Text-Phase Synergy Network with Dual Priors for Unsupervised Cross-Domain Image Retrieval

Supplementary Material

In the technical appendices and supplementary material, we provide:

- Additional implementation details of the proposed methods.
- Further experimental results and ablation studies.
- A broader discussion of the limitations of this work.

A. Implementation Details

A.1. Re-Paired Image and Text Embeddings.

In DPG, the loss in Eq.(1) and Eq.(2) are computed between re-paired image and text embeddings. The re-pairing process is performed dynamically during each training iteration based on the cosine similarity, defined as:

$$s(I_i, T_c) = \frac{I_i^\top T_c}{\|I_i\|\|T_c\|}. \tag{13}$$

Here, I_i denotes the image embedding, and T_c represents the c-th domain prompt embedding, where $c \in 1, \ldots, C$ and C is the number of domain prompts. Subsequently, the image embedding is paired with the text embedding corresponding to the domain prompt with the highest similarity:

$$y_i = \arg\max_c s(I_i, T_c), \quad T_i = T_{y_i}. \tag{14}$$

Thus, the re-paired image-text pair (I_i, T_i) is dynamically updated at each iteration to maximize semantic consistency. This strategy enhances the quality of pseudo-labels and facilitates improved semantic representation learning despite domain discrepancies.

A.2. Phase Feature Encoder.

In PPFE, we utilize the Phase Feature Encoder to extract phase features. Specifically, after applying the Fast Fourier Transform (FFT), we retain the phase spectrum while replacing the amplitude spectrum with a constant value R, which helps preserve domain-invariant information. To investigate the influence of the constant value on the reconstructed image, we analyze the Inverse Fast Fourier Transform (IFFT) of a complex signal constructed as $R \cdot e^{j\phi(u,v)}$. Due to the linearity of the $\mathcal{F}^{-1}$, we obtain:

$$\mathcal{F}^{-1}[R \cdot e^{j\phi(u,v)}] = R \cdot \mathcal{F}^{-1}[e^{j\phi(u,v)}], \tag{15}$$

which demonstrates that the reconstructed image is a scaled version of the phase-only image, with its structure entirely determined by the phase component.

Next, we apply min-max normalization to the reconstructed image. For an image $\hat{x}^{phase}$, the min-max normalization is defined as:

$$x^{phase} = \frac{\hat{x}^{phase} - \min(\hat{x}^{phase})}{\max(\hat{x}^{phase}) - \min(\hat{x}^{phase})}. \tag{16}$$

When we apply the same normalization to the image reconstructed with the constant amplitude R, we have:

$$x^{phase'} = \frac{R \cdot \hat{x}^{phase} - \min(R \cdot \hat{x}^{phase})}{\max(R \cdot \hat{x}^{phase}) - \min(R \cdot \hat{x}^{phase})} \tag{17}$$

$$= \frac{R \cdot (\hat{x}^{phase} - \min(\hat{x}^{phase}))}{R \cdot (\max(\hat{x}^{phase}) - \min(\hat{x}^{phase}))} \tag{18}$$

$$= x^{phase}. \tag{19}$$

Thus, we observe that the normalized result $x^{phase'}$ is invariant to the choice of R, as long as $R \neq 0$. Therefore, under min-max normalization, the image structure is solely determined by the phase component, while any non-zero constant magnitude R preserves the structural semantics and eliminates domain-specific amplitude variations.

B. Additional Experiments

B.1. Domain Prompt Generation Module.

The Domain Prompt Generation Module involves prompt template design and the impact of K-means clustering on pseudo-label quality, which merit further discussion.

Prompt Templates. We use the template "An image of a $[X]^1 \ldots [X]^M$" for each domain, where the learnable tokens are category-specific and not shared across classes. We analyze category-agnostic compared with category-specific prompts, as well as the impact of the number of learnable tokens M.

In fully unsupervised settings, category names are unavailable. We therefore adopt category-specific phrasing (e.g., "An image of a") while learning category-conditioned prompt tokens. We further analyze a fully category-agnostic design, where both phrasing and prompt tokens are shared across all categories. In this case, performance degrades: P@50 drops by 3.3% on DomainNet, and P@1 drops by 9.8% on Office-Home (65 categories). This result indicates that category-specific prompts are crucial for maintaining discriminability, especially in scenarios with a large number of categories.

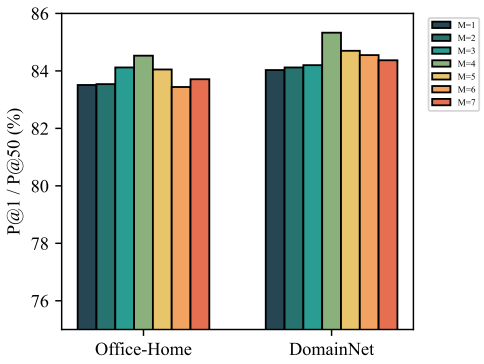

Figure 7. Analysis on the number of learnable text tokens M on two datasets.

We analyze the impact of the number of learnable text tokens M on retrieval performance. As illustrated in Figure 7, the model achieves optimal performance on both datasets when $M = 4$. A small value of M may be insufficient to effectively capture rich class-specific semantics and provide adequate supervisory signals. Conversely, setting M too large can result in overfitting of the text prompts and a substantial increase in training time, ultimately hindering generalization.

Pseudo-label Quality. Following existing works, we set the number of clusters K equal to the number of classes in the training set, i.e., 65 for Office-Home and 7 for DomainNet. Therefore, the issue of incorrect number of clusters is avoided during training.

To further examine the impact of using an incorrect number of clusters, we conducted additional experiments on DomainNet by setting K to 5 and 9 instead of the ground-truth value of 7. The average P@50 accuracy dropped to 84.59% and 85.01%, respectively, compared to 85.33% under the correct setting. These results indicate that using an inappropriate cluster number can degrade retrieval performance due to inaccurate pseudo-labels and unreliable domain prompts. Nonetheless, the overall performance remains competitive. We also observe that setting K smaller than the actual number of classes tends to result in more severe misclustering, as multiple distinct categories are forced into a single cluster. In contrast, using a larger K may split some categories into multiple clusters, which is generally less detrimental.

To investigate how misclassified points affect the final performance, we evaluate our method under a fully supervised setting, removing the influence of pseudo-label noise. The results on the DomainNet dataset indicate that under the unsupervised setting, where the clustering accuracy is approximately 85%, the fully supervised counterpart achieves only a modest improvement of 1.64% in P@50. This confirms that misclassified instances do affect final performance. However, the relatively small performance gap suggests that our unsupervised framework is sufficiently robust to mitigate the negative effects of noisy pseudo-labels. This robustness can be attributed to the following: although some images are incorrectly clustered, the domain prompts are primarily influenced by the majority of correctly clustered samples, resulting in semantically meaningful representations. Consequently, our method achieves competitive retrieval performance without relying on manual annotations.

B.2. Ablation Study on the impact of the Phase Prior.

To validate the effectiveness of the phase prior, we conduct an ablation study comparing full-spectrum phase features with high-frequency (HF) and low-frequency (LF) phase variants in the fusion process. As shown in Table 5, both HF and LF phase priors improve retrieval performance by reducing cross-domain discrepancies while preserving certain semantic cues. However, full-spectrum phase features consistently achieve superior performance. This is because the complete phase representation jointly encodes global spatial layout (low-frequency components) and local structural boundaries (high-frequency components), thereby preserving more holistic and domain-invariant structural semantics.

Furthermore, to investigate the effect of different fusion strategies for integrating the phase prior, we explore four distinct methods to combine phase features with original image features: concatenation, learnable weights, gate-based fusion, and self-attention. As presented in Table 6, the self-attention-based fusion achieves the best performance, highlighting its superior ability to adaptively emphasize informative components and facilitate robust domain-invariant representation learning.

B.3. Ablation Study on the Impact of Vision-Language Pretraining Architectures.

We further investigate the impact of different vision-language pretraining architectures on the performance of TPSNet. Specifically, we compare CLIP [30], BLIP [21], SigLIP [46] and SigLIP2 [35] as backbone models. The experimental results on the OfficeHome and DomainNet datasets are presented in Table 7 and Table 8, with the average performance summarized in Table 9. Empirical results reveal that TPSNet achieves the best performance with CLIP as the backbone. In contrast, its performance noticeably degrades when using BLIP. We attribute this to BLIP's multi-task pretraining objective, which incorporates both image-text matching and image-conditioned text generation. While such generative objectives enhance performance in tasks like image captioning and visual question answering, they may limit the model's ability to learn fine-

Table 5. Average accuracy (%) for ablation study on phase prior.

Method	Office-Home			DomainNet		
	$P@1$	$P@5$	$P@15$	$P@50$	$P@100$	$P@200$
Baseline (w/o Prior)	81.63	80.13	77.72	82.05	79.94	75.97
TPSNet (LF Prior)	83.71	81.06	78.19	84.25	83.02	80.10
TPSNet (HF Prior)	84.02	82.99	80.78	84.37	83.22	80.41
TPSNet (Phase Prior)	**84.53**	**83.44**	**81.29**	**85.33**	**84.34**	**81.91**

Table 6. Ablation study on fusion strategies of the phase prior on the Office-Home dataset (Concat: concatenation, LW: learnable weights, Gate: gate-based fusion, SA: self-attention).

Method		$Art \rightarrow Real$			$Real \rightarrow Art$			$Art \rightarrow Product$			$Product \rightarrow Art$		
		$P@1$	$P@5$	$P@15$	$P@1$	$P@5$	$P@15$	$P@1$	$P@5$	$P@15$	$P@1$	$P@5$	$P@15$
ViT-B	Concat	88.50	88.13	87.23	91.30	**91.26**	**87.05**	78.78	77.68	76.14	87.02	85.20	79.04
	LW	82.86	81.61	79.84	86.39	84.18	78.17	73.01	71.16	67.74	80.83	78.87	70.39
	Gate	84.14	82.51	80.56	87.65	85.00	79.25	77.96	76.48	74.73	79.03	79.04	72.09
	SA	**89.53**	**88.61**	**87.77**	**91.74**	90.71	86.81	**81.17**	**80.60**	**79.43**	**88.80**	**86.14**	**81.01**
Method		$Clipart \rightarrow Real$			$Real \rightarrow Clipart$			$Product \rightarrow Real$			$Real \rightarrow Product$		
		$P@1$	$P@5$	$P@15$	$P@1$	$P@5$	$P@15$	$P@1$	$P@5$	$P@15$	$P@1$	$P@5$	$P@15$
ViT-B	Concat	73.20	72.41	71.79	88.41	87.17	84.31	91.60	91.77	91.38	90.57	90.88	90.03
	LW	67.40	64.56	62.16	84.00	82.24	77.66	89.93	87.59	85.44	86.71	85.87	84.87
	Gate	68.98	66.72	64.59	85.54	83.59	79.59	90.61	88.77	86.11	87.19	86.30	84.95
	SA	**73.36**	**72.73**	**71.85**	**89.03**	**88.01**	**84.86**	**93.26**	**92.82**	**91.78**	**92.10**	**92.05**	**91.78**
Method		$Product \rightarrow Clipart$			$Clipart \rightarrow Product$			$Art \rightarrow Clipart$			$Clipart \rightarrow Art$		
		$P@1$	$P@5$	$P@15$	$P@1$	$P@5$	$P@15$	$P@1$	$P@5$	$P@15$	$P@1$	$P@5$	$P@15$
ViT-B	Concat	**88.98**	**87.28**	84.46	72.30	**72.90**	**72.74**	**84.71**	**82.49**	79.17	69.14	67.44	64.07
	LW	84.84	83.40	80.74	66.16	65.67	64.47	78.99	76.41	73.16	62.47	59.97	55.52
	Gate	84.95	83.37	80.16	67.40	66.05	64.93	79.23	75.14	70.37	62.66	59.58	53.79
	SA	88.31	86.74	**84.49**	**72.67**	72.30	71.79	84.55	81.94	**79.36**	**69.87**	**68.67**	**64.55**

grained, discriminative representations that are critical for cross-domain image retrieval tasks.

Similarly, SigLIP also exhibits inferior performance compared to CLIP, despite adopting a contrastive learning framework. We attribute this discrepancy to differences in their loss function formulations. Specifically, SigLIP employs a sigmoid-based contrastive loss that treats each image-text pair independently, whereas CLIP utilizes a softmax-based loss that jointly considers all positive and negative pairs within a batch. This global contrastive objective in CLIP fosters stronger alignment and discrimination between modalities, which in turn enhances the model's capacity to learn semantically consistent representations across domains. SigLIP2 [35], an improved variant of SigLIP, mitigates some of these limitations by introducing architectural enhancements and refined training strategies, leading to consistently better performance than SigLIP across two datasets. However, its performance remains inferior to CLIP, primarily because the sigmoid-based contrastive loss fails to capture the global batch-level interactions inherent in CLIP's softmax-based objective.

B.4. Ablation Study on the Impact of Image Encoder Initialization.

To evaluate the effectiveness of TPSNet without attributing the observed improvements to the CLIP image encoder, we replace the CLIP pre-trained image encoder with MoCov2 [6] and DINO [5], which have been widely adopted in previous works [13, 20, 38]. This setup ensures that any performance gains are solely due to the effectiveness of TPSNet itself. The detailed experimental results are provided in Tables 10 and 11.

As demonstrated in Table 10 and Table 11, even when MoCov2 and DINO are employed as initialization param-

Table 7. Ablation study on vision-language pretraining architectures on the Office-Home dataset. The table reports results for each of the 12 cross-domain retrieval scenarios individually.

	Method	*Art* → *Real*			*Real* → *Art*			*Art* → *Product*			*Product* → *Art*		
		P@1	*P*@5	*P*@15	*P*@1	*P*@5	*P*@15	*P*@1	*P*@5	*P*@15	*P*@1	*P*@5	*P*@15
ViT-B	TPSNet (BLIP) [21]	50.27	49.54	47.89	58.43	53.34	47.45	43.68	43.79	42.68	55.67	49.70	42.62
	TPSNet (SigLIP) [46]	65.14	63.63	61.62	66.01	62.36	56.33	57.11	53.01	47.67	44.60	42.53	37.48
	TPSNet (SigLIP2) [35]	67.11	66.25	62.21	68.31	66.87	63.23	62.11	60.12	58.97	47.54	45.33	44.91
	TPSNet (CLIP)	**89.53**	**88.61**	**87.77**	**91.74**	**90.71**	**86.81**	**81.17**	**80.60**	**79.43**	**88.80**	**86.14**	**81.01**
	Method	*Clipart* → *Real*			*Real* → *Clipart*			*Product* → *Real*			*Real* → *Product*		
		P@1	*P*@5	*P*@15	*P*@1	*P*@5	*P*@15	*P*@1	*P*@5	*P*@15	*P*@1	*P*@5	*P*@15
ViT-B	TPSNet (BLIP) [21]	26.03	23.89	22.26	50.40	45.69	38.95	72.70	69.48	65.94	67.04	65.27	63.13
	TPSNet (SigLIP) [46]	36.49	35.17	33.50	61.33	57.89	51.34	58.84	54.87	51.82	67.82	64.78	60.38
	TPSNet (SigLIP2) [35]	38.21	37.01	35.03	62.09	61.77	60.01	60.00	59.04	57.77	70.32	68.43	66.09
	TPSNet (CLIP)	**73.36**	**72.73**	**71.85**	**89.03**	**88.01**	**84.86**	**93.26**	**92.82**	**91.78**	**92.10**	**92.05**	**91.78**
	Method	*Product* → *Clipart*			*Clipart* → *Product*			*Art* → *Clipart*			*Clipart* → *Art*		
		P@1	*P*@5	*P*@15	*P*@1	*P*@5	*P*@15	*P*@1	*P*@5	*P*@15	*P*@1	*P*@5	*P*@15
ViT-B	TPSNet (BLIP) [21]	56.14	49.97	42.35	24.56	24.33	24.09	35.15	31.91	28.22	18.99	17.23	14.92
	TPSNet (SigLIP) [46]	45.57	43.77	39.08	32.97	32.20	29.92	55.79	52.01	45.87	30.65	28.20	25.31
	TPSNet (SigLIP2) [35]	49.32	48.88	47.61	33.21	32.90	30.05	57.09	54.87	53.21	32.64	30.40	28.65
	TPSNet (CLIP)	**88.31**	**86.74**	**84.49**	**72.67**	**72.30**	**71.79**	**84.55**	**81.94**	**79.36**	**69.87**	**68.67**	**64.55**

Table 8. Ablation study on vision-language pretraining architectures on the Domainnet dataset. The table reports results for each of the 12 cross-domain retrieval scenarios individually.

	Method	*Clipart* → *Sketch*			*Sketch* → *Clipart*			*Infograph* → *Real*			*Real* → *Infograph*		
		P@50	*P*@100	*P*@200	*P*@50	*P*@100	*P*@200	*P*@50	*P*@100	*P*@200	*P*@50	*P*@100	*P*@200
ViT-B	TPSNet (BLIP) [21]	81.48	79.61	72.53	77.04	75.26	68.83	58.33	58.21	57.84	87.03	78.33	64.53
	TPSNet (SigLIP) [46]	84.37	79.48	73.07	81.02	76.99	70.93	63.19	62.90	62.43	73.55	64.03	50.73
	TPSNet (SigLIP2) [35]	89.42	86.18	81.06	88.24	85.88	80.63	73.02	72.85	72.67	80.28	78.06	66.05
	TPSNet (CLIP)	**97.92**	**97.84**	**97.73**	**98.27**	**98.20**	**98.06**	**78.72**	**78.55**	**78.31**	**93.01**	**93.04**	**85.71**
	Method	*Infograph* → *Sketch*			*Sketch* → *Infograph*			*Painting* → *Clipart*			*Clipart* → *Painting*		
		P@50	*P*@100	*P*@200	*P*@50	*P*@100	*P*@200	*P*@50	*P*@100	*P*@200	*P*@50	*P*@100	*P*@200
ViT-B	TPSNet (BLIP) [21]	50.91	48.61	43.12	59.20	52.93	44.21	86.19	84.71	76.74	81.09	79.39	75.76
	TPSNet (SigLIP) [46]	59.46	56.58	53.44	70.11	65.79	62.02	88.79	81.97	79.91	86.50	84.51	81.63
	TPSNet (SigLIP2) [35]	64.64	61.40	55.93	73.45	70.22	64.87	94.83	88.82	88.25	89.42	88.75	87.81
	TPSNet (CLIP)	**75.87**	**75.61**	**75.03**	**95.21**	**92.04**	**84.78**	**98.49**	**98.45**	**98.26**	**97.84**	**97.73**	**96.64**
	Method	*Painting* → *Quickdraw*			*Quickdraw* → *Painting*			*Quickdraw* → *Real*			*Real* → *Quickdraw*		
		P@50	*P*@100	*P*@200	*P*@50	*P*@100	*P*@200	*P*@50	*P*@100	*P*@200	*P*@50	*P*@100	*P*@200
ViT-B	TPSNet (BLIP) [21]	32.06	30.40	28.18	23.75	22.97	22.07	22.34	23.08	23.36	34.92	32.75	30.16
	TPSNet (SigLIP) [46]	37.21	36.46	35.51	18.47	17.49	16.69	19.60	18.60	17.71	26.48	25.84	24.67
	TPSNet (SigLIP2) [35]	37.04	36.22	35.04	17.22	17.23	17.36	17.66	17.21	17.45	32.45	30.56	29.04
	TPSNet (CLIP)	**81.67**	**77.04**	**71.34**	**65.12**	**64.51**	**63.31**	**62.50**	**62.21**	**61.56**	**79.34**	**76.89**	**72.14**

eters for the image encoder, TPSNet consistently outperforms current state-of-the-art methods on both datasets. Furthermore, when the image encoder is initialized with parameters pre-trained by CLIP, TPSNet's performance is further enhanced.

B.5. Dynamic Prompt Generation vs. Our Domain Prompts

For prompt generation, we also considered alternative learnable prompt-generation strategies. Regarding the use of pre-trained vision–language models to automatically refine prompts, we experimented with employing QwenVL to generate class-relevant prompts directly from images. However, we found that such models often produce semantically inconsistent textual outputs for instances within the same

Table 9. Average accuracy (%) compared with universal retrieval models on the Office-Home and DomainNet Datasets.

Method	Office-Home			DomainNet		
	$P@1$	$P@5$	$P@15$	$P@50$	$P@100$	$P@200$
TPSNet (BLIP) [21]	46.59	43.68	40.04	57.86	55.52	50.61
TPSNet (SigLIP) [46]	51.86	49.20	45.03	59.06	55.89	52.40
TPSNet (SigLIP2) [35]	54.00	52.66	50.65	63.14	61.12	58.01
TPSNet (CLIP)	**84.53**	**83.44**	**81.29**	**85.33**	**84.34**	**81.91**

Table 10. Ablation study on the impact of image encoder initialization on the Office-Home dataset. The table reports results for each of the 12 cross-domain retrieval scenarios individually.

	Method	*Art* → *Real*			*Real* → *Art*			*Art* → *Product*			*Product* → *Art*		
		$P@1$	$P@5$	$P@15$	$P@1$	$P@5$	$P@15$	$P@1$	$P@5$	$P@15$	$P@1$	$P@5$	$P@15$
ResNet-50	ShieldIR (SOTA) [34]	51.46	50.20	49.27	57.33	53.50	49.26	44.75	46.16	45.33	56.31	52.71	47.39
	TPSNet (MoCov2)	60.94	60.06	59.04	65.89	62.10	57.50	53.07	50.67	48.89	60.24	56.92	49.49
	TPSNet	**69.06**	**67.22**	**65.02**	**74.00**	**69.72**	**62.73**	**59.41**	**57.21**	**53.50**	**63.82**	**59.98**	**52.31**
ViT-B	SA-MoE (SOTA) [38]	71.12	68.93	66.10	73.86	68.85	60.37	64.69	62.39	58.63	66.57	62.82	55.12
	TPSNet (DINO)	73.22	71.54	70.50	79.27	76.15	69.79	64.98	64.23	62.85	74.32	69.94	63.11
	TPSNet	**89.53**	**88.61**	**87.77**	**91.74**	**90.71**	**86.81**	**81.17**	**80.60**	**79.43**	**88.80**	**86.14**	**81.01**
	Method	*Clipart* → *Real*			*Real* → *Clipart*			*Product* → *Real*			*Real* → *Product*		
		$P@1$	$P@5$	$P@15$	$P@1$	$P@5$	$P@15$	$P@1$	$P@5$	$P@15$	$P@1$	$P@5$	$P@15$
ResNet-50	ShieldIR (SOTA) [34]	43.36	42.58	41.55	52.05	52.38	50.81	70.37	68.33	66.08	61.46	62.08	61.85
	TPSNet (MoCov2)	46.30	45.45	44.66	59.03	57.49	54.19	76.57	73.56	71.22	71.15	69.66	68.25
	TPSNet	**51.68**	**49.91**	**47.56**	**63.53**	**61.36**	**57.40**	**78.85**	**75.87**	**72.48**	**74.39**	**73.00**	**70.94**
ViT-B	SA-MoE (SOTA) [38]	52.99	50.00	47.33	70.41	65.78	60.06	78.91	74.96	70.40	76.52	74.54	71.18
	TPSNet (DINO)	54.41	52.77	51.86	73.35	70.38	66.94	82.18	80.61	78.45	79.94	78.32	76.59
	TPSNet	**73.36**	**72.73**	**71.85**	**89.03**	**88.01**	**84.86**	**93.26**	**92.82**	**91.78**	**92.10**	**92.05**	**91.78**
	Method	*Product* → *Clipart*			*Clipart* → *Product*			*Art* → *Clipart*			*Clipart* → *Art*		
		$P@1$	$P@5$	$P@15$	$P@1$	$P@5$	$P@15$	$P@1$	$P@5$	$P@15$	$P@1$	$P@5$	$P@15$
ResNet-50	ShieldIR (SOTA) [34]	52.10	52.43	50.23	40.84	39.87	39.45	41.24	39.05	37.44	32.21	30.13	28.39
	TPSNet (MoCov2)	59.92	**57.74**	**56.20**	42.54	42.68	**42.37**	48.53	45.74	42.67	36.38	33.58	31.20
	TPSNet	**60.71**	57.49	53.26	**46.05**	**43.77**	42.22	**56.98**	**54.07**	**50.29**	**43.14**	**39.66**	**35.58**
ViT-B	SA-MoE (SOTA) [38]	65.28	61.09	56.01	47.47	45.81	43.36	61.80	57.82	53.23	46.90	44.39	39.97
	TPSNet (DINO)	72.38	68.05	64.58	49.90	49.23	47.57	63.12	60.66	56.65	44.42	41.87	38.56
	TPSNet	**88.31**	**86.74**	**84.49**	**72.67**	**72.30**	**71.79**	**84.55**	**81.94**	**79.36**	**69.87**	**68.67**	**64.55**

class. For example, images labeled as "bird" may yield prompts such as "a parrot" or "a hummingbird", leading to semantic fragmentation that severely undermines retrieval consistency. Moreover, these methods typically depend on manually crafted templates (e.g., "What is the specific object in this image?") to guide prompt generation, which introduces additional engineering overhead. We also observed that each inference takes approximately 0.91 seconds and consumes more than 4 GB of GPU memory per image, rendering this approach computationally expensive and impractical for large-scale unsupervised datasets.

Therefore, in our setting, we adopt a learnable prompt format of "An image of a $[X]^1[X]^2 \dots [X]^M$", which does not rely on class labels. For each class, we generate a more descriptive yet consistent prompt expression that avoids the semantic fragmentation issue of dynamically generated prompts while remaining robust across domains. We believe this design provides a practical and effective balance between performance and training cost.

B.6. Limitations of the LMM-based Universal Retrieval Models for UCDIR.

In addition, we compare TPSNet with the current state-of-the-art large multimodal models (LMMs) for retrieval performance, including InternVL3.5 [41], SAIL-VL2 [45], MM-Embed [23], Qwen2.5VL [3], and LamRA [25]. The average retrieval performance is summarized in Table 12. Since InternVL3.5 and SAIL-VL2 are primarily designed

Table 11. Ablation study on the impact of image encoder initialization on the Domainnet dataset. The table reports results for each of the 12 cross-domain retrieval scenarios individually.

	Method	*Clipart* → *Sketch*			*Sketch* → *Clipart*			*Infograph* → *Real*			*Real* → *Infograph*		
		P@50	*P*@100	*P*@200	*P*@50	*P*@100	*P*@200	*P*@50	*P*@100	*P*@200	*P*@50	*P*@100	*P*@200
ResNet-50	ShieldIR (SOTA) [34]	87.16	87.12	86.74	89.95	89.54	88.56	54.41	54.93	55.34	82.26	75.35	64.15
	TPSNet (MoCov2)	**95.64**	**95.85**	**95.93**	**96.13**	**96.08**	**96.03**	47.28	47.46	47.40	83.82	76.42	61.52
	TPSNet	95.43	95.31	94.63	95.50	95.31	94.26	**70.04**	**69.55**	**68.74**	**86.07**	**81.91**	**71.99**
ViT-B	SA-MoE (SOTA) [38]	83.97	82.08	78.40	88.11	86.07	81.21	57.29	57.55	57.67	87.37	80.08	66.24
	TPSNet (DINO)	96.18	96.27	96.19	97.18	97.13	96.82	61.54	61.47	61.35	90.21	83.78	69.13
	TPSNet	**97.92**	**97.84**	**97.73**	**98.27**	**98.20**	**98.06**	**78.72**	**78.55**	**78.31**	**93.01**	**93.04**	**85.71**
	Method	*Infograph* → *Sketch*			*Sketch* → *Infograph*			*Painting* → *Clipart*			*Clipart* → *Painting*		
		P@50	*P*@100	*P*@200	*P*@50	*P*@100	*P*@200	*P*@50	*P*@100	*P*@200	*P*@50	*P*@100	*P*@200
ResNet-50	ShieldIR (SOTA) [34]	47.39	47.10	46.34	72.29	67.68	55.00	93.96	93.47	90.99	85.32	85.34	84.30
	TPSNet (MoCov2)	46.86	46.66	45.82	82.04	73.52	58.22	**95.50**	**95.00**	**93.95**	92.83	92.55	**91.70**
	TPSNet	**67.59**	**67.00**	**65.96**	**90.20**	**86.27**	**74.90**	94.07	93.27	90.61	**94.56**	**93.99**	91.08
ViT-B	SA-MoE (SOTA) [38]	53.76	53.07	50.37	79.56	72.29	58.62	95.00	94.59	93.23	90.60	90.42	89.25
	TPSNet (DINO)	60.03	59.92	59.44	91.94	85.83	69.64	97.89	97.68	96.77	97.11	96.89	95.29
	TPSNet	**75.87**	**75.61**	**75.03**	**95.21**	**92.04**	**84.78**	**98.49**	**98.45**	**98.26**	**97.84**	**97.73**	**96.64**
	Method	*Painting* → *Quickdraw*			*Quickdraw* → *Painting*			*Quickdraw* → *Real*			*Real* → *Quickdraw*		
		P@50	*P*@100	*P*@200	*P*@50	*P*@100	*P*@200	*P*@50	*P*@100	*P*@200	*P*@50	*P*@100	*P*@200
ResNet-50	ShieldIR (SOTA) [34]	65.33	61.25	57.98	58.07	57.42	**55.11**	67.19	67.36	67.53	**75.70**	73.60	70.63
	TPSNet (MoCov2)	64.56	63.20	61.86	46.74	48.92	49.88	65.81	66.40	66.97	71.26	71.54	70.75
	TPSNet	**67.21**	**65.38**	**61.88**	**59.27**	**57.93**	54.30	**69.34**	**68.88**	**68.17**	75.06	**74.70**	**72.96**
ViT-B	SA-MoE (SOTA) [38]	72.54	71.16	69.22	61.32	61.21	61.03	55.78	55.59	55.49	59.02	57.55	55.81
	TPSNet (DINO)	69.00	67.33	64.19	62.70	60.42	56.22	**68.98**	**67.80**	**65.70**	78.93	**77.04**	**75.04**
	TPSNet	**81.67**	**77.04**	**71.34**	**65.12**	**64.51**	**63.31**	62.50	62.21	61.56	**79.34**	76.89	72.14

for question-answering tasks, we use only their pre-trained vision encoders. All other methods utilize the complete output vectors of LMMs as feature representations. Experimental results demonstrate that, although LMM-based universal retrieval models exhibit strong zero-shot generalization ability, there remains a noticeable performance gap compared to TPSNet, particularly in scenarios involving significant domain discrepancies. Specifically, their performance deteriorates significantly in domains with substantial background noise and abstract visual patterns, such as Infograph and Quickdraw. This degradation can be attributed to the inherent bias of foundation models towards natural-image domains prevalent in their pretraining datasets, which limits their ability to generalize to highly stylized domains.

Therefore, despite their strong generalization capacity, universal retrieval models still fall short in effectively addressing the specific challenges of UCDIR. Moreover, these models typically require a substantial number of parameters, whereas our TPSNet, based on ViT-B, achieves superior performance with a significantly reduced parameter count. This demonstrates the necessity and practical value of developing targeted methods tailored for UCDIR, especially in scenarios with severe domain discrepancies.

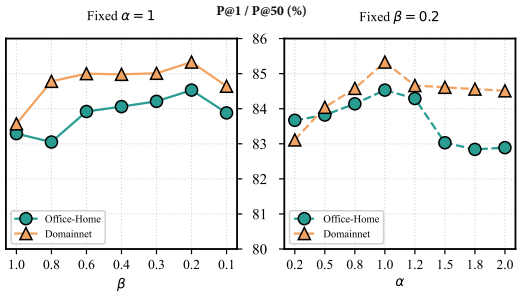

Figure 8. Analysis on coefficients α and β on two datasets.

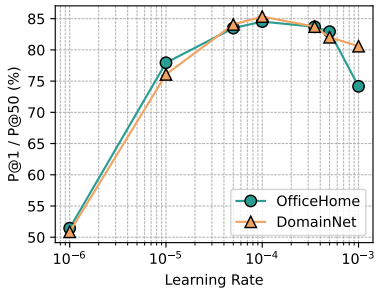

Figure 9. Analysis on learning rate.

B.7. Analysis on Coefficients α and β.

To investigate the impact of the coefficients α and β in Eq.(12), we conduct hyperparameter analysis experiments on Office-Home and Domainnet datasets, as shown in Figure 8. The model achieves the best performance when

Table 12. Average accuracy (%) compared with the LMM-based universal retrieval models on the Office-Home and DomainNet Datasets.

Method	Office-Home			DomainNet		
	$P@1$	$P@5$	$P@15$	$P@50$	$P@100$	$P@200$
InternVL3.5 - 6B [41]	66.08	60.88	53.81	58.71	54.82	47.11
SAIL-VL2 - 0.6B [45]	73.08	67.47	58.76	58.29	53.38	46.16
MM-Embed - 7B [23]	79.38	76.20	71.01	78.05	75.18	70.27
Qwen2.5VL - 3B [3]	77.84	75.01	69.40	78.00	77.25	75.81
Qwen2.5VL - 7B [3]	82.56	80.53	76.12	81.45	81.47	80.44
LamRA - 7B [25]	84.20	82.60	79.75	80.77	78.39	74.80
TPSNet - 0.086B	**84.53**	**83.44**	**81.29**	**85.33**	**84.34**	**81.91**

$\alpha = 1$ and $\beta = 0.2$.

B.8. Analysis on Learning Rate.

We conduct a learning rate sensitivity analysis on both the Office-Home and DomainNet datasets to examine its impact on model performance. As shown in Figure 9, the best performance is achieved when the learning rate is set to 10^{-4}.

B.9. More Visualization Results

Here, we present additional t-SNE and Grad-CAM visualizations across more scenarios and samples in Figure 10, demonstrating the explainability of our TPSNet method on multiple cross-domain tasks and datasets.

B.10. Computational Cost Analysis

We evaluate the additional computational overhead introduced by the phase prior extraction and text prior cross-modal attention mechanisms. To quantify this overhead, we conduct detailed ablation studies on four key metrics: parameter count, FLOPs, inference time, and memory usage. The results are summarized in Table 13.

As shown in our ablation study, the phase branch—though built with a lightweight convolutional encoder—introduces an additional 1.03M parameters and 1.96G FLOPs, primarily due to processing an extra phase image in parallel with the RGB input. Despite this, the increase in inference time is negligible (+0.006ms), and the memory overhead remains modest (+37MB), demonstrating the efficiency of the design. In contrast, text prior cross-modal attention mechanism adds only 0.02M parameters and 0.01G FLOPs, as it reuses existing features and performs lightweight token-level cross-modal interactions. However, it results in a more noticeable increase in inference time (+0.064ms), attributed to the sequential attention operations—including Q-K-V projections, matrix multiplications, and softmax computations—which are computationally intensive and less parallelizable than convolutional operations. Overall, the full model introduces just 2.08M additional parameters (1.8%) and 2.0G more FLOPs (4.4%) compared to the baseline. Inference time rises from 0.031ms to 0.104ms, and memory usage increases moderately from 1067MB to 1119MB. These results demonstrate that our proposed method maintains high efficiency and is well-suited for scalable deployment on larger datasets or in real-time applications.

C. Limitations

Although TPSNet achieves significant improvements on the UCDIR task, it still has certain limitations. In DPGM, the generation of domain prompts relies on a manually designed prompt in the form of "An image of a $[X]^1[X]^2 \dots [X]^M$." While this provides a simple and interpretable way to encode image semantics, it may limit the expressiveness and flexibility of the prompts, especially in complex or diverse domains. In future work, we plan to explore more dynamic or learnable prompt generation strategies to enhance generalization across broader domain distributions.

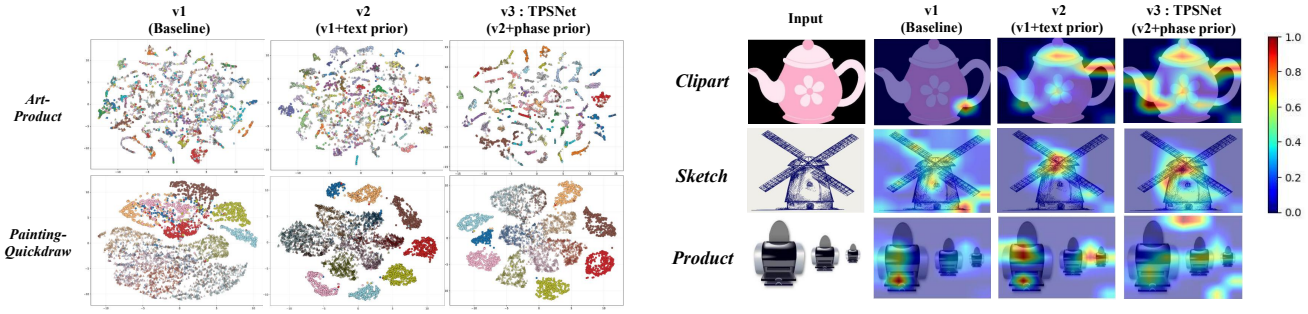

Figure 10. Additional t-SNE and Grad-CAM visualizations of last-layer features for the baseline model (v1), baseline with text prior (v2), and TPSNet (v3) across two datasets.

Table 13. The additional computational overhead introduced by the phase spectrum extraction and cross-modal attention mechanisms.

DPG	TPFE	PPFE	Params_all	FLOPs	Inference Time	Memory Usage
✗	✗	✗	114.518M	45.081G	0.031ms	1067.23MB
✗	✗	✓	115.549M	47.042G	0.037ms	1104.28MB
✓	✓	✗	115.569M	45.088G	0.095ms	1085.37MB
✓	✓	✓	116.600M	47.049G	0.104ms	1119.16MB