SAHR: SEMANTICS-AWARE HEAD RECONFIGURATION FOR EFFICIENT TASK ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language models (VLMs) such as CLIP have shown strong transferability across downstream tasks. However, efficiently adapting CLIP under limited supervision remains challenging because existing parameter-efficient fine-tuning (PEFT) methods typically modify prompts, lightweight modules, or support features without explicitly identifying which internal visual Transformer units are most relevant to a target task. In this work, we observe that attention heads in the CLIP visual Transformer exhibit semantic specialization, with different heads consistently capturing different visual or semantic attributes. Accordingly, we propose Semantics-Aware Head Reconfiguration (SAHR), an efficient framework that directly updates task-relevant attention heads during CLIP adaptation. SAHR constructs text-aligned semantic profiles for attention heads and quantifies their semantic correspondence to downstream tasks through task-aware descriptions. Based on this head-task correspondence, SAHR adapts only a sparse set of top-ranked heads. Extensive experiments on 11 few-shot classification benchmarks, Flickr30K and MSCOCO image-text retrieval, and OOD robustness show that SAHR outperforms recent PEFT methods with low parameter scale. In the 16-shot setting, SAHR achieves 83.1% average accuracy on 11 benchmarks with only 1.18M parameters, outperforming the strongest PEFT method while reducing parameters by 94.6%.

1 INTRODUCTION

Vision-language models (VLMs) have achieved strong performance across tasks such as image recognition Zhai et al. (2023), image-text retrieval Cui et al. (2024), and zero-shot transfer Zhang et al. (2024). In particular, contrastive VLMs such as Contrastive Language-Image Pre-training (CLIP) Radford et al. (2021) have demonstrated particularly strong transferable representations and impressive zero-shot generalization on a wide range of downstream visual recognition tasks. Nevertheless, effective adaptation of CLIP remains crucial when the target domain involves fine-grained visual distinctions, noticeable distribution shifts, or limited labeled data. Full fine-tuning is computationally expensive and prone to overfitting in the low-data regime, which has motivated extensive research on parameter-efficient fine-tuning (PEFT) for CLIP adaptation.

Existing PEFT methods for CLIP mainly improve adaptation through prompt learning Zhou et al. (2022b;a), lightweight parameter adaptation Gao et al. (2024); Wu et al. (2024), or cache-based matching Zhang et al. (2022). Prompt-based methods learn task-specific contexts or prompts to better align class texts with visual features. Adapters and LoRA-based methods update a small subset of model parameters to improve downstream transfer while preserving efficiency. Cache-based methods exploit support-set prototypes or feature memories for low-cost adaptation. However, these methods still lack fine-grained and interpretable control over the internal visual Transformer, which often makes it difficult to achieve a strong balance between adaptation accuracy and efficiency.

Meanwhile, recent interpretability studies Gandelsman et al. (2024); Balasubramanian et al. (2024) suggest that representations inside vision-language Transformers are highly structured rather than homogeneous. As illustrated in Figure 1, attention heads in the later layers of CLIP exhibit fine-grained functional specialization, where different heads encode complementary visual concepts, ranging from local details and textures to broader scene context and color semantics. Across model depth, the encoded information tends to evolve from lower-level visual patterns to more task-relevant

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.

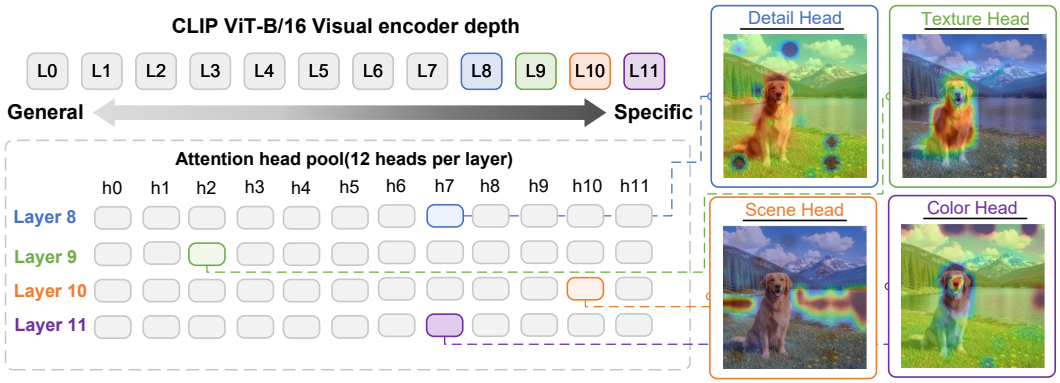

Figure 1: Semantic specialization across late attention heads in CLIP ViT-B/16. Representative attention maps and text-aligned profiles suggest that different heads capture complementary cues, ranging from local detail and texture to scene context and color, motivating selective rather than uniform adaptation.

semantics, while different heads within these layers may further specialize in distinct visual or semantic attributes. However, existing studies have mainly used these findings for interpretation, retrieval, grounding, or model analysis. They do not explicitly quantify which attention heads are semantically aligned with a given downstream task, nor do they use such head-task correspondence as an adaptation signal. These observations motivate using head-level semantic specialization to select task-relevant units for adaptation.

In this work, we propose Semantics-Aware Head Reconfiguration (SAHR), a parameter-efficient framework that quantifies the semantic correspondence between attention heads and the target task, and selectively adapts the most task-aligned heads for CLIP adaptation. Specifically, SAHR first builds semantic profiles for attention heads using text-based head descriptions, and then constructs task-aware semantic cues with the help of LLM-generated discriminative image prompts. By measuring semantic similarity between head profiles and task semantics, SAHR scores and ranks attention heads, and selects a sparse set of top-ranked heads for adaptation. It then performs selective tuning by updating only the chosen heads while freezing the remaining parameters. In this way, SAHR turns interpretable head semantics into an explicit control signal for parameter-efficient adaptation. Extensive experiments on few-shot classification and image-text retrieval benchmarks show that SAHR outperforms competitive baselines while tuning only a sparse set of attention-head parameters, yielding a favorable balance between effectiveness and efficiency. SAHR also maintains competitive performance under distribution shifts and across different CLIP backbones.

In summary, our contributions are as follows:

- We propose SAHR, a parameter-efficient framework for semantics-aware reconfiguration of attention heads, which turns interpretable head semantics into an explicit adaptation signal.
- We develop a task-aware head adaptation mechanism that combines text-based head descriptions with LLM-generated discriminative semantic cues to automatically score, rank, and selectively fine-tune a sparse set of semantically important attention heads.
- Extensive experiments on 11 few-shot classification and image-text retrieval benchmarks show that SAHR achieves a favorable accuracy–parameter trade-off with a small number of trainable parameters, yielding a strong trade-off between adaptation accuracy and efficiency.

2 Related Work

Parameter-Efficient CLIP Adaptation As large-scale pretrained models such as CLIP become increasingly transferable, adapting them efficiently under limited supervision has become an important research direction. Full fine-tuning is often computationally expensive and can easily overfit

in few-shot settings, which has motivated a broad range of parameter-efficient adaptation methods. Existing methods can be broadly grouped into three categories. Prompt learning methods, such as CoOp Zhou et al. (2022b), CoCoOp Zhou et al. (2022a), and MaPLe Khattak et al. (2023), optimize textual or visual prompts to better align downstream classes with CLIP representations. Adapter-based and LoRA-style methods Mohammad et al. (2026) improve transfer by inserting lightweight trainable modules or updating a small subset of parameters while keeping most of the backbone frozen. Cache-based methods, such as Tip-Adapter-F Zhang et al. (2022), and TaskRes Yu et al. (2023), exploit support-set prototypes or feature memories for efficient few-shot matching. However, these methods still provide limited interpretability into which internal visual units are most useful for downstream adaptation, and offer little fine-grained control over the visual Transformer. As a result, they often struggle to achieve precise head-level adaptation, leading to a weaker trade-off between adaptation performance and efficiency.

LLM-Generated Semantic Descriptions Recent work Roth et al. (2023); Menon & Vondrick (2022) has shown that large language models can provide richer and more discriminative textual descriptions than hand-crafted templates for vision-language adaptation. Such descriptions have been used to improve zero-shot classification, refine class semantics, or enhance textual representations under training-free and lightly supervised settings. In particular, methods such as CuPL Pratt et al. (2023) and related prompt-generation approaches demonstrate that LLM-generated descriptions can better capture fine-grained visual cues than simple class names. However, existing studies mainly use these descriptions to directly improve text prompts or classification scores. By contrast, we use LLM-generated semantic descriptions as task-aware cues for identifying and ranking semantically important attention heads inside CLIP.

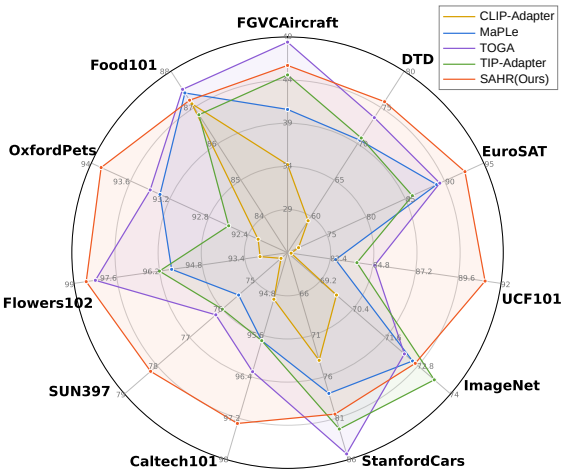

Figure 2: Comparison of 16-shot classification accuracy on 11 benchmark datasets. Our method achieves the best average performance among the compared PEFT methods while tuning only 1.18M parameters.

Interpreting Vision-Language Transformers A growing body of work studies the internal mechanisms of vision-language Transformers through interpretability and decomposition analyses. In CLIP and related models, recent studies have shown that intermediate components, including attention heads and residual units, often exhibit structured and interpretable roles. Some works Gandelsman et al. (2024); Balasubramanian et al. (2024); Gentile et al. (2026) characterize such roles through text-aligned descriptions, revealing specialization patterns associated with attributes such as shape, texture, location, or semantic categories. Others Madasu et al. (2025); Lin et al. (2025); Bousselham et al. (2025) use these findings for representation analysis, retrieval, grounding, or model understanding. While these studies provide strong evidence that Transformer representations are not homogeneous, they are mostly aimed at interpretation rather than adaptation. Our work builds on this line of research, but differs in that it turns interpretable head semantics into an explicit control signal for parameter-efficient adaptation.

3 METHOD

3.1 SEMANTIC HEAD PROFILING

We build SAHR on CLIP, which learns an image encoder $f_I(\cdot)$ and a text encoder $f_T(\cdot)$ aligned in a shared vision-language embedding space. Given an image I and a text description t, CLIP measures

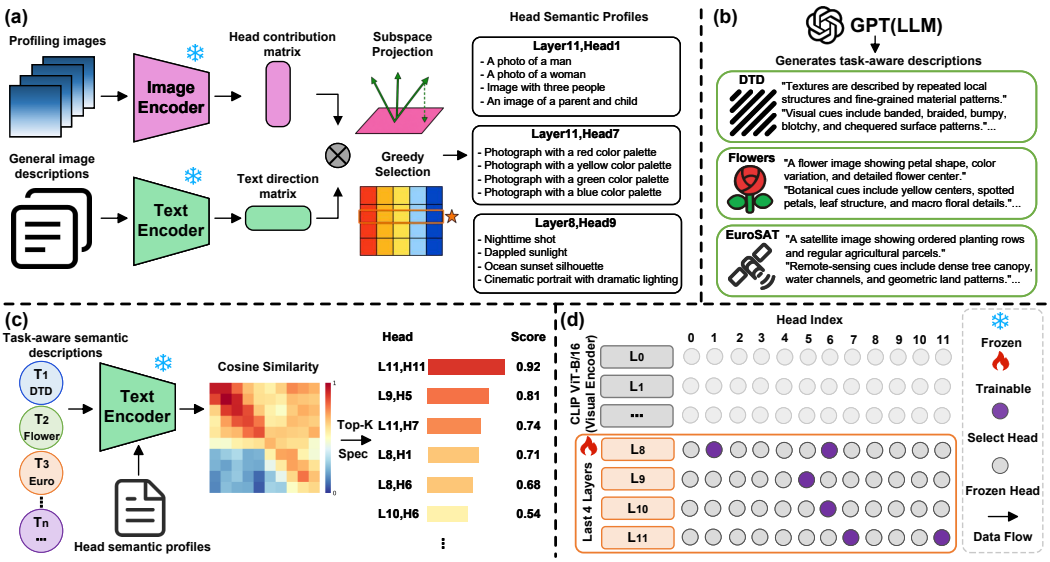

Figure 3: Overview of SAHR. (a) We construct text-aligned semantic profiles for candidate attention heads by decomposing their contributions along text directions. (b) We construct task-aware semantic descriptions from dataset-specific templates and LLM-generated class-wise visual descriptions. (c) We compute head–task correspondence and derive the Specialty Score to rank candidate heads. (d) We selectively adapt the top-K heads in the late layers while freezing the remaining parameters.

their semantic alignment by cosine similarity:

$$\mathrm{sim}(I,t) = \frac{\langle f_I(I), f_T(t)\rangle}{\|f_I(I)\|_2 \, \|f_T(t)\|_2}. \tag{1}$$

For a downstream task with label set $\mathcal{C}$, prediction is performed by comparing the image representation against the textual representations of candidate class descriptions. This shared embedding space provides the basis for using text semantics to analyze and compare internal visual components.

Our method focuses on candidate attention heads in the late layers of the CLIP visual Transformer, which tend to have a stronger direct effect on model outputs and exhibit more pronounced semantic specialization. Instead of treating these heads as homogeneous computational units, we characterize each head by a text-aligned semantic profile. Following recent text-based decomposition analyses Gandelsman et al. (2024), we construct a fixed profiling set of 50,000 images randomly sampled from the Open Images V7 training split Kuznetsova et al. (2020), while discarding all annotations and metadata. The sampled images provide diverse visual inputs to the frozen CLIP visual encoder, from which we collect head contributions in CLIP's joint vision-language embedding space and interpret their semantic variations using CLIP text embeddings from a fixed pool of general image descriptions. The candidate descriptions are generated by prompting GPT to produce general visual descriptions covering objects, attributes, scenes, colors, textures, and compositions. The profiling images and descriptions are used only to build fixed attention-head semantic profiles offline, and are not used during downstream training or inference. We project each head contribution onto the subspace spanned by these text directions and then greedily select the directions that explain the largest residual variation, recording their normalized explanatory weights.

Formally, for each attention head (l,h), we denote its semantic profile as

$$\mathcal{P}_{l,h} = \{(s_j^{l,h}, \lambda_j^{l,h})\}_{j=1}^{M}, \tag{2}$$

where $s_j^{l,h}$ denotes the j-th textual span associated with the head and $\lambda_j^{l,h}$ denotes its normalized explanatory weight, satisfying $\sum_{j=1}^{M} \lambda_j^{l,h} = 1$. The weight $\lambda_j^{l,h}$ reflects the relative amount of head-contribution variation explained by the selected span during the greedy projection process. In practice, we use $M = 20$ spans per head. The resulting weighted semantic profiles capture complementary semantic aspects of attention heads and serve as semantic anchors for the task-aware scoring mechanism introduced next.

3.2 TASK-AWARE SEMANTIC HEAD SCORING

Given the weighted semantic profile $\mathcal{P}_{l,h}$ of each attention head, we next quantify its relevance to a downstream task. For each task τ, we construct a task-aware semantic description set $\mathcal{T}_\tau = \{q_1^\tau, \ldots, q_{N_\tau}^\tau\}$. Rather than relying only on class names or a single hand-crafted template, we enrich the task semantics with LLM-generated class-wise visual descriptions. For each class, the description set consists of 5 sentence-level descriptions and 10 short visual phrases generated by GPT-4.5 using only the dataset name and class name as input, without access to any image samples from the target task. These descriptions summarize characteristic visual attributes, parts, textures, and contextual cues of the target task, and serve as fixed semantic descriptions of the downstream task. They are used to score the relevance of attention heads to the target task, rather than being optimized as trainable prompts.

We encode both the task-aware descriptions and the head spans using the frozen CLIP text encoder:

$$\mathbf{q}_i^\tau = \frac{f_T(q_i^\tau)}{\|f_T(q_i^\tau)\|_2}, \qquad \mathbf{e}_j^{l,h} = \frac{f_T(s_j^{l,h})}{\|f_T(s_j^{l,h})\|_2}. \tag{3}$$

We then measure the semantic alignment between the j-th span of head (l, h) and the i-th task description by cosine similarity:

$$\mathrm{sim}_{j,i}^{l,h,\tau} = \left\langle \mathbf{e}_j^{l,h},\, \mathbf{q}_i^\tau \right\rangle. \tag{4}$$

For each span $s_j^{l,h}$, we retain only its top-k_a matches in $\mathcal{T}_\tau$ and average them to obtain its relevance to task τ:

$$a_j^{l,h,\tau} = \mathrm{TopMean}_{k_a}\left(\{\mathrm{sim}_{j,i}^{l,h,\tau}\}_{i=1}^{N_\tau}\right), \tag{5}$$

where $\mathrm{TopMean}_{k_a}(\cdot)$ denotes the arithmetic mean of the top-k_a largest values in a set, and we set $k_a = 5$ in all experiments. We then aggregate all span-level relevance scores using the profile weights and obtain the raw task-matching score of head (l, h) on task τ:

$$r_{l,h}^\tau = \sum_{j=1}^{M} \lambda_j^{l,h}\, a_j^{l,h,\tau}. \tag{6}$$

Intuitively, $r_{l,h}^\tau$ measures how strongly the semantic profile of head (l, h) matches the current task.

Raw matching scores measure task relevance, while task specificity requires comparing the score against the head's average relevance across tasks. To emphasize relative task preference, we further normalize the raw score across tasks. Let $\mathcal{D}$ denote the available task collection used for task-relative head scoring, consisting of 11 classification datasets or the two retrieval datasets, Flickr30K and MSCOCO, depending on the evaluation setting. We define the Specialty Score of head (l, h) on task τ as

$$\mathrm{Spec}_{l,h}^\tau = \frac{r_{l,h}^\tau}{\frac{1}{|\mathcal{D}|}\sum_{\tau' \in \mathcal{D}} r_{l,h}^{\tau'} + \epsilon}, \tag{7}$$

where $\epsilon = 10^{-3}$ is a small constant for numerical stability. This normalization suppresses heads that are broadly relevant across many tasks and highlights those that are relatively more specialized to the current task. We then rank all candidate heads by $\mathrm{Spec}_{l,h}^\tau$ in descending order and select the top-K heads as the adaptation targets.

3.3 STRUCTURED SPARSE HEAD ADAPTATION

The task-aware head scoring in Section 3.2 indicates that different attention heads should not be adapted uniformly. Since later-layer attention heads exhibit distinct functional specializations and encode complementary semantic factors, tuning all heads is both unnecessary and inefficient. SAHR therefore concentrates adaptation on a small subset of top-ranked heads whose semantic profiles are most strongly aligned with the target task. In this way, the ranking produced by the Specialty Score is turned into an explicit sparse adaptation budget.

Concretely, for each selected head, we update only the parameter slices associated with its query, key, value, and output projections in the multi-head self-attention module. Heads that are not selected remain frozen. This yields a structured sparse adaptation pattern at the granularity of individual heads, rather than coarse-grained updates over entire Transformer blocks or external adaptation modules. Compared with uniform tuning, such a design is better aligned with our empirical observation that different heads encode different semantic factors and contribute unequally to downstream transfer. Because SAHR reconfigures trainability only across existing parameter slices, it introduces no additional trainable modules during training, thereby incurring no additional forward-pass computation at inference time.

SAHR uses a shared offline head-profiling procedure across downstream tasks. For each target task, it constructs task-aware semantic descriptions, recomputes head–task correspondence scores, and selects a task-specific subset of attention heads for adaptation. For few-shot classification, the selected parameter slices are optimized using the classification loss on the support set. For image–text retrieval, SAHR follows the same profiling and scoring procedure but optimizes the task-specific selected slices using the corresponding retrieval objective.

4 Experiments

4.1 Datasets

We evaluate few-shot image classification on 11 standard benchmarks: Caltech101 Fei-Fei et al. (2004), DTD Cimpoi et al. (2014), EuroSAT Helber et al. (2019), FGVCAircraft Maji et al. (2013), Food101 Bossard et al. (2014), Flowers102 Nilsback & Zisserman (2008), ImageNet Deng et al. (2009), OxfordPets Parkhi et al. (2012), StanfordCars Krause et al. (2013), SUN397 Xiao et al. (2010), and UCF101 Soomro et al. (2012). For each dataset, we consider 1-, 2-, 4-, 8-, and 16-shot settings, where $K \in \{1, 2, 4, 8, 16\}$ labeled samples per class are used for training and the official test split is used for evaluation. We report the average accuracy over three random seeds.

We evaluate image-text retrieval on Flickr30K Young et al. (2014) and MSCOCO Lin et al. (2014), following the standard Flickr30K 1K and MSCOCO 5K test protocols. We report both text retrieval (TR) and image retrieval (IR) results using Recall@1, Recall@5, and Recall@10.

For out-of-domain distributions (OOD) robustness, we adopt the standard protocol based on 16-shot ImageNet adaptation. The model is first adapted on 16-shot ImageNet-1K and then directly evaluated on ImageNet-V2 Recht et al. (2019) and ImageNet-Sketch Wang et al. (2019) without further tuning. This setting assesses whether the selected semantic heads remain robust under distribution shift.

4.2 Implementation Details

Unless otherwise specified, we use CLIP ViT-B/16 as the default backbone in main experiments. We optimize the trainable parameters using AdamW with an initial learning rate of 1×10^{-4}, which is decayed following a cosine annealing schedule. The batch size is set to 64. The head selection budget is fixed to $K = 6$ by default. All experiments are implemented in PyTorch and conducted on a single NVIDIA A100-SXM4-40GB GPU.

4.3 Comparison with State-of-the-Art Methods

We compare SAHR with representative CLIP adaptation baselines on both few-shot classification and image-text retrieval. For few-shot classification, the compared methods include recent prompt-based, adapter-based, and cache-based approaches, such as TIMO Li et al. (2025), CLIP-LoRA Zanella & Ben Ayed (2024), Tip-Adapter-F Zhang et al. (2022), CoOp Zhou et al. (2022b), TaskRes Yu et al. (2023), MaPLe Khattak et al. (2023), CLIP-Adapter Gao et al. (2024), SAFT Nguyen et al. (2024), TextRefiner Xie et al. (2025), and TOGA Mohammad et al. (2026). For image-text retrieval, we compare against both full fine-tuning and lightweight adaptation baselines, including VPT Jia et al. (2022), Partial, Projection, CoOp, and MaPLe.

Table 1: Comparison of few-shot classification accuracy (%) on 11 benchmark datasets. The best and second-best performance in each shot-group are marked in bold and underlined, respectively. Dataset abbreviations: INet (ImageNet), SUN (SUN397), Air (FGVC-Aircraft), Euro (EuroSAT), Cars (Stanford Cars), Food (Food101), Pets (OxfordPets), Flow (Flowers102), Cal (Caltech101), DTD (Describable Textures), UCF (UCF101).

Shots	Method	Venue	INet	SUN	Air	Euro	Cars	Food	Pets	Flow	Cal	DTD	UCF	Avg
0	CLIP Radford et al. (2021)	ICML'21	66.7	62.6	24.7	47.5	65.6	86.1	89.1	71.4	89.3	43.6	66.7	64.8
1	TIP-Adapter-F Zhang et al. (2022)	ECCV'22	69.4	67.2	28.8	67.8	67.1	85.8	90.6	83.8	94.0	51.6	73.4	70.9
	CoOp Zhou et al. (2022b)	IJCV'22	65.7	67.0	20.8	56.4	67.5	84.3	90.2	78.3	92.5	50.1	71.2	67.6
	MaPLe Khattak et al. (2023)	CVPR'23	68.7	68.4	28.0	41.9	68.1	84.4	**91.5**	78.8	93.7	51.0	72.6	67.9
	SAFT Nguyen et al. (2024)	ECCV'24	68.5	68.7	30.1	70.8	**69.7**	83.8	91.2	81.2	93.3	53.1	**75.9**	71.5
	TextRefiner Xie et al. (2025)	AAAI'25	69.0	68.0	25.6	53.9	65.1	85.4	88.2	70.1	92.1	44.8	69.9	66.6
	TOGA Mohammad et al. (2026)	CVPR'26	69.2	68.1	**31.0**	67.4	69.1	**86.2**	91.2	**88.2**	**94.3**	55.2	74.5	**72.2**
	SAHR (Ours)	-	**70.0**	**70.1**	27.7	**71.1**	67.1	85.8	91.3	84.9	92.9	**59.1**	73.8	**72.2**
2	TIP-Adapter-F	ECCV'22	70.0	68.6	32.8	73.2	70.8	86.0	91.6	90.1	93.9	57.8	76.2	73.7
	CoOp	IJCV'22	67.0	67.0	25.9	65.1	70.4	84.4	89.9	88.0	93.1	54.1	74.1	70.8
	MaPLe	CVPR'23	69.5	70.5	30.9	72.3	69.8	85.0	**92.6**	84.5	94.2	55.6	74.3	72.7
	SAFT	ECCV'24	68.9	70.1	33.5	**81.1**	70.0	84.3	91.5	87.0	94.5	59.8	**78.3**	74.5
	TextRefiner	AAAI'25	70.1	69.1	26.6	53.4	65.2	85.7	90.1	73.4	93.4	48.7	69.8	67.8
	TOGA	CVPR'26	69.9	70.7	**34.8**	74.9	**72.8**	86.5	92.2	**91.9**	**95.0**	58.4	77.9	**75.0**
	SAHR (Ours)	-	**70.4**	**71.0**	30.1	79.8	69.4	**86.6**	**92.6**	90.1	93.5	**61.1**	78.1	74.8
4	TIP-Adapter-F	ECCV'22	70.7	70.8	35.7	76.8	74.1	86.5	91.9	92.1	94.8	59.8	78.1	75.6
	CoOp	IJCV'22	70.6	69.7	30.9	69.7	74.4	84.5	92.5	92.2	94.5	59.5	77.6	74.2
	MaPLe	CVPR'23	70.8	72.7	32.6	76.6	72.0	85.7	**93.3**	90.5	95.3	61.2	77.9	75.3
	SAFT	ECCV'24	70.0	72.0	35.8	85.5	75.2	84.9	92.7	90.8	95.1	63.6	80.4	76.9
	TextRefiner	AAAI'25	70.7	70.8	29.4	56.0	67.9	**87.0**	92.6	76.2	93.1	51.4	71.9	69.7
	TOGA	CVPR'26	**71.0**	**73.2**	**38.3**	80.3	**76.7**	86.7	92.7	**96.4**	**95.7**	64.5	81.7	**77.9**
	SAHR (Ours)	-	70.9	73.1	35.2	**85.7**	73.7	86.8	**93.3**	93.2	94.6	**66.5**	**82.4**	77.8
8	TIP-Adapter-F	ECCV'22	**71.7**	73.5	39.5	81.3	78.3	86.9	91.8	94.3	95.2	66.7	82.0	78.3
	CoOp	IJCV'22	70.6	71.9	38.5	77.1	79.0	82.7	91.3	94.9	94.5	64.8	80.0	76.8
	MaPLe	CVPR'23	71.6	74.0	37.0	84.86	74.6	86.5	93.1	94.2	95.4	65.6	81.0	78.0
	SAFT	ECCV'24	71.1	73.7	42.5	87.7	**79.3**	85.0	93.2	94.0	95.6	66.7	83.1	79.3
	TextRefiner	AAAI'25	70.8	72.2	31.0	59.6	69.9	**87.0**	92.7	84.1	95.2	55.2	75.4	72.1
	TOGA	CVPR'26	71.5	75.1	**44.2**	84.1	78.2	86.8	93.4	**97.3**	**95.8**	69.6	83.9	80.0
	SAHR (Ours)	-	**71.7**	**75.4**	39.0	**90.1**	76.7	86.9	**93.7**	96.1	**95.8**	**71.5**	**87.3**	**80.4**
16	TIP-Adapter-F	ECCV'22	**73.4**	76.0	44.6	85.9	82.3	86.8	92.6	96.2	95.7	70.8	83.9	80.7
	CoOp	IJCV'22	71.9	74.9	43.2	85.0	82.9	84.2	92.0	96.8	95.8	69.7	83.1	80.0
	MaPLe	CVPR'23	72.6	75.5	40.6	89.0	78.0	87.4	93.3	95.8	95.7	70.5	82.7	80.1
	SAFT	ECCV'24	72.8	75.4	**49.0**	90.9	84.2	85.9	**94.2**	97.0	96.3	71.1	85.9	82.1
	TextRefiner	AAAI'25	70.9	74.3	34.4	64.3	72.1	87.2	93.1	92.4	95.3	61.9	79.6	75.0
	TOGA	CVPR'26	72.3	76.2	48.4	89.4	**85.3**	**87.5**	93.4	98.3	96.3	73.6	84.9	82.3
	SAHR (Ours)	-	72.7	**78.2**	45.7	**92.6**	80.5	87.2	93.9	**98.6**	**97.3**	**75.8**	**91.1**	**83.1**

Table 2: Comparison of image–text retrieval performance on Flickr30K and MSCOCO using CLIP ViT-B/32. TR denotes text retrieval (image-to-text), and IR denotes image retrieval (text-to-image).

Method	Params	Flickr30K(1K test set)						MSCOCO(5K test set)					
		TR			IR			TR			IR		
		R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
CLIP Radford et al. (2021)	–	78.8	94.9	98.2	58.8	83.6	90.2	50.1	75.0	83.5	30.5	56.0	66.9
Full-tuning	123.5M	88.3	98.0	99.2	75.7	94.1	96.7	66.0	88.0	93.4	49.2	76.6	85.3
Partial Jia et al. (2022)	10.2M	81.7	95.5	97.9	66.5	89.4	94.0	57.2	**83.3**	**90.5**	42.7	**71.4**	**81.5**
Projection Jia et al. (2022)	0.7M	82.7	96.9	99.0	66.9	88.9	94.1	55.1	81.5	89.2	40.0	68.3	78.7
CoOp Zhou et al. (2022b)	0.01M	82.3	96.1	98.1	64.1	88.3	93.1	53.7	77.9	86.1	36.1	62.2	72.9
VPT Jia et al. (2022)	0.05M	83.4	96.4	98.9	69.3	90.5	94.9	55.5	79.5	87.9	40.1	67.0	77.4
MaPLe Khattak et al. (2023)	4.8M	**87.5**	97.6	99.1	**72.6**	91.9	**96.0**	**59.8**	82.8	89.9	43.4	70.8	80.3
SAHR (Ours)	1.18M	87.2	**97.8**	**99.4**	71.5	**92.1**	95.7	57.6	82.4	90.0	**43.5**	70.8	80.7

Performance Analysis. Table 1 reports the few-shot classification results on the 11 benchmark datasets. Overall, SAHR delivers consistently strong performance across all shot settings, with its advantage becoming more pronounced as the number of labeled examples increases. It achieves the best average accuracy at 8-shot and 16-shot, reaching 80.4% and 83.1%, respectively. In particular, under 16-shot, SAHR surpasses the strongest compared baseline by 0.8 percentage points while

using 94.6% fewer trainable parameters. These results show that semantic head scoring provides an effective and parameter-efficient way to allocate adaptation capacity to task-relevant visual units.

Table 2 further shows that the same adaptation principle transfers well to image-text retrieval using CLIP ViT-B/32. On Flickr30K, SAHR achieves competitive retrieval performance with fewer trainable parameters than full fine-tuning and MaPLe, obtaining the best or near-best results on several Recall@K metrics. On MSCOCO, SAHR obtains the best image-retrieval R@1 and competitive performance on the remaining retrieval metrics under a lightweight adaptation budget. These results indicate that the proposed semantic head scoring mechanism is not limited to few-shot classification, but can also provide useful adaptation signals for cross-modal retrieval.

Table 3: Comparison of trainable parameters and 16-shot classification accuracy under the ViT-B/16 setting.

Method	TIMO	CoOp	SAFT	CLIP-LoRA	TaskRes	CLIP-Adapter	TextRefiner	MaPLe	Tip-Adapter-F	TOGA	**SAHR**
Params	Free	0.01M	0.15M	0.18M	0.51M	0.52M	1.05M	3.55M	8.19M	21.7M	1.18M
Acc. (%)	77.6	80.0	82.1	83.0	80.8	75.5	75.1	80.1	80.7	82.3	**83.1**

Backbone Scaling. We further evaluate SAHR with different CLIP visual backbones. As shown in Figure 4, the proposed method consistently achieves strong performance across ViT-B/32, ViT-B/16, and ViT-L/14, indicating that the effectiveness of semantic head selection is not tied to a specific backbone scale.

Efficiency. We evaluate the efficiency of SAHR from both parameter and runtime perspectives. As shown in Table 3, under the 16-shot ViT-B/16 setting, SAHR achieves the best average accuracy of 83.1% with only 1.18M trainable parameters, outperforming substantially heavier baselines such as Tip-Adapter-F and TOGA while remaining more effective than lighter PEFT methods. We further report training and inference FPS on ImageNet in Figure 5, measured on a single NVIDIA A100-SXM4-40GB GPU. The results show that SAHR maintains efficiency comparable to existing methods, while adding almost no extra inference cost at test time. Overall, SAHR achieves a favorable trade-off among adaptation accuracy, parameter cost, and runtime efficiency.

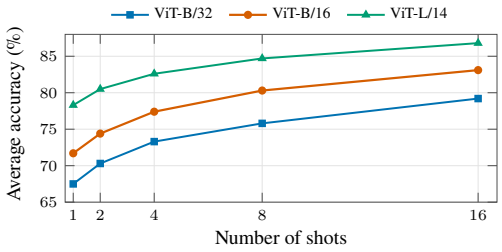

Figure 4: Accuracy across CLIP visual backbones.

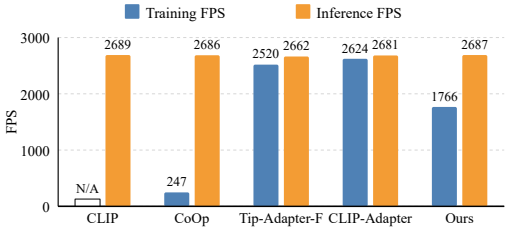

Figure 5: Efficiency comparison on ImageNet.

Robustness Analysis. To evaluate robustness under distribution shift, we evaluate SAHR on ImageNet and its shifted variants, including ImageNet-V2 and ImageNet-Sketch. As shown in Table 4, SAHR achieves the best performance on both the source ImageNet setting and ImageNet-V2, while remaining competitive on ImageNet-Sketch. These results suggest that the heads selected by SAHR capture transferable semantic factors rather than merely overfitting the in-domain support set, thereby maintaining robustness under distribution shifts.

Table 4: Cross-domain generalization results.

Method	Source	Target		Avg
	ImageNet	-V2	-Sketch	
CLIP	66.7	60.8	46.2	57.9
CoOp	71.9	64.2	48.0	61.4
MaPLe	70.7	64.1	**49.2**	61.3
TextRefiner	70.9	62.3	48.6	60.6
SAHR	**72.7**	**64.6**	46.9	**61.4**

4.4 Ablation Studies

Sensitivity to the head budget K. Table 5 examines the sensitivity of SAHR to the head selection budget K. Overall, the performance remains stable when K varies from 6 to 10, indicating that SAHR is not overly sensitive to the exact choice of head budget. Among the tested settings, $K = 6$ achieves the best overall performance while using the smallest number of trainable parameters.

Table 5: Ablation study on the head selection budget K.

Method	Params	$N = 1$	2	4	8	16
K=10	1.97M	71.9	74.2	77.2	80.1	82.9
K=8	1.57M	71.8	74.2	77.3	80.0	82.9
K=6	1.18M	**72.2**	**74.8**	**77.8**	**80.4**	**83.1**

Ablation on selection and adaptation design. Table 6 evaluates four key design choices in SAHR. First, random top-K selection consistently underperforms SAHR, confirming the benefit of semantic head selection. Second, removing specialty normalization (w/o Spec) and ranking heads by raw task-matching scores also reduces accuracy, showing that cross-task normalization better distinguishes task-specific from broadly relevant heads. Third, excluding the V-projection update causes a systematic drop, supporting full Q/K/V/O adaptation. Finally, fine-tuning the last four layers performs worse despite using substantially more parameters, highlighting the advantage of structured sparse adaptation over dense late-layer tuning under few-shot supervision.

Table 6: Ablation study on selection and adaptation strategies.

Method	Params	$N = 1$	2	4	8	16
Random	1.18M	69.4	72.8	75.9	78.9	81.9
w/o Spec	1.18M	71.0	73.9	76.5	79.5	82.1
w/o V	0.88M	69.5	72.6	75.8	78.5	81.7
Last-4	28.35M	65.7	71.0	74.9	79.0	79.2

4.5 Visualization and Analysis

To qualitatively assess class separability across distinct visual domains, we visualize the prediction logits of eight representative classes from each of EuroSAT and DTD using t-SNE under the 16-shot setting in Figure 6. On EuroSAT, SAHR separates visually related land-use categories and improves the silhouette score from 0.005 to 0.413. On DTD, it reduces the overlap among fine-grained texture categories and raises the score from 0.073 to 0.324. This improvement is consistent with SAHR's task-aware reconfiguration, which concentrates adaptation on attention heads that are semantically aligned with the target task.

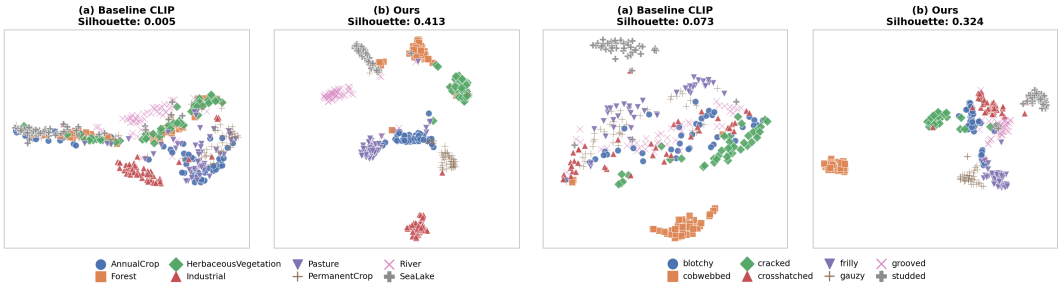

Figure 6: t-SNE visualizations of class logits on EuroSAT (left pair) and DTD (right pair) under the 16-shot setting with CLIP ViT-B/16.

5 Conclusion

In this paper, we propose SAHR, a parameter-efficient framework for semantics-aware reconfiguration of attention heads. SAHR combines text-aligned semantic profiles of attention heads with task-aware descriptions to identify and selectively adapt a sparse set of task-relevant heads. Extensive experiments on few-shot classification, image-text retrieval, and OOD robustness benchmarks demonstrate strong transfer performance with only 1.18M trainable parameters and negligible inference overhead.

AI USE STATEMENT

Generative AI tools were used to produce the dataset-agnostic and class-wise visual descriptions employed by SAHR, as detailed in Sections 3.1–3.2 and Appendix A. They were also used for retrieval and discovery of related research materials. The authors reviewed all AI-assisted outputs and manuscript edits, verified the experimental implementation and reported results, and take full responsibility for the final content, claims, and artifacts.

ETHICS STATEMENT

This work uses established public vision benchmarks and does not involve human-subject experiments or the collection of new personal data. As SAHR relies on pretrained CLIP representations and LLM-generated descriptions, it may inherit biases present in these models and their training data. The method is intended for general research purposes, and applying it to sensitive domains requires additional evaluation of bias, privacy, and safety considerations.

REPRODUCIBILITY STATEMENT

The model formulation and head-selection procedure are described in Section 3, while datasets, evaluation protocols, optimization settings, and compute resources are reported in Section 4. Appendix A further specifies the profiling procedure, description-generation protocol, prompts, task bank, and scoring hyperparameters. These details, together with the reported implementation settings, define the experimental configuration needed to reproduce the reported results.

REFERENCES

Sriram Balasubramanian, Samyadeep Basu, and Soheil Feizi. Decomposing and interpreting image representations via text in vits beyond CLIP. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=Vhh7ONtfvV.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pp. 446–461. Springer, 2014.

Walid Bousselham, Angie Boggust, Sofian Chaybouti, Hendrik Strobelt, and Hilde Kuehne. Legrad: An explainability method for vision transformers via feature formation sensitivity. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20336–20345, October 2025.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3606–3613, 2014.

Zhenyu Cui, Yuxin Peng, Xun Wang, Manyu Zhu, and Jiahuan Zhou. Continual vision-language retrieval via dynamic knowledge rectification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11704–11712, 2024.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pp. 178–178. IEEE, 2004.

Yossi Gandelsman, Alexei A Efros, and Jacob Steinhardt. Interpreting CLIP's image representation via text-based decomposition. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=5Ca9sSzuDp.

Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *International journal of computer vision*, 132(2):581–595, 2024.

Francesco Gentile, Nicola Dall'Asen, Francesco Tonini, Massimiliano Mancini, Lorenzo Vaquero, and Elisa Ricci. From weights to concepts: Data-free interpretability of clip via singular vector decomposition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2895–2906, 2026.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European conference on computer vision*, pp. 709–727. Springer, 2022.

Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19113–19122, 2023.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pp. 554–561, 2013.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International journal of computer vision*, 128(7):1956–1981, 2020.

Yayuan Li, Jintao Guo, Lei Qi, Wenbin Li, and Yinghuan Shi. Text and image are mutually beneficial: Enhancing training-free few-shot classification with clip. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 5039–5047, 2025.

Feng Lin, Marco Chen, Haokui Zhang, Xiaotian Yu, Guangming Lu, and Rong Xiao. Not all attention heads are what you need: Refining clip's image representation with attention ablation. *arXiv preprint arXiv:2507.00537*, 2025.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Avinash Madasu, Vasudev Lal, and Phillip Howard. Pruning the paradox: How clip's most informative heads enhance performance while amplifying bias. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 4611–4626, 2025.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

Sachit Menon and Carl Vondrick. Visual classification via description from large language models. *arXiv preprint arXiv:2210.07183*, 2022.

Mohammed Rahman Sherif Khan Mohammad, Ardhendu Behera, Sandip Pradhan, Swagat Kumar, and Amr Ahmed. Training-only heterogeneous image-patch-text graph supervision for advancing few-shot learning adapters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19613–19622, 2026.

Bac Nguyen, Stefan Uhlich, Fabien Cardinaux, Lukas Mauch, Marzieh Edraki, and Aaron Courville. Saft: Towards out-of-distribution generalization in fine-tuning. In *European Conference on Computer Vision*, pp. 138–154. Springer, 2024.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pp. 722–729. IEEE, 2008.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 3498–3505. IEEE, 2012.

Sarah Pratt, Ian Covert, Rosanne Liu, and Ali Farhadi. What does a platypus look like? generating customized prompts for zero-shot image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15691–15701, October 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pp. 5389–5400. PMLR, 2019.

Karsten Roth, Jae Myung Kim, Andrew Koepke, Oriol Vinyals, Cordelia Schmid, and Zeynep Akata. Waffling around for performance: Visual classification with random words and broad concepts. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 15746–15757, 2023.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

Haohan Wang, Songwei Ge, Zachary Lipton, and Eric Xing. Learning robust global representations by penalizing local predictive power. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. Reft: Representation finetuning for language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 63908–63962. Curran Associates, Inc., 2024. doi: 10.52202/079017-2041. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/75008a0fba53bf13b0bb3b7bff986e0e-Paper-Conference.pdf.

Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 3485–3492. IEEE, 2010.

Jingjing Xie, Yuxin Zhang, Jun Peng, Zhaohong Huang, and Liujuan Cao. Textrefiner: Internal visual feature as efficient refiner for vision-language models prompt tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 8718–8726, 2025.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the association for computational linguistics*, 2:67–78, 2014.

Tao Yu, Zhihe Lu, Xin Jin, Zhibo Chen, and Xinchao Wang. Task residual for tuning vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10899–10909, 2023.

Maxime Zanella and Ismail Ben Ayed. Low-rank few-shot adaptation of vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1593–1603, 2024.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11975–11986, 2023.

Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 46(8):5625–5644, 2024.

Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of clip for few-shot classification. In *European conference on computer vision*, pp. 493–510. Springer, 2022.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16816–16825, 2022a.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International journal of computer vision*, 130(9):2337–2348, 2022b.

APPENDIX OVERVIEW

To support the findings presented in the main paper, this appendix provides implementation details, complete comparison results, additional ablations, efficiency analysis, and qualitative examples. Specifically, we include:

1. Detailed reproducibility information for semantic head profiling, task-aware description generation, and head scoring hyperparameters. (Appendix A)
2. A full comparison with state-of-the-art prompt-based adaptation methods across 11 few-shot classification benchmarks and five shot settings. (Appendix B)
3. Qualitative t-SNE visualizations showing how SAHR improves class-logit separability on representative datasets. (Appendix C)
4. Complete ablation results for the selected head budget and additional parameter-efficiency comparisons. (Appendix D)
5. Qualitative examples of semantic head profiles, task-aware semantic descriptions, and dataset-specific prompt templates. (Appendix E)

A REPRODUCIBILITY DETAILS

This section records the fixed implementation choices used by SAHR. We provide additional details for semantic head profiling, task-aware description generation, and head scoring so that the selected heads can be reproduced from the same CLIP backbone, profiling data, text pools, and scoring hyperparameters.

A.1 SEMANTIC HEAD PROFILING

We construct the semantic profile of each candidate attention head before downstream adaptation. We use a fixed profiling set of 50,000 images randomly sampled from the Open Images V7 training split. Only the raw image pixels are used, while all annotations and metadata are discarded. No category information is used during profiling. The sampled images provide diverse visual inputs for collecting projected head contributions from the frozen CLIP visual encoder. For each CLIP backbone, the semantic profiles are constructed once offline and then reused across all downstream tasks, few-shot settings, and random seeds. The profiling images are not used during downstream training, validation, or inference.

The candidate text pool consists of general image descriptions generated by prompting GPT to produce dataset-agnostic visual descriptions. The descriptions cover common visual factors, including objects, attributes, scenes, colors, textures, spatial layouts, and image compositions. We encode all candidate descriptions with the frozen CLIP text encoder and use the same text pool for all candidate heads.

For each candidate attention head, we follow a TextSpan-style greedy decomposition described in Section 3.1. Given the collected projected head contributions and the candidate CLIP text embeddings, the algorithm repeatedly selects the text direction that explains the largest residual variation in each head contribution. The selected span is assigned an explanatory weight, and the residual is updated before the next selection step. We keep the top $M = 20$ spans for each head and normalize their weights so that $\sum_{j=1}^{M} \lambda_j^{l,h} = 1$. The resulting weighted spans form the fixed semantic profile used in all subsequent head scoring and adaptation experiments.

A.2 TASK-AWARE DESCRIPTION GENERATION

For each downstream classification task, we construct the task-aware semantic description set from dataset-specific template prompts and LLM-generated class-wise visual descriptions. For each class, GPT-4.5 generates 5 sentence-level visual descriptions and 10 short visual phrases using the dataset name and class name as input. The generation input contains no support, validation, or test images.

The generation prompt asks the LLM to describe visible properties of the class that can be useful for visual recognition, including attributes, parts, shapes, textures, viewpoints, and scene context. The

prompt takes the dataset name and class name as input, asks for 5 sentence-level visual descriptions and 10 short visual phrases, and requires a structured JSON output. The sentence-level descriptions provide CuPL-style class descriptions, while the short phrases provide compact visual cues that can match the shorter textual spans in head profiles.

We use temperature 0.7 and allow at most 3 retries for invalid generations. The generated output is stored in a structured format containing the class index, class name, sentence-level descriptions, short visual phrases, and their union. Duplicate descriptions are removed before head scoring. After generation, all task descriptions are encoded by the frozen CLIP text encoder and remain fixed during adaptation.

A.3 Head Scoring Hyperparameters

For span-level matching, we use $k_a = 5$ in the TopMean aggregation. This means each head span is scored by its five strongest semantic matches in the task-aware description set, reducing the influence of unrelated class descriptions. For task-relative normalization, we set $\epsilon = 10^{-3}$ to stabilize the ratio when the cross-task average score is close to zero.

The task collection $\mathcal{D}$ is constructed from the downstream task semantics available in each evaluation setting. For the reported classification results, $\mathcal{D}$ contains the 11 benchmark tasks used in the evaluation. For retrieval experiments, $\mathcal{D}$ contains Flickr30K and MSCOCO. The same task collection is used for all few-shot settings and random seeds. The task collection is used only for relative normalization of head relevance and does not provide any image samples, class labels, or optimization signals.

B Additional Classification Results

Table 7 reports the full few-shot classification comparison with state-of-the-art methods on 11 benchmarks. We include zero-shot CLIP and five representative prompt-learning baselines, together with TaskRes and CLIP-Adapter. The table expands the summarized results in the main paper by showing per-dataset performance under 1-, 2-, 4-, 8-, and 16-shot settings.

Low-shot regime. In the 1-shot setting, SAHR achieves an average accuracy of 72.2%, outperforming the strongest baseline TaskRes by 1.4 percentage point. The improvement is especially clear on datasets requiring localized or structural visual cues, such as EuroSAT (71.1% vs. 65.4%), DTD (59.1% vs. 54.5%), and Flowers102 (84.9% vs. 80.4%). In the 2-shot setting, SAHR reaches 74.8% average accuracy, maintaining the best overall performance and showing consistent gains on EuroSAT and DTD. These results indicate that selecting semantically aligned attention heads provides useful task-specific capacity when supervision is extremely limited.

Medium-shot regime. With 4 and 8 shots, SAHR continues to achieve the best average accuracy, reaching 77.8% and 80.4%, respectively. The advantage remains strong on texture-centric and scene-centric datasets. For example, on DTD, SAHR improves the 4-shot accuracy to 66.5% and the 8-shot accuracy to 71.5%, surpassing all compared baselines. On UCF101, SAHR reaches 82.4% in the 4-shot setting and 87.3% in the 8-shot setting, suggesting that action recognition benefits from adapting heads associated with body, motion, and scene-context cues.

High-shot regime. In the 16-shot setting, SAHR obtains the highest average accuracy of 83.1%, improving over PLOT++ by 1.0 point while using a structured sparse head adaptation strategy. The gains are particularly pronounced on DTD (75.8%), UCF101 (91.1%), SUN397 (78.2%), and Flowers102 (98.6%). SAHR remains competitive on object-centric datasets such as ImageNet, Food101, and OxfordPets, where CLIP already provides strong zero-shot representations. This behavior matches the motivation of our method: task-aware head scoring allocates adaptation to heads whose semantic profiles match the visual factors emphasized by each downstream task.

Overall, the complete results show that the average improvement of SAHR is not driven by a single benchmark or a single shot setting. The method provides stable gains across low-, medium-, and high-shot regimes, with the largest benefits appearing on datasets that require fine-grained texture, scene layout, or action-related semantic cues.

Table 7: Comparison of few-shot classification accuracy (%) on 11 benchmark datasets. We evaluate our method against additional SOTA approaches. Dataset abbreviations: INet (ImageNet), SUN (SUN397), Air (Aircraft), Euro (EuroSAT), Cars (Stanford Cars), Food (Food101), Pets (Oxford-Pets), Flow (Flowers102), Cal (Caltech101), DTD (Describable Textures), UCF (UCF101).

Shots	Method	Venue	INet	SUN	Air	Euro	Cars	Food	Pets	Flow	Cal	DTD	UCF	Avg
0	CLIP	ICML'21	66.7	62.6	24.7	47.5	65.6	86.1	89.1	71.4	89.3	43.6	66.7	64.8
1	CoCoOp	CVPR'22	69.4	68.7	28.1	55.4	67.6	84.9	91.9	73.4	94.1	52.6	70.4	68.8
	PLOT++	ICLR'23	66.4	66.7	28.6	65.4	68.8	86.1	91.8	80.4	94.3	54.5	74.3	70.7
	KgCoOp	CVPR'23	68.9	68.4	26.8	61.9	66.7	86.4	92.1	74.7	94.2	52.7	72.8	69.6
	MaPLe	CVPR'23	68.7	68.4	28.0	41.9	68.1	84.4	91.5	78.8	93.7	51.0	72.6	67.9
	ProGrad	ICCV'23	67.0	67.0	28.8	57.0	68.2	84.9	91.4	80.9	93.5	52.8	73.3	69.5
	TaskRes	CVPR'23	69.6	68.1	31.3	65.4	68.8	84.6	90.2	81.7	93.6	53.8	71.7	70.8
	CLIP-Adapter	IJCV'24	67.9	65.4	25.2	49.3	65.7	86.1	89.0	71.3	92.0	44.2	66.9	65.7
	(Ours)	-	70.0	70.1	27.7	71.1	67.1	85.8	91.3	84.9	92.9	59.1	73.8	72.2
2	CoCoOp	CVPR'22	70.1	69.4	29.3	61.8	68.4	85.9	91.9	77.8	94.4	52.3	73.4	70.4
	PLOT++	ICLR'23	68.2	68.0	31.1	76.8	73.1	86.3	92.2	89.8	94.6	56.7	76.7	74.0
	KgCoOp	CVPR'23	69.6	69.6	28.0	69.2	68.2	86.6	92.3	79.8	94.5	55.3	74.6	71.6
	MaPLe	CVPR'23	69.5	70.5	30.9	72.3	69.8	85.0	92.6	84.5	94.2	55.6	74.3	72.7
	ProGrad	ICCV'23	69.1	69.0	31.1	66.3	72.4	84.8	91.5	87.5	93.6	56.0	75.6	72.4
	TaskRes	CVPR'23	70.2	70.5	32.7	70.2	72.1	85.6	90.7	84.4	94.3	55.6	75.2	72.9
	CLIP-Adapter	IJCV'24	68.2	67.2	27.0	51.2	66.6	86.2	89.7	71.7	93.4	45.4	68.4	66.8
	(Ours)	-	70.4	71.0	30.1	79.8	69.4	86.6	92.6	90.1	93.5	61.1	78.1	74.8
4	CoCoOp	CVPR'22	70.8	70.4	30.6	61.7	69.5	86.3	92.7	81.5	94.8	55.7	75.3	71.8
	PLOT++	ICLR'23	70.4	71.7	35.2	83.2	76.2	86.4	92.5	92.9	95.1	62.4	79.7	76.9
	KgCoOp	CVPR'23	70.2	71.5	32.2	71.8	69.5	86.9	92.6	87.0	95.0	58.7	77.6	73.9
	MaPLe	CVPR'23	70.8	72.7	32.6	76.6	72.0	85.7	93.3	90.5	95.3	61.2	77.9	75.3
	ProGrad	ICCV'23	71.3	71.7	34.1	69.6	75.0	85.4	92.1	91.1	94.4	59.7	77.9	74.8
	TaskRes	CVPR'23	71.0	72.7	33.4	74.2	76.0	86.0	91.9	85.0	95.0	60.1	76.2	74.7
	CLIP-Adapter	IJCV'24	68.6	68.0	27.9	51.2	67.5	86.5	90.8	73.1	94.0	46.1	70.6	67.7
	(Ours)	-	70.9	73.1	35.2	85.7	73.7	86.8	93.3	93.2	94.6	66.5	82.4	77.8
8	CoCoOp	CVPR'22	70.8	71.5	32.4	69.1	70.4	87.0	93.3	86.3	94.9	60.1	75.9	73.8
	PLOT++	ICLR'23	71.3	73.9	41.4	88.3	81.2	86.5	93.0	95.4	95.5	66.4	82.8	79.6
	KgCoOp	CVPR'23	70.2	72.6	34.8	73.9	72.8	87.0	93.0	91.5	95.1	65.6	80.0	76.0
	MaPLe	CVPR'23	71.6	74.0	37.0	84.9	74.6	86.5	93.1	94.2	95.4	65.6	81.0	78.0
	ProGrad	ICCV'23	71.3	73.0	37.7	77.8	78.7	86.1	92.2	95.0	94.8	63.9	80.5	77.4
	TaskRes	CVPR'23	72.3	74.6	40.3	77.5	79.6	86.4	92.0	96.0	95.3	66.7	81.6	78.4
	CLIP-Adapter	IJCV'24	69.1	71.7	30.5	61.6	70.7	86.9	91.9	83.3	94.5	50.5	76.2	71.5
	(Ours)	-	71.7	75.4	39.0	90.1	76.7	86.9	93.7	96.1	95.8	71.5	87.3	80.4
16	CoCoOp	CVPR'22	71.1	72.6	33.3	73.6	72.3	87.4	93.4	89.1	95.1	63.7	77.2	75.3
	PLOT++	ICLR'23	72.6	76.0	46.7	92.0	84.5	87.1	93.5	97.5	96.0	71.4	85.3	82.1
	KgCoOp	CVPR'23	70.4	73.3	36.5	76.2	74.8	87.2	93.2	93.4	95.2	68.7	81.7	77.3
	MaPLe	CVPR'23	72.6	75.5	40.6	89.0	78.0	87.4	93.3	95.8	95.7	70.5	82.7	80.1
	ProGrad	ICCV'23	72.1	75.1	43.0	83.6	82.9	85.8	92.8	96.6	95.9	68.8	82.7	79.9
	TaskRes	CVPR'23	73.0	76.1	44.9	82.7	83.5	86.9	92.4	97.5	95.8	71.5	84.0	80.8
	CLIP-Adapter	IJCV'24	69.8	74.2	34.2	71.4	74.0	87.1	92.3	92.9	94.9	59.4	80.2	75.5
	(Ours)	-	72.7	78.2	45.7	92.6	80.5	87.2	93.9	98.6	97.3	75.8	91.1	83.1

C Qualitative Analysis of Class-Logit Separability via t-SNE

To qualitatively evaluate class separability, we visualize the logit distributions using t-Distributed Stochastic Neighbor Embedding (t-SNE).The Silhouette scores are computed in the original normalized class-logit space rather than the two-dimensional t-SNE coordinates. We compare baseline CLIP and our method under the 16-shot setting with CLIP ViT-B/16 as the visual backbone. For each dataset, we randomly select 8 representative categories from EuroSAT, DTD, and UCF101.

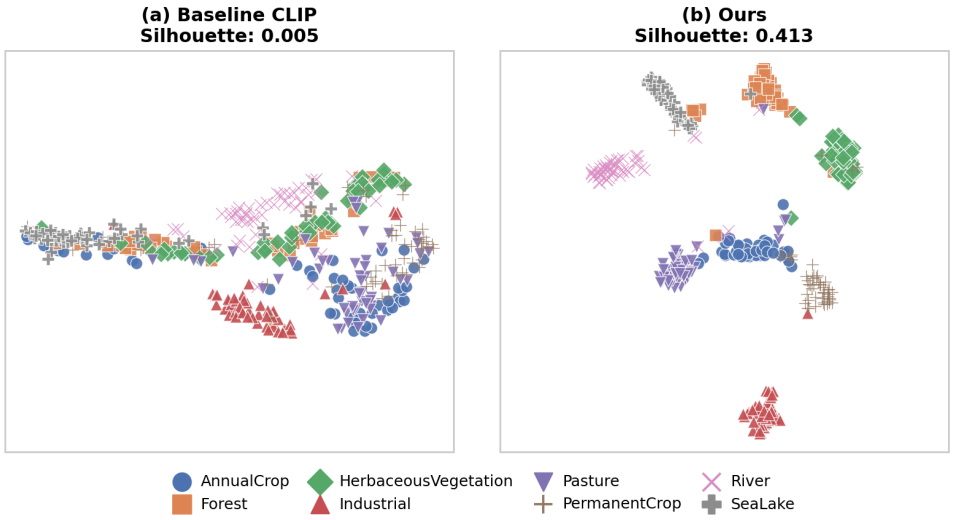

Figure 7: t-SNE visualization of logits on the EuroSAT dataset under the 16-shot setting with CLIP ViT-B/16. We visualize 8 representative land-use categories. Compared with the baseline CLIP class-logit vectors, which exhibit severe class overlap and weak decision boundaries with a Silhouette score of 0.005, our method produces more compact intra-class clusters and clearer inter-class separation. The Silhouette score is improved to 0.413, indicating that our method better distinguishes visually related remote-sensing categories such as Pasture, Herbaceous Vegetation, and Permanent Crop.

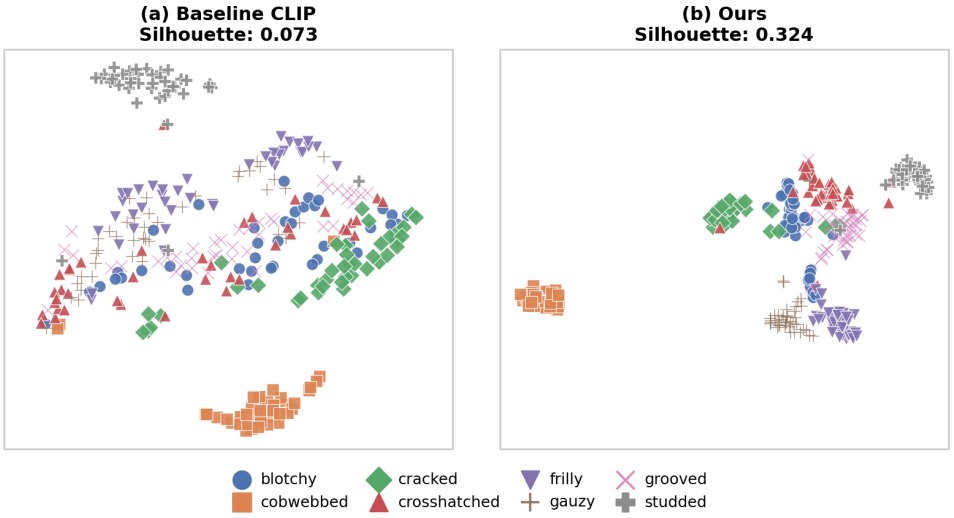

Figure 8: t-SNE visualization of logits on the DTD dataset under the 16-shot setting with CLIP ViT-B/16. The plots show the class-logit distributions of 8 texture categories. Baseline CLIP struggles to separate several fine-grained texture patterns, resulting in noticeable overlap among categories such as grooved, gauzy, and crosshatched, with a Silhouette score of 0.073. In contrast, our method reshapes the class-logit space into more discriminative clusters and raises the Silhouette score to 0.324, demonstrating improved separability for texture recognition.

The visualizations show that baseline CLIP often produces entangled class-logit structures, especially on datasets with fine-grained visual ambiguity. On EuroSAT, several land-use categories are heavily mixed, leading to a very low Silhouette score of 0.005. Our method substantially improves the clustering structure, separating confusing categories such as Pasture, Herbaceous Vegetation, and Permanent Crop, and increases the Silhouette score to 0.413. Similar improvements are observed on DTD, where texture categories with subtle local patterns become more distinguishable,

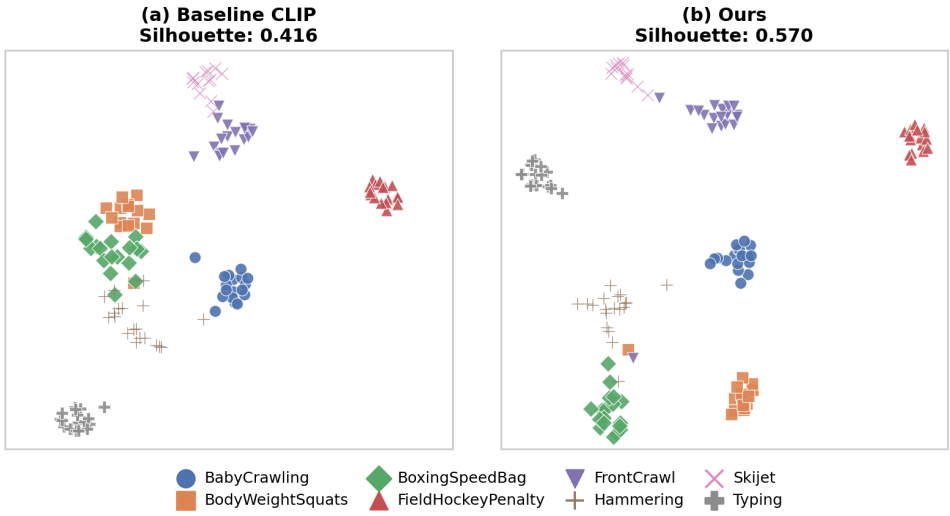

Figure 9: t-SNE visualization of logits on the UCF101 dataset under the 16-shot setting with CLIP ViT-B/16. We randomly select 8 action categories for visualization. Baseline CLIP already forms partially separable clusters but still shows ambiguity among semantically or visually related actions. Our method further tightens class-specific clusters and enlarges the margins between different action categories, improving the Silhouette score from 0.416 to 0.570.

Table 8: Ablation on the selected head number K. Accuracy (%) of our method with different selected head numbers ($K \in \{6, 8, 10\}$) under $N \in \{1, 2, 4, 8, 16\}$-shot settings.

Shots	Method	INet	SUN	Air	Euro	Cars	Food	Pets	Flow	Cal	DTD	UCF	Avg
1	Top-K=6	70.0	70.1	27.7	71.1	67.1	85.8	91.3	84.9	92.9	59.1	73.8	72.2
	Top-K=8	69.6	70.1	27.5	68.6	67.1	85.8	90.9	84.3	92.9	59.0	73.7	71.8
	Top-K=10	69.9	70.0	27.4	70.7	67.0	85.6	91.5	83.9	92.9	58.3	73.7	71.9
2	Top-K=6	70.4	71.0	30.1	79.8	69.4	86.6	92.6	90.1	93.5	61.1	78.1	74.8
	Top-K=8	70.1	70.9	29.7	76.0	68.2	86.1	92.6	89.6	93.1	61.0	78.4	74.2
	Top-K=10	70.3	71.0	29.9	78.2	68.6	85.3	92.5	89.0	93.5	60.2	77.5	74.2
4	Top-K=6	70.9	73.1	35.2	85.7	73.7	86.8	93.3	93.2	94.6	66.5	82.4	77.8
	Top-K=8	70.9	73.2	35.1	84.5	71.1	85.6	93.2	93.1	94.4	67.0	82.5	77.3
	Top-K=10	70.8	72.9	34.4	84.1	71.6	86.6	93.7	92.7	94.4	65.3	82.6	77.2
8	Top-K=6	71.7	75.4	39.0	90.1	76.7	86.9	93.7	96.1	95.8	71.5	87.3	80.4
	Top-K=8	71.6	75.3	38.9	88.7	76.1	86.8	93.6	96.0	95.1	71.7	86.1	80.0
	Top-K=10	71.5	75.6	37.4	89.3	76.4	86.6	93.7	96.0	95.8	71.4	87.3	80.1
16	Top-K=6	72.7	78.2	45.7	92.6	80.5	87.2	93.9	98.6	97.3	75.8	91.1	83.1
	Top-K=8	72.4	78.2	45.7	92.1	80.5	86.8	93.4	98.6	97.2	75.9	91.1	82.9
	Top-K=10	72.7	78.1	45.0	91.7	80.6	86.9	93.7	98.6	97.3	75.3	91.6	82.9

improving the score from 0.073 to 0.324. On UCF101, although already yields moderately separated class-logit clusters, our method further tightens the action clusters and enlarges inter-class margins, raising the Silhouette score from 0.416 to 0.570. These representative visualizations are consistent with the quantitative results across all 11 benchmarks, suggesting that our method generally enhances intra-class compactness and inter-class separability rather than overfitting to a specific dataset type.

Table 9: Comparison of trainable parameters and classification accuracy of 16-shot learning (ViT-B/16 setting).

Method	Trainable Params	Accuracy(%)
TIMO	Free	77.6
CoOp	0.01M	80.0
SAFT	0.15M	82.1
CLIP-Adapter	0.52M	75.5
TaskRes	0.51M	80.8
TextRefiner	1.05M	75.1
MaPLe	3.55M	80.1
Tip-Adapter-F	8.19M	80.7
TOGA	21.7M	82.3
(Ours)	1.18M	**83.1**

D Additional Ablation and Efficiency Results

Table 8 provides the full per-dataset ablation for the selected head budget K. The results show that $K = 6$ gives the best average performance while using the smallest adaptation budget among the tested settings.

Increasing K from 6 to 8 or 10 does not consistently improve performance. In several datasets, adding more heads slightly reduces accuracy, suggesting that lower-ranked heads may introduce less task-specific updates. We therefore use $K = 6$ as the default setting, which offers a compact and stable adaptation budget.

The trend is consistent across different shot regimes. In the 1-shot and 2-shot settings, the selected heads need to extract transferable semantic factors from very limited supervision, and the smaller budget avoids introducing additional trainable slices with weaker task relevance. In the 8-shot and 16-shot settings, increasing the budget still brings little benefit, indicating that the top-ranked heads already provide sufficient task-specific capacity. This supports the central design choice of SAHR: the adaptation budget is assigned according to semantic relevance, not by uniformly expanding the number of trainable heads.

Table 9 further compares parameter efficiency under the 16-shot ViT-B/16 setting. SAHR tunes 1.18M parameters, substantially fewer than heavier adaptation baselines such as MaPLe, Tip-Adapter-F, and TOGA, while achieving the best average accuracy in this comparison. Methods with smaller parameter counts, such as CoOp and SAFT, remain efficient but do not reach the same accuracy. These results show that head-level sparse adaptation provides a favorable balance between capacity and selectivity.

The parameter count of SAHR comes from updating only the query, key, value, and output projection slices associated with the selected heads. All unselected heads and the remaining backbone parameters stay frozen. This structured sparsity keeps the optimization target small while preserving the original CLIP representation outside the selected semantic heads.

E Qualitative Examples and Prompt Templates

Table 11 provides examples of task-aware semantic descriptions used for head scoring. Each row corresponds to a downstream task and shows representative class-level descriptions sampled from its generated task description set.

Table 10 shows representative semantic profiles of attention heads in CLIP ViT-B/16. The selected spans reveal recurring specialization patterns, including color, location, people, animals, illumination, artistic style, text symbols, scene context, and body or motion cues.

These examples illustrate why head-level adaptation can be guided by text semantics. A task involving flowers or textures can benefit from heads aligned with colors, textures, and fine-grained visual attributes, while scene or action recognition can rely more on heads aligned with locations,

Table 10: Examples of text-aligned semantic profiles for attention heads in CLIP ViT-B/16. Each cell lists representative textual spans associated with a head, revealing recurring specialization patterns such as color, location, people, animals, illumination, artistic style, text symbols, scene context, and body or motion cues.

L11.H1(*People*)	**L11.H7(*Color*)**	**L11.H6(*Location*)**
A photo of a woman	Photograph with a blue color palette	Photo taken in Bangkok, Thailand
Image with three people	Photograph with a red color palette	Picture taken in the Scotland countryside
A photo of a man	Photograph with a yellow color palette	Picture taken in Arizona, USA
A photo of a young person	Photograph with a green color palette	Picture taken in Bavaria, Germany
An image of a parent and child	Photograph with a purple color palette	Image snapped in the Alaskan wilderness
L11.H11(*Animals*)	**L10.H6(*Body / Motion Cues*)**	**L9.H6(*Artistic Style*)**
Detailed reptile close-up	A leg	Ethereal double exposure photography
An image with dogs	Colorful procession	Street art-inspired mural painting
Image with a ladybug	Contemplative monochrome portrait	Detailed charcoal sketch
An image with seagulls	Graceful wings in motion	Impressionist-style digital painting
An elephant	A barbed wire design	Stark and minimalist urban scene
L8.H9(*Illumination*)	**L10.H4(*Scene Context*)**	**L10.H9(*Text / Symbols*)**
Dappled sunlight	Picture taken in a cozy mountain cabin	A graffiti with a sentence
Nighttime shot	Photograph taken in a train station	Artwork featuring barcode arrangement
Ocean sunset silhouette	Photo taken in Cairo, Egypt	A book
Urban rooftop panorama	Cozy outdoor picnic	Collage of vintage magazine clippings
Photograph taken in a music store	Photograph taken in a retro diner	Detailed charcoal sketch

context, people, and motion cues. SAHR uses the scoring mechanism in Section 3.2 to perform this matching automatically for each task.

Table 12 lists the dataset-specific class templates used to construct the task-aware semantic description sets. These templates are combined with the LLM-generated class-wise descriptions described in Appendix A.2.

The templates provide a simple class-name anchor for each dataset, while the generated descriptions expand the task semantics with visual attributes and discriminative cues. ImageNet uses the common ensemble of seven templates, following standard CLIP evaluation practice.

Table 11: Examples of task-aware semantic descriptions used for head scoring. For each task, we show representative class-level sentence descriptions and several short visual phrases sampled from the generated task description set.

Task	Example Sentence-Level Descriptions	Example Short Visual Phrases
DTD	A blotchy surface has visible patches of color or texture. A chequered surface has a regular pattern of squares. A braided texture has thin strands woven in a criss-cross pattern. Common material cues include fine-grained detail and repeated local structure.	Uneven color patches Regular square grid Interlacing strands Repeated local texture
Flowers102	A rose has red, pink, or white petals arranged in a spiral pattern. A sunflower has a large head with bright yellow petals. A water lily has a flat flower with pale petals and yellow stamens. A tiger lily is a type of flower with orange or red petals and black spots.	Spiral petals Thorny stem Radiating yellow petals Flat floating flower
EuroSAT	Forest regions show dense tree canopy and dark green land cover. Rivers appear as elongated water channels with meandering boundaries. Crop land shows rectangular parcels and alternating color blocks. The overhead layout typically contains mixed urban blocks.	Dense tree canopy Meandering water channel Rectangular field parcels Cultivated land grid
UCF101	Playing guitar shows hands interacting with the guitar body and neck. A volleyball spike shows an overhead arm pose near the ball. Writing on board shows a hand marking a chalkboard or whiteboard. A person doing biking looks like they are sitting on a bike and pedaling with their legs.	Hand-object interaction Overhead hitting pose Board-writing gesture Visible action context Human Body Pose

Table 12: Dataset-specific prompt templates. Following common practice, all datasets except ImageNet and its variants use a single prompt template.

Dataset	Template
Caltech101	“a photo of a {class}.”
Stanford Cars	“a photo of a {class}.”
SUN397	“a photo of a {class}.”
DTD	“{class} texture.”
EuroSAT	“a centered satellite photo of {class}.”
FGVC Aircraft	“a photo of a {class}, a type of aircraft.”
Food101	“a photo of {class}, a type of food.”
Flowers102	“a photo of a {class}, a type of flower.”
OxfordPets	“a photo of a {class}, a type of pet.”
UCF101	“a photo of a person doing {class}.”
ImageNet	Ensemble of 7 templates