Bridging the Modality Gap in Compositional Zero-Shot Learning via Sparse Alignment and Unimodal Memory Bank

Supplementary Material

This appendix provides additional details for the CVPR 2026, titled "Bridging the Modality Gap in Compositional Zero-Shot Learning via Sparse Alignment and Unimodal Memory Bank". It is orgnized as follows:

- §A Detailed Experiment Settings.
- §B More Ablation Experiments.
- §C More Qualitative Experiments.
- §D Pseudo-code.

A. Detailed Experiment Settings

Detailed Dataset Split Statistics. We conduct experiments on three widely-used datasets: UT-Zappos, MIT-States, and C-GQA. UT-Zappos is a fine-grained dataset composed of 50,025 shoes images with 16 attributes ($e.g.$, Cotton, Nylon), 12 objects ($e.g.$, Shoes.Heels, Boots.Ankle) and 116 compositions. MIT-States contains 53,753 natural images with 115 attributes ($e.g.$, Ancient, Broken), 245 objects ($e.g.$, Computer, Tree) and 1962 compositions. C-GQA is the most extensive dataset containing 39,298 images with 453 attributes, 870 objects and more than 9,500 compositions. Following the standard split, we divide the compositions into *train* / *validation* / *test* splits. The detailed splits are shown in Tab. 8. $|\mathcal{C}_s|$ indicates the number of seen compositions, $|\mathcal{C}_u|$ is the number of unseen compositions, $\mathcal{X}$ represents the number of samples in the corresponding splits.

Detailed Evaluation Metrics. Following the generalized CZSL evaluation protocol [26, 29], our method is evaluated on both seen and unseen compositions. We report the four widely used metrics for a comprehensive evaluation. Seen Accuracy (S) and Unseen Accuracy (U) are computed to evaluate the best classification performance on seen and unseen compositions. Using Seen Accuracy as x-axis and Unseen Accuracy as y-axis, we calibrate a bias scalar [29] on Unseen Accuracy and obtain a seen-unseen accuracy curve. Then, we compute and report the Area Under the Curve (AUC). Meanwhile, we compute the best Harmonic Mean (HM) between Seen Accuracy and Unseen Accuracy at a specific bias scalar.

More Implementation Details. For network initialization, we load the weights of CLIP [35] and tune the image encoder with LoRA [58]. The *Sparse Alignment* suppresses semantically irrelevant regions to achieve information balance in image-text pairs. The overall pipeline of *Sparse Alignment* is illustrated in Fig. 7. The *Visual Adaptive condensation* module is implemented with K blocks composed of multi-head attention and feed-forward network. The number of blocks K is set to 3, 3 and 1 for UT-Zappos, MIT-States and C-GQA, respectively. The *Dynamically Updated Memory Bank* does not introduce additional parameters, as the retrieval and prediction processes are calculated on the condensed visual representations without transformation. The coefficient α for distillation loss in Eq. 8 is set to 0.5, 0.9 and 0.5 for three datasets. The coefficient β in Eq. 11 is set to 0.3, 0.7 and 0.7. The coefficient γ in Eq. 11 is set to 0.5, 0.4 and 0.1. For the number of stored samples in *Dynamically Updated Memory Bank* is set to 16, 24 and 16. We train our model for 15, 10 and 15 epochs with Adam Optimizer [16]. The learning rates are initialized at $2e-4$, $5e-5$ and $5e-4$, where the learning rate is scheduled by the StepLR [33]. During training, we set batch size to 64, 64 and 16 for three datasets. All the experiments are conducted on a single NVIDIA RTX 3090 GPU. More ablation experiments on hyper-parameters is presented in Sec. B.

B. More Ablation Experiments

More Comparison with SOTA Methods. Due to space limitations, we report here a more comprehensive comparison of experiments, in which we additionally include more impressive CLIP-based methods. The results are reported in Tab 10 and Tab 11.

Ablation Study on the Arrival Order of Test Samples. Since predictions for the current test batch depend on the most recently updated memory bank, the order of test samples may influence performance. To evaluate the robustness of our method, we analyze the impact of arrival order on the final performance by testing with different random seeds. As shown in Tab.9, the arrival order of samples does influence the predictions of memory bank, causing performance fluctuations. However, our final classification does not rely solely on the memory bank, which ensures overall robustness. Notably, SAM does not alter default arrival order for samples under all experiments.

More Ablation Study on Hyper-Parameteres. We further study the impact of hyper-parameters on performance, including weight coefficient α in distillation loss Eq. 7, weight coefficient β, γ in inference Eq. 11 and number of blocks K in VAC. The detailed analysis is as follows:

Influence of Loss Coefficient Weight α. First, we conduct experiments on α to investigate the impact of the distillation loss in Eq. 8 on the Visual Adaptive Condensation module and the results are reported in Tab. 12. According to the analysis, we set the α as 0.5, 0.9 and 0.5 for UT-Zappos,

Dataset	Compositions			Train		Val		Test	
	$\lvert\mathcal{A}\rvert$	$\lvert\mathcal{O}\rvert$	$\lvert\mathcal{A}\rvert \times \lvert\mathcal{O}\rvert$	$\lvert\mathcal{C}_s\rvert$	$\lvert\mathcal{X}\rvert$	$\lvert\mathcal{C}_s\rvert/\lvert\mathcal{C}_u\rvert$	$\lvert\mathcal{X}\rvert$	$\lvert\mathcal{C}_s\rvert/\lvert\mathcal{C}_u\rvert$	$\lvert\mathcal{X}\rvert$
UT-Zappos	16	12	192	83	22998	15 / 15	3214	18 / 18	2914
MIT-States	115	245	28175	1262	30338	300 / 300	10420	400 / 400	12995
C-GQA	413	674	278362	5592	26920	1252 / 1040	7280	888 / 923	5098

Table 8. Detail of data split statistics.

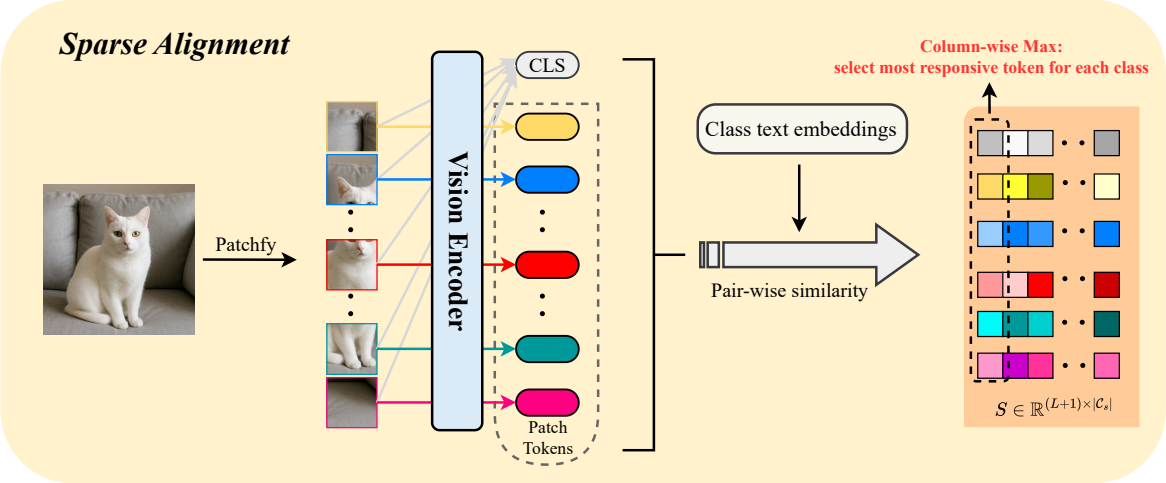


Figure 7. Pipeline of sparse alignment.

Method	UT-Zappos		MIT-States		C-GQA	
	HM	**AUC**	**HM**	**AUC**	**HM**	**AUC**
Seed0	61.8	49.8	40.6	23.8	34.8	16.2
Seed1	62.0	50.0	40.7	23.9	34.7	16.1
Seed2	61.6	49.5	40.1	23.6	34.9	16.3
Seed3	61.4	49.3	40.2	23.7	34.9	16.3
Seed4	61.4	49.3	40.5	23.8	34.8	16.2

Table 9. **Ablation Study on the Arrival order of Test Samples.**

MIT-States and C-GQA, respectively. As we can see, the distillation loss provides a clear performance gain for the VAC module. Ablating this loss (*e.g.*, setting the weight to 0) reduces the VAC objective to a standard classification loss, resulting in notably poorer performance. Notably, the weight α is set to 0.9 for MIT-States due to its noisy annotations, where VAC requires supervision from SA rather than the incorrect labels.

Influence of Inference Weight of β and γ. Then, we conduct experiments on inference weight β and γ in Eq. 11 and the results are reported in Tab. 13 and Fig. 8, respectively. We observe that the optimal parameter settings differ across benchmarks. We hypothesize that this arises from varying dataset characteristics, including differences in object or attribute contamination from surrounding regions. Consequently, adjusting the contribution of our modules yields different levels of performance gain. Therefore, based on our experimental results, we set β as 0.3, 0.7 and 0.7, and set γ as 0.5, 0.4 and 0.1 for UT-Zappos, MIT-States and C-GQA, respectively.

Influence of Number of Blocks in VAC. In addition, we report the ablation study for K, number of blocks in *Visual Adaptive Condensation* module. After a comprehensive evaluation, we ultimately set K as 3, 3 and 1 for UT-zappos, MIT-States and C-GQA, respectively. The detailed performance are reported in Tab. 17.

Influence of Hyper-parameters in Memory Bank. We empirically investigate the impact of the number of stored samples N, temperature τ_{mb}, entropy threshold and test batch size. As illustrated in Tab. 18, we observe that a small memory size leads to suboptimal and unstable performance due to limited sample diversity, and the performance becomes consistent as the memory size increases. However, continually increasing the memory size (*e.g.*, by initializing new slots as zero vectors) may dilute retrieval weights in Eq. 10. Based on this analysis, we set the number of samples to 16, 24, and 16 for datasets UT-Zappos, MIT-States, and C-GQA, respectively. We set the temperature τ_{mb} as 0.1 in Eq. 10, which provides stable performance in Tab. 14, to sharpen the weight of effective samples. The entropy

Method	UT-Zappos				MIT-States				C-GQA			
	S	U	HM	AUC	S	U	HM	AUC	S	U	HM	AUC
CLIP[ICML'21] [35]	15.8	49.1	15.6	5.0	30.2	46.0	26.1	11.0	7.5	25.0	8.6	1.4
CoOp[IJCV'22] [66]	52.1	49.3	34.6	18.8	34.4	47.6	29.8	13.5	20.5	26.8	17.1	4.4
PCVL[Arxiv'22] [50]	64.4	64.0	46.1	32.2	48.5	47.2	35.3	18.3	-	-	-	-
HPL[IJCAI'23] [44]	63.0	68.8	48.2	35.0	47.5	50.6	37.3	20.2	30.8	28.4	22.4	7.2
CSP[ICLR'23] [31]	64.2	66.2	46.6	33.0	46.6	49.9	36.3	19.4	28.8	26.8	20.6	6.2
DFSP[CVPR'23] [27]	66.7	71.7	47.2	36.0	46.9	52.0	37.3	20.6	38.2	32.0	27.1	10.5
PLID[ECCV'24] [1]	67.3	68.8	52.4	38.7	49.7	52.4	39.0	22.1	38.8	33.0	27.9	11.0
CDS[CVPR'24] [20]	63.9	74.8	52.7	39.5	50.3	52.9	39.2	22.4	38.3	34.2	28.1	11.1
Troika[CVPR'24] [9]	66.8	73.8	54.6	41.7	49.0	53.0	39.3	22.1	41.0	35.7	29.4	12.4
CAILA[WACV'24] [63]	67.8	74.0	57.0	44.1	51.0	53.9	39.9	23.4	43.9	38.5	32.7	14.8
RAPR[AAAI'24] [11]	69.4	72.8	56.5	44.5	50.0	53.3	39.2	22.5	45.6	36.0	32.0	14.4
LogiCzsl[CVPR'25] [47]	69.6	74.9	57.8	45.8	50.8	53.9	40.5	23.4	44.4	39.4	33.3	15.3
ClusPro[ICLR'25] [34]	70.7	76.0	58.5	46.6	52.1	**54.0**	40.7	23.8	44.3	37.8	32.8	14.9
SAM-CZSL	**73.3**	**76.8**	**62.0**	**50.0**	**53.2**	53.0	**40.8**	**24.0**	**45.8**	**39.5**	**34.8**	**16.2**

Table 10. The experimental results on closed-world settings.

Method	UT-Zappos				MIT-States				C-GQA			
	S	U	HM	AUC	S	U	HM	AUC	S	U	HM	AUC
CLIP[ICML'21] [35]	15.7	20.6	11.2	2.2	30.1	14.3	12.8	3.0	7.5	4.6	4.0	0.3
CoOp[IJCV'22] [66]	52.1	31.5	28.9	13.2	34.6	9.3	12.3	2.8	21.0	4.6	5.5	0.7
PCVL[Arxiv'22] [50]	64.6	44.0	37.1	21.6	48.5	16.0	17.7	6.1	-	-	-	-
HPL[IJCAI'23] [44]	63.4	48.1	40.2	24.6	46.4	18.9	19.8	6.9	30.1	5.8	7.5	1.4
CSP[ICLR'23] [31]	64.1	44.1	38.9	22.7	46.3	15.7	17.4	5.7	28.7	5.2	6.9	1.2
DFSP[CVPR'23] [27]	66.8	60.0	44.0	30.3	47.5	18.5	19.3	6.8	38.3	7.2	10.4	2.4
PLID[ECCV'24] [1]	67.6	55.5	46.6	30.8	49.1	18.7	20.0	7.3	39.1	7.5	10.6	2.5
CDS[CVPR'24] [20]	64.7	61.3	48.2	32.3	49.4	21.8	22.1	8.5	37.6	8.2	11.6	2.7
Troika[CVPR'24] [9]	66.4	61.2	47.8	33.0	48.8	18.7	20.1	7.2	40.8	7.9	10.9	2.7
CAILA[WACV'24] [63]	67.8	59.7	49.4	32.8	51.0	20.2	21.6	8.2	43.9	8.0	11.5	3.1
RAPR[AAAI'24] [11]	69.4	59.4	47.9	33.3	49.9	20.1	21.8	8.2	45.5	11.2	14.6	4.4
LogiCzsl[CVPR'25] [47]	69.6	63.7	50.8	36.2	50.7	21.4	22.4	8.7	43.7	9.3	12.6	3.4
ClusPro[ICLR'25] [34]	71.0	66.2	54.1	39.5	51.2	**22.1**	23.0	9.3	41.6	8.3	11.6	3.0
SAM-CZSL	**72.9**	**66.7**	**54.8**	**42.3**	**52.9**	21.2	**23.1**	**9.4**	**45.5**	**11.5**	**15.3**	**4.6**

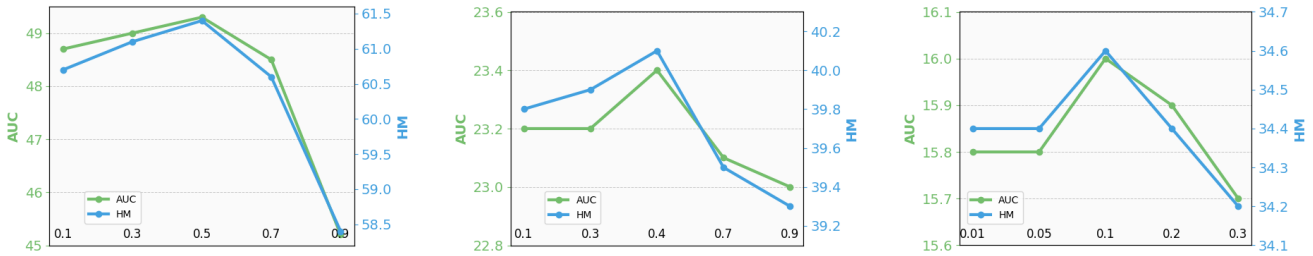
Table 11. The experimental results on open-world settings.

threshold is set to 4, and remain stable unless it goes quite large or small. The test batch size is set to 32, and when size approaches to much larger, *e.g.*, 128, the model steadies decrease performance as update becomes infrequent. We also report the comparison results of inference time and performance with Test-Time Adaptation (TTA) and non-TTA method in Tab. 15 and Tab. 16, respectively.

C. More Qualitative Experiments

More Qualitative Results. Here, we report more qualitative results in UT-Zappos, MIT-States and C-GQA datasets. As shown in Fig. C, our method can predict accurate results where the baseline makes mistakes. For example, baseline is struggle to distinguish similar objects, *e.g.*, "*countertop*" and "*drawer*", "*box*" and "*cooler*". Meanwhile, without filtering redundant information, baseline is misled by extraneous visual content, *e.g.*, baseline focuses on object "*iron fence*", not "*calm water*". These results demonstrate the effectiveness of our method: by suppressing redundant information, our method is able to make more accurate predictions.

More Visualization Results. As shown in



(a) UT-Zappos (b) MIT-States (c) CGQA

Figure 8. Impact of γ across three datasets.

Dataset	α	S	U	HM	AUC
UT-Zappos	0.0	70.3	75.7	58.1	46.2
	0.1	70.9	75.4	59.4	46.9
	0.3	40.6	**77.3**	59.0	47.3
	0.5	**71.6**	76.9	**59.8**	**48.4**
	0.7	71.0	76.0	59.7	47.5
	0.9	70.9	75.4	58.4	46.1
MIT-States	0.0	50.4	52.3	38.9	22.2
	0.1	50.5	52.2	39.0	22.2
	0.3	50.4	52.1	39.0	22.2
	0.5	50.6	52.1	39.3	22.4
	0.7	50.6	52.0	39.2	22.3
	0.9	**50.8**	**52.5**	**39.2**	**22.6**
C-GQA	0.0	70.3	75.7	34.0	15.6
	0.1	45.6	38.7	33.7	15.7
	0.3	45.7	37.9	33.9	15.6
	0.5	**45.5**	**38.7**	**34.3**	**15.8**
	0.7	45.5	38.5	34.1	15.6
	0.9	45.0	38.0	34.1	15.4

Table 12. **Ablation Study on α.**

Fig. 10, we present more visualization results of *Visual Adaptive Condensation* (VAC) module in C-GQA dataset. We can observe that our proposed VAC is capable of excavating critical visual information without disturbing by redundant visual cues, such as, "$bear$" in "$green\ leaf$", "$wall$" in "$mess\ fence$" and "cat" in "$gray\ seat$", where the main objects are more salient and occupy greater space. These results demonstrate the effectiveness of our proposed VAC.

D. Pseudo-code

Training Scheme for SAM. In this section, we provide a detailed training scheme for our proposed SAM framework, which can be divided into three stages. **Stage I: Sparse Alignment**, we conduct sparse alignment between textual representations and patch visual repre-

Dataset	β	S	U	HM	AUC
UT-Zappos	0.1	**72.5**	74.8	59.7	48.0
	0.3	72.1	**76.4**	**60.2**	**48.6**
	0.5	71.2	76.2	58.9	46.7
	0.7	69.9	77.2	59.6	47.4
	0.9	68.4	76.3	57.6	45.5
MIT-States	0.1	49.1	50.5	37.9	20.9
	0.3	49.5	51.6	38.6	21.6
	0.5	50.6	52.1	39.3	22.4
	0.7	**50.7**	**52.9**	**39.3**	**22.7**
	0.9	50.5	52.6	39.2	22.5
C-GQA	0.1	44.5	36.2	32.4	14.4
	0.3	44.9	38.0	33.3	15.2
	0.5	**45.6**	38.6	34.1	15.7
	0.7	45.1	**39.1**	**34.3**	**15.8**
	0.9	45.1	38.2	33.8	15.4

Table 13. **Ablation Study on β.**

τ_{mb}	UT-Zappos		MIT-States		C-GQA	
	HM	AUC	HM	AUC	HM	AUC
0.01	60.7	48.7	39.7	23.1	34.3	15.8
0.1	**61.4**	**49.3**	**40.1**	**23.4**	**34.6**	**16.0**
0.5	61.1	49.1	40.0	23.2	34.5	15.9
1.0	61.0	49.0	39.9	23.2	34.5	15.9

Table 14. Ablation study on temperature of memory bank.

Method	UT-Zappos	MIT-States	C-GQA
SAM	25ms	66ms	177ms
Troika [9]	20ms	61ms	154ms
TOMCAT [51]	47ms	145ms	-

Table 15. Comparison of inference time between TTA and non-TTA methods.

Figure 9. More qualitative results of our method on three datasets.

Method	UT-Zappos		MIT-States		C-GQA	
	HM	AUC	HM	AUC	HM	AUC
TOMCAT[51]	60.2	48.3	39.5	22.6	34.0	16.0
SAM	**62.0**	**50.0**	**40.8**	**24.0**	**34.8**	**16.2**

Table 16. Comparison between SAM and TTA-based methods.

sentations. Leveraging this information-balanced training data, we optimize LoRA [58] for the visual encoder in CLIP. **Stage II: Visual Adaptive Condensation**, with the reduced visual information in the above alignment, the module is guided to adaptively excavate critical visual information within the image, which preserves potential discarded yet valuable information in stage I. **Stage III: Dynamically Updated Memory Bank**, we first initialize memory bank through training data and dynamicall update the memory bank during inference.

	S	U	HM	AUC
K=1	67.6	74.1	57.0	43.7
K=3	**71.2**	**76.2**	**58.9**	**46.7**
K=5	69.5	76.0	57.7	45.5

(a) UT-Zappos

	S	U	HM	AUC
K=1	50.3	52.1	38.6	22.0
K=3	**50.6**	**52.1**	**39.3**	**22.4**
K=5	50.6	51.9	39.1	22.2

(b) MIT-States

	S	U	HM	AUC
K=1	**45.6**	38.6	**34.1**	**15.7**
K=2	45.2	**39.1**	33.9	15.7
K=3	45.7	37.8	33.6	15.4

(c) C-GQA

Table 17. Impact of K in VAC across three datasets.

	S	U	HM	AUC
N=2	72.3	76.2	61.1	48.6
N=4	72.6	76.2	61.1	48.8
N=8	72.3	76.2	60.9	48.8
N=16	**73.1**	**76.2**	**61.0**	**49.2**
N=24	72.9	76.2	60.9	49.0

(a) UT-Zappos

	S	U	HM	AUC
N=4	50.5	52.6	38.7	22.2
N=8	51.4	52.6	39.3	22.6
N=16	51.5	52.6	39.3	22.6
N=24	**51.9**	**52.6**	**39.2**	**22.7**
N=32	51.9	52.6	39.0	22.7

(b) MIT-States

	S	U	HM	AUC
N=2	43.9	38.8	33.5	15.2
N=4	44.7	38.8	33.6	15.4
N=8	44.9	38.8	33.9	15.5
N=16	**45.1**	**38.8**	**34.0**	**15.6**
N=24	45.0	38.8	34.0	15.6

(c) C-GQA

Table 18. Impact of N in memory bank across three datasets.

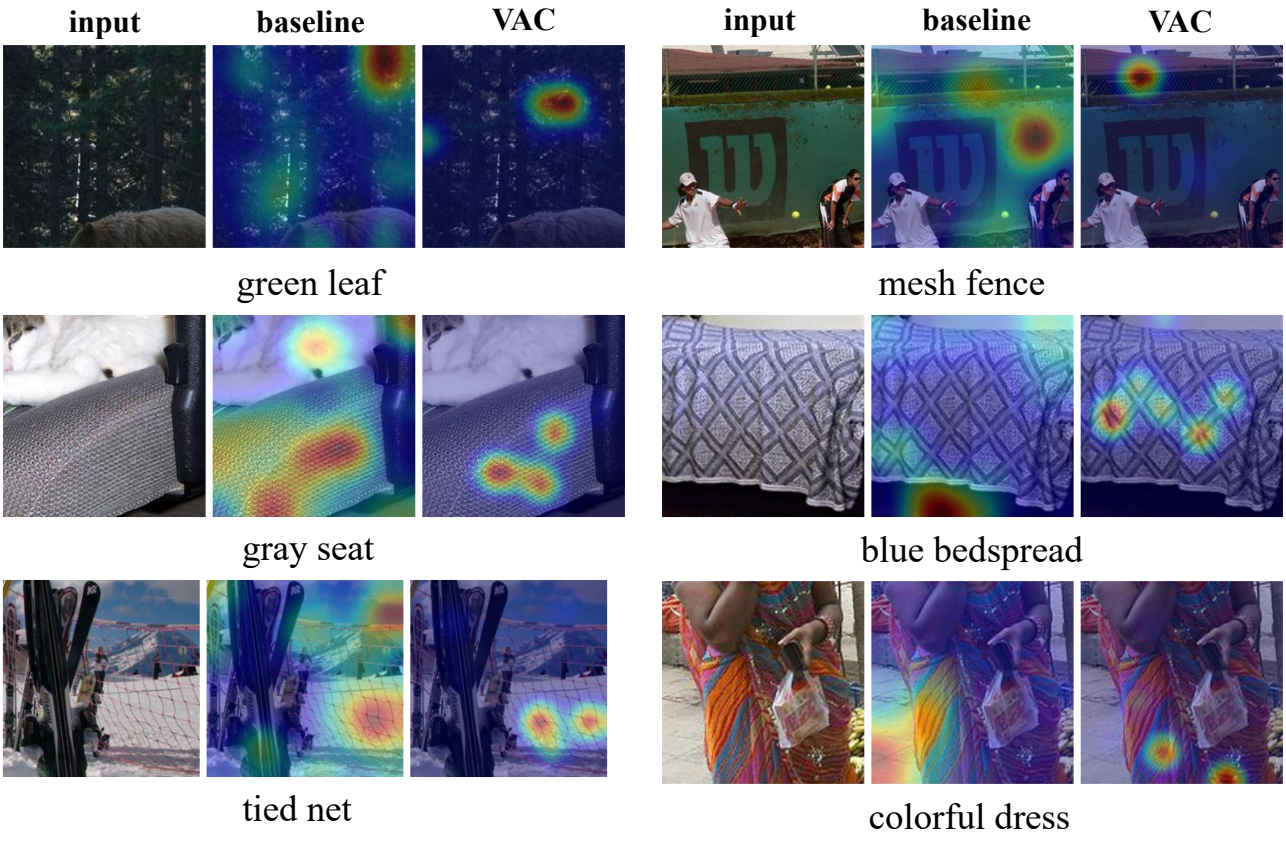


Figure 10. More visualization results of VAC module in C-GQA dataset.

Algorithm 1 Training Scheme for SAM.

Input: training data $\mathcal{D}_{tr}$, visual encoder of CLIP ϕ_{vis}, textual encoder of CLIP ψ_{txt}, learnable soft prompts $\boldsymbol{\theta}_t = [\boldsymbol{\theta}_a,\ \boldsymbol{\theta}_o,\ \boldsymbol{\theta}_{\boldsymbol{c}}]$, visual adaptive condensation module $\boldsymbol{\theta}_{vac}$, LoRA weight $\boldsymbol{\theta}_{LoRA}$, memory bank $\mathbf{B}$.

Output: optimized: LoRA weight $\boldsymbol{\theta}_{LoRA}$, learnable soft prompts $\boldsymbol{\theta}_t = [\boldsymbol{\theta}_a,\ \boldsymbol{\theta}_o,\ \boldsymbol{\theta}_{\boldsymbol{c}}]$, visual adaptive condensation module $\boldsymbol{\theta}_{vac}$; updated memory bank $\mathbf{B}$.

Stage I:, randomly initialize parameters $\boldsymbol{\theta}_{LoRA}$; load pre-trained parameters visual encoder of CLIP ϕ_{vis}, textual encoder of CLIPψ_{txt}, learnable soft prompts $[\boldsymbol{\theta}_a,\ \boldsymbol{\theta}_o,\ \boldsymbol{\theta}_{\boldsymbol{c}}]$.
while *not converged* **do**
  batch of training data $(\mathcal{X}_b, \mathcal{Y}_b)$
  conducting sparse alignment by visual reduction in Eq. 2
  calculating basic learning objective $\mathcal{L}_{base}$ in Eq. 4
  optimize parameters $\boldsymbol{\theta}$ $(\boldsymbol{\theta}_{LoRA},\ \boldsymbol{\theta}_t) = \boldsymbol{\theta} - \nabla_{\boldsymbol{\theta}}(\mathcal{L}_{base}(\mathcal{X}_b, \mathcal{Y}_b; \boldsymbol{\theta}))$
end while
Stage II: randomly initialize parameters $\boldsymbol{\theta}_{VAC}$.
while *not converged* **do**
  batch of training data $(\mathcal{X}_b, \mathcal{Y}_b)$
  condense visual information within image into $\boldsymbol{v}_q$
  calculation prediction $\boldsymbol{p}_{vac}$ of VAC by Eq. 5 and $\boldsymbol{p}_{sa}$ of SA by Eq. 3
  calculating learning objective $\mathcal{L}_{base}^{vac}$ in Eq. 6 and $\mathcal{L}_{kl}$ in Eq. 7
  optimize parameters $\boldsymbol{\theta}$ $(\boldsymbol{\theta}_{vac}) = \boldsymbol{\theta} - \nabla_{\boldsymbol{\theta}}((1-\alpha) \cdot \mathcal{L}_{base}^{vac}(\mathcal{X}_b, \mathcal{Y}_b; \boldsymbol{\theta}) + \alpha \cdot \mathcal{L}_{kl}(\mathcal{X}_b, \mathcal{Y}_b; \boldsymbol{\theta}))$
end while
Stage III: initialize stored samples for seen compositions in memory bank $\mathbf{B}$ by Eq. 9.
for batch of testing data $\mathcal{X}_b$ **do**
  calculating predictions $\boldsymbol{p}_{sa}$, $\boldsymbol{p}_{vac}$ and $\boldsymbol{p}_{bank}$ from three modules by Eq. 3, Eq. 5 and Eq. 10, respectively
  obtain final prediction by Eq. 11
  utilizing $\boldsymbol{p}_{vac}$ to update memory bank by Eq. 9
end for
calculating the results of each evaluation metric with the final prediction
return optimized LoRA weight $\boldsymbol{\theta}_{LoRA}$, learnable soft prompts $\boldsymbol{\theta}_t = [\boldsymbol{\theta}_a,\ \boldsymbol{\theta}_o,\ \boldsymbol{\theta}_{\boldsymbol{c}}]$, visual adaptive condensation module $\boldsymbol{\theta}_{vac}$; updated memory bank $\mathbf{B}$.