dependent prompt generation method. *arXiv preprint arXiv:2204.04497*, 2022.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*, 2021.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *ICLR*, 2023.

Hao Yu, Aoran Gan, Kai Zhang, Shiwei Tong, Qi Liu, and Zhaofeng Liu. Evaluation of retrieval-augmented generation: A survey. In *CCF Conference on Big Data*, pp. 102–120. Springer, 2024.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. AdaLoRA: Adaptive budget allocation for parameter-efficient fine-tuning. *ICLR*, 2023.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

Jiachen Zhao. In-context exemplars as clues to retrieving from large associative memory. *arXiv preprint arXiv:2311.03498*, 2023.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. Large language models are not robust multiple choice selectors. *arXiv preprint arXiv:2309.03882*, 2023.

A EXEMPLAR-ORDER INVARIANCE

In this section, we argue that the soft prompt generated by MHA-ESP is invariant to the order of in-context exemplars/documents.

Claim 1 (Exemplar-Order Invariance)**.** *Let $\boldsymbol{q}$ be a query vector for a given x and for $k = 1 \dots K$, let $\boldsymbol{k}_k \in \mathcal{R}^d$ and $\boldsymbol{v}_k \in \mathcal{R}^d$ be keys and value vectors derived from each retrieved example. Define the attention weights as $\alpha_k \coloneqq \frac{e^{s_k}}{\sum_{j=1}^{K} e^{s_j}}$ where score, $s_k \coloneqq \frac{\boldsymbol{q}\cdot\boldsymbol{k}_k}{\sqrt{d}}$. The attention output for each head is given by $\boldsymbol{z} = \sum_{k=1}^{K} \alpha_k \boldsymbol{v}_k$. Then $\boldsymbol{z}$ is invariant under any permutation of the retrieved examples: permuting the indexing of the pairs $\{(\boldsymbol{k}_k, \boldsymbol{v}_k)\}_{k=1}^{K}$ does not change $\boldsymbol{z}$.*

Proof. Let π be a permutation of $1 \dots K$ and consider the permuted sequence $(\boldsymbol{k}_{\pi(1)}, \boldsymbol{v}_{\pi(1)}), \dots, (\boldsymbol{k}_{\pi(K)}, \boldsymbol{v}_{\pi(K)})$. For the permuted sequence, the scores and weights are

$$s'_k = \frac{\boldsymbol{q} \cdot \boldsymbol{k}_{\pi(k)}}{\sqrt{d}} = s_{\pi(k)} \qquad \alpha'_k = \frac{e^{s'_k}}{\sum_{j=1}^{K} e^{s'_j}} = \frac{e^{s_{\pi(k)}}}{\sum_{j=1}^{K} e^{s_{\pi(j)}}}$$

Because $\{s_{\pi(j)} : j = 1, \dots, K\}$ is a reordering of $\{s_j : j = 1, \dots, K\}$, we have $\sum_{j=1}^{K} e^{s_{\pi(j)}} = \sum_{j=1}^{K} e^{s_j}$. Therefore, $\alpha'_k = \alpha_{\pi(k)}$.

Now, the head's resulting output under the permuted attention is $\boldsymbol{z}' = \sum_{k=1}^{K} \alpha'_k \boldsymbol{v}_{\pi(k)} = \sum_{k=1}^{K} \alpha_{\pi(k)} \boldsymbol{v}_{\pi(k)}$. Without loss of generality, reindex the sum by setting $j = \pi(k)$. Because π is a bijection, $k \mapsto j$ permutes the index set $\{1, \dots, K\}$, resulting in $\boldsymbol{z}' = \sum_{j=1}^{K} \alpha_j \boldsymbol{v}_j = \boldsymbol{z}$.

Thus $\boldsymbol{z}$ is unchanged by the permutation π. The result holds for any permutation, so the MHA-ESP 's per-head output is order invariant. □

A.1 ADDITIONAL RESULTS

Table 7: Order-(in)variance analysis for *Qwen3-0.6B* with $K = 5$: standard deviation in performance when exemplar order is randomized across 5 seeded shuffles. (Lower is better; zero means order-invariant). **Bold** shows the lowest standard deviation achieved.

Benchmarks	ICL	U-ESP	P-ESP	Ours
BACE	5.36	1.09	1.08	**0.0**
BBBP	2.83	1.23	2.66	**0.0**
ClinTox	29.31	6.34	3.90	**0.0**

We observe that, compared to Table 4, Qwen3-0.6B with ICL in Table 7 is much more sensitive to the order of exemplars, leading to greater performance degradation. In comparison, MHA-ESP , by design, is order-invariant and independent of model scale.

B ILLUSTRATION: EXEMPLAR SOFT PROMPT

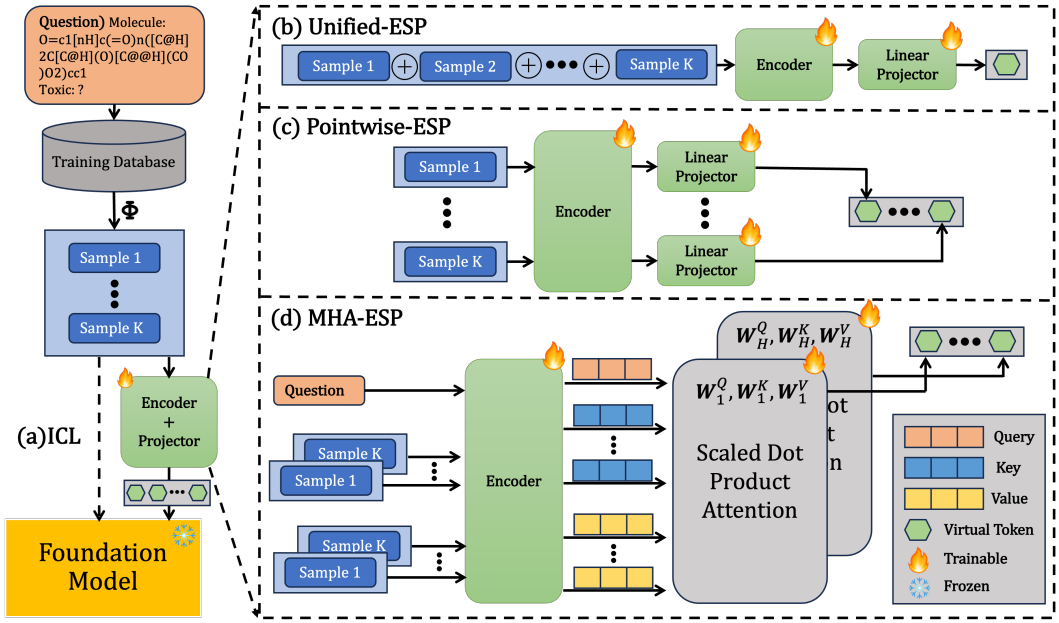


Figure 4: Comparison of exemplar-driven domain-adaptation methods: (a) ICL uses retrieved exemplars directly as context, while (b) U-ESP , (c) P-ESP , and (d) MHA-ESP derive soft-prompt representations from exemplars. The figure illustrates how each method forms its representation (single vector in U-ESP , K vectors in P-ESP , multi-head representations in MHA-ESP) and highlights their trainable components. Among the four methods, only MHA-ESP —due to its use of scaled dot-product attention—is invariant to the order of exemplars.

While P-ESP and MHA-ESP share the idea of representing retrieved exemplars as multiple embeddings, MHA-ESP introduces three fundamental differences:

- **Joint representation vs. independent encoding.** P-ESP encodes each exemplar independently via a shared MLP, and the resulting embeddings are concatenated. In contrast, MHA-ESP produces each soft prompt slot via an attention head that jointly attends over all retrieved exemplars, enabling each head to extract query-relevant information (analogous to multi-head attention in Transformers), rather than treating exemplars in isolation. This distinction is key to the improved domain adaptation performance observed in Table 2.
- **Flexible prompt capacity.** In P-ESP , the number of retrieved exemplars (K) directly determines the number of soft prompt embeddings. MHA-ESP decouples these factors: the number of attention heads controls the soft prompt size, making it a tunable hyperparameter independent of K and offering greater flexibility in balancing performance and efficiency.
- **Order invariance vs. order sensitivity.** P-ESP preserves exemplar order when constructing the soft prompt, leading to sensitivity to exemplar ordering (as evidenced in Table 4). In contrast, MHA-ESP aggregates information through attention, making it inherently order-invariant by design and thus more robust.

C BENCHMARK: INEQUALITY MATH

Table 8: Baseline comparison on *IneqMath* requiring step-by-step reasoning (with $K = 2$). We report the Final Accuracy, measured as the exact match between derived numerical values and ground truth. Underline indicates the best performance.

Accuracy	Off-the-Shelf	w ICL	w LoRA	w MHA-ESP
Llama3.2-1B-Instruct				
Bound	14.0	4.0	12.0	12.0
Relation	12.0	16.0	26.0	20.0
Final	13.0	10.0	19.0	16.0
Llama3.2-3B-Instruct				
Bound	10.00	10.0	12.0	14.0
Relation	22.00	24.0	26.0	22.0
Final	16.00	17.0	19.0	18.0

In this section, we investigate whether MHA-ESP can improve performance on mathematical-reasoning tasks. Specifically, we evaluate on *Inequality Math* [train/dev split: 1252/100], an Olympiad-level benchmark that requires proving bounds-preserving inequalities and establishing relations between algebraic expressions Sheng et al. (2026). For retrieval, we compute dense representations of question-exemplar pairs using *Qwen3-Embedding-8B* Zhang et al. (2025), and rank candidates by cosine similarity.

We primarily report results for *Llama3.2* family, because models in the Qwen3 family already achieve high off-the-shelf accuracy (e.g., *Qwen3-4B* at 65%), and adding retrieved exemplars led to performance drops. This observation is consistent with Sheng et al. (2026), which found that only certain model families benefit from in-domain exemplars. A plausible explanation is that high-performing models may have already been exposed to mathematical-reasoning tasks during pre-training, reducing the utility of retrieval.

From Table 8, we observe that MHA-ESP yields improvements over both off-the-shelf and ICL. A possible explanation is that the *Inequality Math* dataset contains multiple problems that rely on the same theorems or follow similar reasoning steps. If retrieval fails to surface such structurally related exemplars, the model cannot fully benefit from exemplar conditioning. Future work could therefore focus on improving retrieval quality—e.g., by incorporating reasoning-aware similarity metrics—which may allow MHA-ESP to better exploit shared problem structure and yield stronger overall gains.

D PERFORMANCE BENCHMARKING ON SMALL-SCALE FOUNDATION MODEL

In this section, we compare the performance of Qwen3-0.6B with the proposed MHA-ESP against baselines on our set of benchmark tasks.

Table 9: Baseline comparison with domain adaptation methods, including soft-prompt based. Hyperparameters are tuned via sweeps: Prompt Tuning (virtual tokens $\in \{1, 5, 10\}$), LoRA (rank $\in \{16, 32, 64\}$), and MHA-ESP ($H \in \{1, 2, 4, 8\}$). **Bold** indicates the best effective accuracy.

Methods	Property Prediction			CHEBI		Avg.
	BACE	*BBBP*	*ClinTox*	*Design*	*Captioning*	
Off-the-Shelf	36.67	22.21	0.0	32.20	0.34	18.28
w ICL	**76.01**	66.62	53.30	61.47	6.50	52.78
w PT	0.0	0.0	0.0	0.44	0.0	0.09
w IDPG	53.75	86.52	94.87	60.74	2.11	59.60
w LoRA	0.0	0.0	0.0	**66.50**	**20.38**	17.38
w U-ESP	40.95	68.81	31.39	29.34	4.68	35.03
w P-ESP	52.27	70.92	35.70	35.84	4.57	39.86
w MHA-ESP	75.08	**87.82**	**97.12**	58.39	15.22	**66.73**

E ROBUSTNESS TO NOISY AND ADVERSARIAL EXEMPLARS

Table 10: Robustness comparison under random exemplar replacement. We report performance averaged across 3 seeded runs with Top-K retrieval, Random-K exemplar replacement, and the corresponding performance drop (Δ) for ICL and the proposed MHA-ESP across three benchmark tasks and two foundation models.

Base Model	Task	ICL			MHA-ESP		
		Top-K	Random-K	Δ	Top-K	Random-K	Δ
Qwen3-4B	BACE	75.87	54.37	-21.50	76.27	58.31	-17.96
	BBBP	69.33	59.36	-9.97	82.49	68.26	-14.23
	ClinTox	44.24	0.0	-44.24	96.32	83.22	-13.10
Llama3.2-3B	BACE	49.89	52.45	+2.56	67.62	52.49	-15.13
	BBBP	46.15	50.74	-4.59	87.61	64.55	-23.06
	ClinTox	37.90	39.40	-1.50	94.44	88.97	-5.47

To assess robustness to noisy exemplars, we conducted an experiment where exemplars were randomly sampled from the database instead of selecting the top-5 most relevant ones. We observed that both ICL and MHA-ESP experience a performance drop in this setting, particularly with Qwen3-4B; however, MHA-ESP still outperforms ICL after random selection. This indicates that (1) exemplar relevance plays a critical role in effective question answering, and (2) MHA-ESP is better at representing and leveraging even randomly sampled (i.e., noisy) exemplars than ICL.

Interestingly, for Llama3.2-3B-Instruct, ICL shows a slight improvement when using random exemplars. Overall, MHA-ESP continues to outperform ICL under both random and relevant exemplar settings, further supporting its superior representational capability for skill acquisition in new domains and tasks.

F Additional Details: Experimental Setup

F.1 Optimising number of heads in MHA-ESP

Table 11: Choice of hyper-parameter m i.e. number of heads used by MHA-ESP for experiments conducted in Table 1, 2

Base Models	BACE	BBBP	ClinTox	Design	Captioning	MMLU-Pro
Qwen3-0.6B	4	8	8	8	8	2
Qwen3-4B	2	8	4	8	8	4
Llama3.2-3B-Instruct	2	8	8	8	8	8
Mistral-7B-Instruct	8	4	8	8	8	8

F.2 Optimising ICL

In this section, we consider several key design dimensions to optimise ICL performance in our benchmark tasks.

Table 12: Performance variance of Qwen3-4B with ICL ($K = 5$) with respect to prompt engineering and exemplar order

Design Dimension	BACE	BBBP	ClinTox
Prompt Engineering	77.0±1.0	66.0±1.0	44.0±0.0
Exemplar Order	70.0±3.2	70.0±2.3	41.1±8.5

Prompt engineering. We evaluate Qwen3-4B (K=5) with ICL using five diverse prompts across three tasks, varying in reasoning style and domain specificity (e.g., biology-grounded explanations vs. rule-based cheminformatics reasoning), while targeting the same objective. We observe that prompt choice has a negligible impact on performance, suggesting that our results are not sensitive to prompt design.

Exemplar ordering. Table 4 reports exemplar-order variance for ICL, with the corresponding averages for Qwen3-4B (K=5) reported here. While favourable orderings can yield modest gains, the improvements remain significantly smaller than those achieved by MHA-ESP . Moreover, we emphasize that sensitivity to exemplar ordering is itself a limitation of standard ICL, which MHA-ESP explicitly addresses through order-invariant aggregation.

Text truncation/compression strategies. Prior hard-prompt compression methods (e.g., EXIT, COCOM) operate by pruning irrelevant content from retrieved text. In our setting, the retrieved context consists of relevant input–output exemplars that do not contain redundant text to be pruned. An analogous filtering step is already performed by the retrieval function, which selects the top-K most relevant exemplars. We study the effect of context size on ICL via ablation over K in Table 5.

Retrieval quality. To assess the impact of retrieval, refer to the experiment in Appendix E where exemplars are randomly sampled. Both ICL and MHA-ESP exhibit performance drops (up to 44.24), confirming that retrieval quality is critical and that our retrieval setup is meaningful and non-trivial.

F.3 Training Details

Training is done for 10 epochs on the limited-data molecular benchmarks (*BACE*, *BBBP*, *ClinTox*), 4 epochs (*ChEBI-Design* and *ChEBI-Captioning*) and 2 epochs (*MMLU-Pro*) on the medium-scale benchmarks. Learning rates are selected from $\{1e-5, 3e-5, 5e-4\}$. The execution platform used 8× V100 GPUs (32GB VRAM each). The embedding models are fine-tuned with LoRA (rank 64).

F.4 OOD Generalisation

For all baselines, we evaluate per-domain accuracy and average across the 7 in-domain and 7 out-of-domain categories separately. MHA-ESP is trained with K=5 retrieved exemplars, m'=8, learning

rate $3e{-}5$, for 2 epochs. We report results on three backbones: Qwen3-4B, Llama3.2-3B, and Mistral-7B. The in-domain training areas across the three splits considered in Table 6 are:

- **Split 1:** law, other, economics, health, psychology, business, history
- **Split 2:** math, physics, chemistry, engineering, biology, computer science, philosophy
- **Split 3:** math, chemistry, law, engineering, other, economics, history

G Deployment Efficiency: Inference Runtime & Memory Footprint

Table 13: Comparison of inference efficiency—measured by average inference time (ms) and peak memory consumption (MB)—when deploying *Llama3.2-3B-Instruct*, *Qwen3-4B* and *Mistral-7b-Instruct* with different domain adaptation baselines on benchmark tasks. For MHA-ESP , we report the metrics corresponding to the best $m \in \{1, 2, 4, 8\}$.

Task	Metrics	Off-the-Shelf	w ICL	w U-ESP	w P-ESP	MHA-ESP
			Qwen3-4B			
BACE	Memory (MB)	7822.7	8041.2	7832.4	7834.2	7880.4
	Time (ms)	0.0551	0.0939	0.0511	0.0522	0.0536
BBBP	Memory (MB)	7807.9	8109.8	7817.1	7818.4	7865.9
	Time (ms)	0.0485	0.0872	0.0490	0.0524	0.0497
ClinTox	Memory (MB)	7883.6	8492.4	7893.4	7895.0	7941.5
	Time (ms)	0.0535	0.1046	0.0543	0.0584	0.0580
Chebi-captioning	Memory (MB)	7919.5	12223.0	10887.4	10413.3	10528.4
	Time (ms)	0.8571	0.8800	0.8538	0.9610	0.8581
Chebi-design	Memory (MB)	7874.3	11728.3	10894.0	10415.6	10530.8
	Time (ms)	1.8095	1.4451	1.3964	1.6220	1.1898
			Llama3.2-3B-Instruct			
BACE	Memory (MB)	6206.0	6365.8	6215.2	6216.8	6268.67
	Time (ms)	0.0411	0.0708	0.0426	0.0433	0.0429
BBBP	Memory (MB)	6196.8	6398.2	6206.2	6207.0	6259.2
	Time (ms)	0.0386	0.0684	0.0402	0.0430	0.0443
ClinTox	Memory (MB)	6247.9	6692.8	6257.5	6259.1	6312.7
	Time (ms)	0.0436	0.0744	0.0442	0.0459	0.0443
Chebi-captioning	Memory (MB)	6278.0	9439.4	9290.4	8815.5	8953.7
	Time (ms)	0.3683	0.5491	0.5563	0.5597	0.6348
Chebi-design	Memory (MB)	6259.2	9084.5	9296.9	8818.0	8956.1
	Time (ms)	0.9929	1.0450	1.0830	0.8731	0.7681
			Mistral-7B-Instruct			
BACE	Memory (MB)	13925.1	14128.3	13935.9	13938.0	14007.0
	Time (ms)	0.2455	0.1523	0.0746	0.0805	0.0801
BBBP	Memory (MB)	13910.3	14226.2	13921.2	13922.0	13990.8
	Time (ms)	0.2315	0.1535	0.0678	0.0719	0.0694
ClinTox	Memory (MB)	13999.9	14624.3	14010.9	14012.7	14084.0
	Time (ms)	0.2526	0.1720	0.0793	0.0823	0.0883
Chebi-captioning	Memory (MB)	14028.6	19011.2	16988.6	16513.2	16697.4
	Time (ms)	0.7827	0.8702	0.6394	0.6409	0.6593
Chebi-design	Memory (MB)	13964.5	18492.8	16995.1	16515.4	16699.6
	Time (ms)	1.2191	1.3008	1.0964	0.9834	0.9061

In this section, we compare the inference efficiency of a base model adapted using different training methods, focusing on average inference latency (ms) and peak GPU memory consumption (GB) measured on identical hardware (Table 13). All measurements are obtained with $K = 5$ for all retrieval-based baselines.

Across this setting, we observe that representing retrieved exemplars via MHA-ESP consistently yields lower inference latency and reduced GPU memory usage compared to standard ICL, a trend that holds across varying numbers of attention heads. Furthermore, MHA-ESP exhibits a comparable inference computational footprint to U-ESP and U-ESP , while achieving superior predictive performance (see Table 2).

H EXPERIMENT: DOES ADDING EXEMPLARS BACK HELP?

Table 14: Effect of re-inserting top-c exemplars in textual form (up to $c = 5$, given a fixed inference budget) into the in-context prompt while still using top-K for soft-prompt computation in MHA-ESP . The $K = 5$, $c = 0$ column corresponds to the MHA-ESP column of Table 1. **Bold** indicates the best effective accuracy. Underlined indicates the value of c that achieves the best effective accuracy for a given K.

Benchmarks	*K=5*			*K=10*		
	c=0	*c=1*	*c=5*	*c=0*	*c=1*	*c=5*
Qwen3-4B						
BACE	76.27	75.27	79.03	52.42	76.27	**79.19**
BBBP	82.49	68.56	**88.14**	80.43	70.01	87.5
ClinTox	96.32	76.63	94.55	89.44	70.72	**99.04**
Llama3.2-3B-Instruct						
BACE	67.62	**79.96**	78.82	54.73	74.23	77.75
BBBP	87.61	88.99	**89.74**	86.80	85.44	86.94
ClinTox	94.44	88.97	**98.40**	89.33	94.49	88.97

To investigate whether performance can be further improved, we augment MHA-ESP 's soft prompts with a small number of exemplars directly included in the model's context at inference. Specifically, we add the top-c retrieved exemplars alongside the soft prompts and study the effect across different K values.

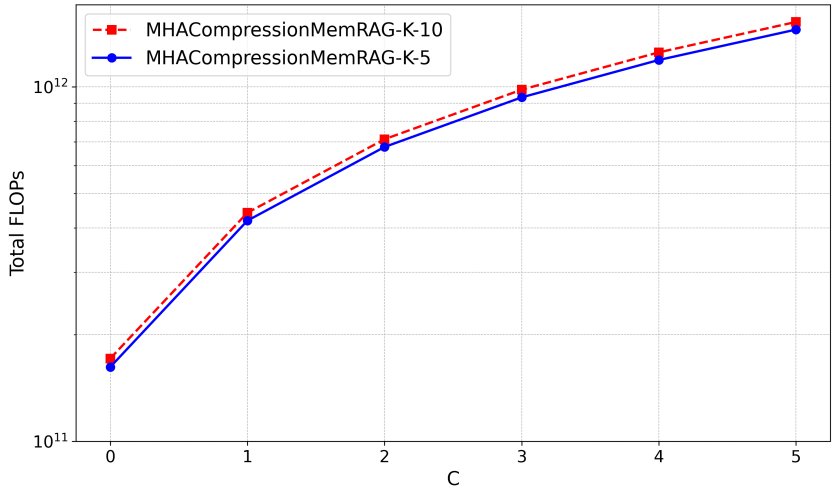
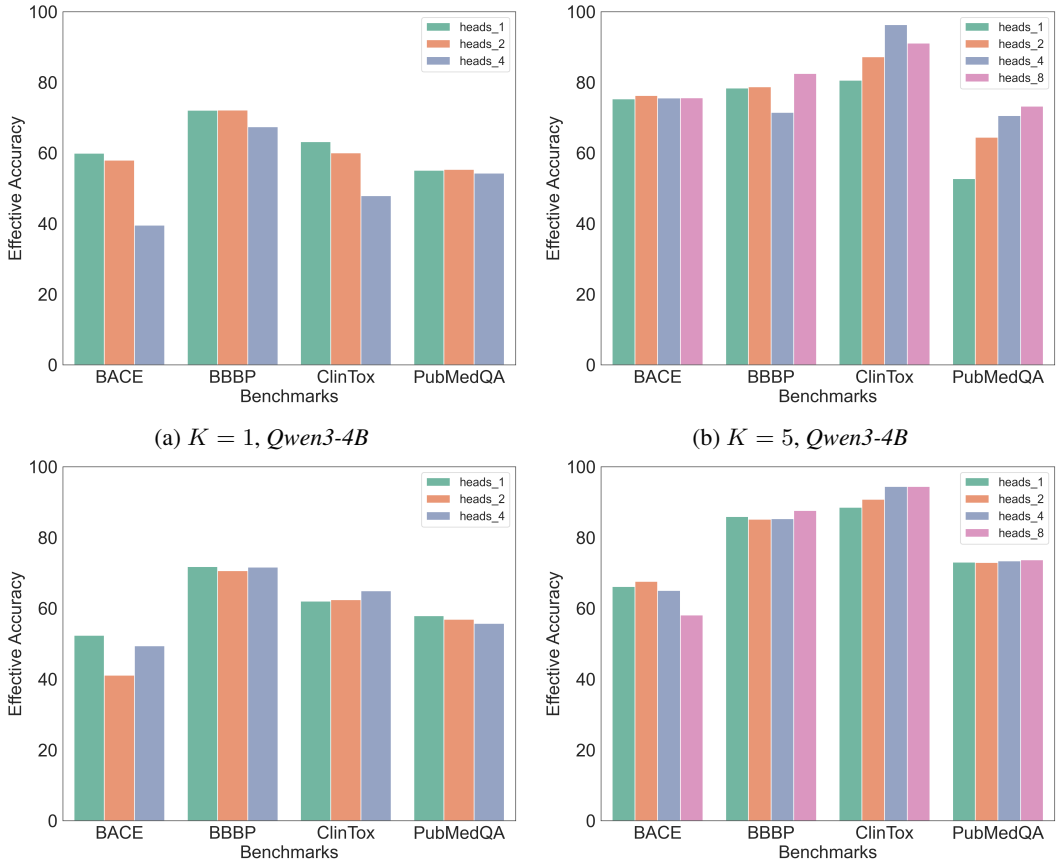


Figure 5: Total FLOPs for inference with encoder ChemBERTa-10M-MTR and foundation model Qwen3-4B using an increasing number of exemplars c in the context, given $K = 5$ and $K = 10$ as input to create a soft prompt.

As shown in Table 14, effective accuracy improves consistently with increasing c, with the best results achieved at $c = 5$ for both $K = 5$ and $K = 10$. The FLOPs analysis in Figure 5 reveals that increasing K from 5 to 10 incurs only a minor cost, while increasing c from 0 to 5 leads to a logarithmic rise in FLOPs. These results suggest that c can serve as an additional tunable knob for balancing accuracy gains against inference cost, allowing practitioners to adapt MHA-ESP to different computational budgets.

I EXPERIMENT: ADDITIONAL ABLATION WITH NUMBER OF HEADS



(a) $K = 1$, *Qwen3-4B* (b) $K = 5$, *Qwen3-4B*

(c) $K = 1$, *Llama3.2-3B-Instruct* (d) $K = 5$, *Llama3.2-3B-Instruct*

Figure 6: Varying number of heads in MHA-ESP . Performance averaged across 3 seeded runs.

We further study the impact of the number of heads. From Figure 6, we observe that when $K = 1$, increasing the number of heads does not improve effective accuracy, likely due to the limited context—only a single exemplar or document—offering little room for multiple heads to learn diverse representations. In contrast, at $K = 5$, increasing the number of heads generally leads to higher effective accuracy across benchmarks, suggesting that the benefit of multiple heads emerges only when sufficient context is available.

Findings. MHA-ESP can effectively exploit additional context with an increasing number of heads.

J BROADER IMPACT

The significance of our work lies in its potential to reduce training and maintenance costs associated with hosting and adapting foundation models to multiple domains. Furthermore, our method enhances user privacy by enabling domain-specific customisation on the user end rather than the server end.