Vision-Language Model Guided Source-Free Domain Adaptation via Optimal Transport

Supplementary Material

1. Notation Table

For clarity and consistency, we list the main symbols used in the VSFOT framework in Table 1.

Symbol	Description
$\mathcal{D}_s$	Source domain
$\mathcal{D}_t$	Target domain
N_s, N_t	Number of source and target samples
K	Number of classes
B	Mini-batch of target samples
$\mathcal{G} = (f, c)$	Source model
$\mathcal{V}$	VLM
h	Projection head
w_j	Class prototype for class j
z_i^m	Feature embedding of target sample i
$\tilde{z}_i^m$	Projection-head embedding of target sample i
q_i^m	Model-predicted class probability of sample i
$\tilde{q}_i^m$	Normalized class score of sample i
z_{img}	VLM image embedding
z_{text}	VLM text embedding
S^v	Image-text similarity matrix in VLM
S^m	Sparse matrix of model-predicted class scores
q_i^v	VLM-derived class probability of sample i
C^m	Cost matrix between target and prototypes
C^v	VLM guided transport cost matrix
Γ	Transport plan
Γ^*	Optimal transport plan
μ_t, μ_p	Target and prototype marginals in OT
$d(\cdot, \cdot)$	Cosine distance
$L(\cdot, \cdot)$	Cross-entropy function
$H(\cdot)$	Entropy function
α	Normalization coefficient
$\mathcal{L}_{\text{Align}}$	Alignment loss
$\mathcal{L}_{\text{Con}}$	Contrastive regularization loss
$\mathcal{L}_{\text{IM}}$	Information maximization loss
$\mathcal{L}_{\text{VLM}}$	Model-to-VLM distillation loss
$\mathcal{L}_1$	VGMA objective
$\mathcal{L}_2$	MGVA objective

Table 1. Summary of the key notations used in the VSFOT framework.

2. Proofs of Theoretical Propositions

This section presents the formal statement and proof of Proposition 1. It establishes a target-risk bound for the VLM-guided alignment objective in Eq. 2 and clarifies how minimizing this objective controls the target-domain risk.

Proposition 1. Under mild regularity conditions on the task loss and the feature mapping, there exist constants $\lambda > 0$ and $\Delta \geq 0$ such that, for any hypothesis h,

$$\mathcal{R}_t(h) \leq \varepsilon_s + \lambda W_1^{C^m}(P_t, P_p) + \Delta, \tag{15}$$

where ε_s denotes the residual source risk of h, Δ accounts for the finite-prototype approximation error, and $W_1^{C^m}$ is the Wasserstein-1 distance induced by the joint cost C^m between the target distribution P_t and the prototype distribution P_p. Consequently, minimizing the alignment loss $\mathcal{L}_{\text{Align}}$ in Eq. 2 tightens this upper bound on the expected target risk.

Proof. Let P_t denote the joint distribution of target-domain samples and labels, and P_p denote the joint distribution induced by class prototypes. Consider a hypothesis $h : \mathcal{X} \to \mathcal{Y}$ and a task loss $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$. The expected target risk can be written as

$$\mathcal{R}_t(h) = \mathbb{E}_{(x^t, y^t) \sim P_t}\left[\ell\left(h\left(x^t\right), y^t\right)\right]. \tag{16}$$

Introduce a coupling $\gamma \in \Pi(P_t, P_p)$ between target samples and prototypes, where $\Pi(P_t, P_p)$ denotes the set of joint distributions on $((x^t, y^t), (w, y))$ with marginals P_t and P_p, respectively. Using this coupling, the target risk can be equivalently expressed as

$$\mathcal{R}_t(h) = \mathbb{E}_{((x^t, y^t), (w, y)) \sim \gamma}\left[\ell\left(h\left(x^t\right), y^t\right)\right]. \tag{17}$$

We now add and subtract the prototype loss inside the expectation:

$$\begin{aligned}
\mathcal{R}_t(h) &= \mathbb{E}_\gamma\left[\ell\left(h\left(x^t\right), y^t\right)\right] \\
&= \mathbb{E}_\gamma\left[\ell\left(h\left(x^t\right), y^t\right) - \ell\left(h\left(w\right), y\right) + \ell\left(h\left(w\right), y\right)\right] \\
&\leq \mathbb{E}_\gamma\left[\ell\left(h\left(w\right), y\right)\right] \\
&\qquad + \mathbb{E}_\gamma\left|\ell\left(h\left(x^t\right), y^t\right) - \ell\left(h\left(w\right), y\right)\right|.
\end{aligned} \tag{18}$$

Under mild regularity assumptions on the feature mapping f, the classifier h, and the task loss ℓ, we assume that the composite loss $\ell(h(\cdot), \cdot)$ is λ-Lipschitz with respect to the joint cost C^m. That is, there exists a constant $\lambda > 0$ such that, for any (x^t, y^t) and (w, y),

$$\left|\ell\left(h\left(x^t\right), y^t\right) - \ell\left(h\left(w\right), y\right)\right| \leq \lambda C^m\left(x^t, w\right). \tag{19}$$

Taking expectation with respect to γ yields

$$\mathbb{E}_\gamma\left|\ell\left(h\left(x^t\right), y^t\right) - \ell\left(h\left(w\right), y\right)\right| \leq \lambda \mathbb{E}_\gamma\left[C^m\left(x^t, w\right)\right]. \tag{20}$$

Moreover, since P_p is induced by class prototypes that approximate the source-domain structure, we can bound the prototype risk by the residual source risk plus an approximation error:

$$\mathbb{E}_{(w,y)\sim P_p}\left[\ell\left(h\left(w\right),y\right)\right] \leq \varepsilon_s + \Delta, \tag{21}$$

where ε_s is the residual risk of the source model and Δ accounts for the finite-prototype approximation. Combining the above inequalities, we obtain

$$\begin{aligned}\mathcal{R}_t(h) &\leq \mathbb{E}_{(w,y)\sim P_p}\left[\ell\left(h\left(w\right),y\right)\right] + \lambda\,\mathbb{E}_{\gamma}\left[C^m\left(x^t,w\right)\right] \\ &\leq \varepsilon_s + \lambda\,\mathbb{E}_{\gamma}\left[C^m\left(x^t,w\right)\right] + \Delta.\end{aligned} \tag{22}$$

Here C^m is the joint cost that combines the feature distance and the semantic divergence as defined in Eq. 1. Minimizing the right-hand side of Eq. 22 over all feasible couplings $\gamma \in \Pi(P_t, P_p)$ yields the optimal bound

$$\mathcal{R}_t(h) \leq \varepsilon_s + \lambda\, W_1^{C^m}\left(P_t, P_p\right) + \Delta, \tag{23}$$

where

$$W_1^{C^m}\left(P_t, P_p\right) := \inf_{\gamma\in\Pi(P_t,P_p)} \mathbb{E}_{\gamma}\left[C^m\left(x^t,w\right)\right] \tag{24}$$

is the Wasserstein-1 distance induced by the cost C^m between P_t and P_p.

In practice, we only have access to finite samples from P_t and P_p, and their empirical counterparts are denoted by μ_t and μ_p as defined in Eq. 6. The transport plan Γ^* used in our method is obtained by solving the VLM-guided OT problem in Eq. 5 with cost C^v. Although Γ^* is optimized with respect to C^v rather than C^m, it still defines a feasible coupling in $\Pi(\mu_t, \mu_p)$ and thus can be used as an empirical approximation of the optimal coupling appearing in the definition of $W_1^{C^m}(P_t, P_p)$. The alignment loss $\mathcal{L}_{\text{Align}}$ in Eq. 2 can be interpreted as the empirical OT cost induced by C^m evaluated on this VLM-regularized plan. Consequently, minimizing $\mathcal{L}_{\text{Align}}$ reduces the empirical counterpart of $W_1^{C^m}(P_t, P_p)$ under a semantically informed coupling, thereby tightening the target-risk bound in Eq. 15 in a prior-guided manner. □

Discussion. Intuitively, the VLM-guided OT aligns target features with semantically consistent prototypes, reducing both geometric and semantic discrepancies between domains. This tightens the Wasserstein distance $W_1^{C^m}(P_t, P_p)$ and thus lowers the upper bound on the target-domain risk.

3. Pseudocode of VSFOT

Based on Eqs. 13 and 14, we design an iterative training procedure. The overall process is summarized in Algorithm 1.

Algorithm 1 Training Procedure of VSFOT

Require: Pretrained source model $\mathcal{G}$; VLM $\mathcal{V}$; target data $\mathcal{D}_t$; prompt template v; number of iterations T.
Ensure: Adapted target model $\mathcal{G}_t$.

```
Initialize the V adapter.
for t = 1 to T do
    Sample a mini-batch B from D_t.
    // Stage: VGMA
    Freeze V; train G.
    Perform a forward pass of B through G to extract features and predictions.
    Perform a forward pass of B through V with the prompt v.
    Compute the cost matrix C^m based on Eq. 1.
    Solve for the OT plan Γ* as defined in Eq. 5.
    Compute the losses (Eqs. 2, 8, and 9); update G using the total loss in Eq. 13.
    // Stage: MGVA
    Freeze G; train the adapter of V.
    Obtain model predictions from G using Eq. 11.
    Obtain predictions from V via Eq. 4.
    Optimize the adapter using the loss in Eq. 14.
end for
Set G_t ← G.
return G_t
```

4. Implementation Details

Source Model Training We follow the same protocol as SHOT [8]. ResNet50 [4] is used as the backbone for Office-31, Office-Home, and DomainNet-126, while ResNet101 is employed for VisDA. The original fully connected layer is replaced with a 256-dimensional bottleneck layer, followed by a linear classifier whose output dimension matches the number of classes in each dataset. A BatchNorm layer [6] is applied after the bottleneck, and WeightNorm [11] is used on the classifier weights. All models are initialized with ImageNet-1K [1] pretrained weights. The source data are randomly split into a training set containing 90% of the samples and a validation set with the remaining 10%. The backbone network is optimized with an initial learning rate of $1e{-}3$, while the bottleneck layer and classifier use $1e{-}2$; both follow a polynomial decay schedule [3]. We use SGD with momentum 0.9 and weight decay $5e{-}4$ for all trainable parameters, and input images are resized to 256×256, randomly cropped to 224×224, horizontally flipped with probability 0.5, and normalized with ImageNet statistics. Training proceeds for 20 epochs, and the model checkpoint yielding the highest validation accuracy is retained for SFDA.

VLM Setting By default, CLIP ViT-B/32 [10] is adopted as the VLM, and lightweight adapters are inserted into its last three transformer layers. Each adapter layer consists

Method	SF	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg
Source	-	45.86	68.29	77.68	52.34	62.59	64.65	55.40	32.79	72.87	65.46	44.34	81.53	59.92
SHOT	✓	57.56	78.17	80.09	66.64	77.56	78.95	68.30	55.01	80.42	77.37	59.91	86.00	72.16
NRC	✓	58.30	80.52	83.50	66.86	79.86	78.46	65.05	56.06	81.40	71.46	59.29	85.46	72.18
AaD	✓	58.29	78.49	83.26	68.31	79.57	79.50	67.23	57.00	79.92	70.39	61.50	85.62	72.42
AdaCon	✓	45.94	75.89	74.83	61.85	73.31	72.65	60.46	45.39	80.61	73.12	54.02	89.30	67.28
CoWA	✓	57.15	78.54	80.51	69.57	79.80	79.61	67.38	57.70	82.76	74.04	59.49	85.60	72.68
PLUE	✓	48.16	74.49	77.08	62.55	73.35	74.76	62.07	48.82	91.00	84.28	76.19	83.44	71.35
TPDS	✓	59.28	78.91	81.09	70.14	79.28	81.18	69.55	56.25	84.35	77.60	59.09	84.33	73.42
DAPL	×	53.90	84.75	86.24	74.44	83.56	85.39	74.66	54.53	81.07	74.24	53.17	82.76	74.06
PADCLIP	×	57.81	83.52	84.00	77.13	85.18	85.06	76.45	59.78	86.36	81.04	59.67	90.03	77.17
ADCLIP	×	56.57	85.13	87.00	77.11	86.28	86.55	76.76	56.34	81.43	78.27	52.26	86.81	75.88
PDA	×	56.83	83.97	87.10	74.81	85.75	85.46	74.49	55.63	86.25	71.88	55.19	87.67	75.42
DAMP	×	60.49	88.31	86.13	76.36	88.49	87.34	76.49	59.75	87.87	76.91	61.63	89.10	78.24
DIFO	✓	70.43	90.42	88.56	83.05	91.15	88.76	80.34	69.92	88.85	81.65	70.77	90.86	82.90
ProDe	✓	74.06	91.11	91.32	81.50	**91.77**	91.10	82.90	72.93	89.64	**84.08**	70.99	89.05	84.20
VSFOT	✓	**75.00**	**91.72**	**91.42**	**83.00**	91.76	**91.40**	**83.49**	**75.01**	**91.36**	83.69	**74.29**	**91.93**	**85.34**

Table 2. Top-1 accuracy (%) on Office-Home. Bold indicates the best performance.

Method	SF	A→D	A→W	D→A	D→W	W→A	W→D	Avg
Source	-	80.50	71.70	61.90	92.10	61.60	91.40	76.53
SHOT	✓	94.20	91.35	73.44	98.95	70.76	100.00	88.12
NRC	✓	94.24	93.44	74.48	98.37	77.82	100.00	89.72
AaD	✓	92.60	92.10	75.54	98.70	72.70	99.80	88.57
AdaCon	✓	97.30	83.23	74.66	91.17	76.36	73.20	82.65
CoWA	✓	91.28	94.95	75.54	95.51	82.30	100.00	89.93
PLUE	✓	91.00	88.27	72.12	97.97	76.88	97.50	87.29
TPDS	✓	93.60	93.87	76.31	96.53	75.29	99.60	89.20
DIFO	✓	96.40	95.63	83.61	97.07	82.39	99.60	92.45
ProDe	✓	**96.80**	96.15	**83.62**	**97.25**	81.98	99.60	92.57
VSFOT	✓	96.59	**95.60**	83.46	96.98	**83.39**	**99.93**	**92.66**

Table 3. Top-1 accuracy (%) on Office-31. Bold indicates the best performance.

of two linear layers with an intermediate ReLU activation, a hidden dimension of 64, and a residual connection. For prompt templates, we use the standard format "a photo of a {CLASS}", where {CLASS} is replaced with the class name. Target images input to CLIP follow the official preprocessing pipeline, and during adaptation all CLIP parameters are frozen except for the adapter layers.

OT Solver Setting We employ the POT [2] library as the OT solver in VSFOT. The optimization is performed using the Sinkhorn algorithm with the entropy regularization term coefficient set to 0.2 across all experiments. OT is computed at the mini-batch level, and at most $1e3$ Sinkhorn iterations are run with a stopping tolerance of $1e-9$.

SFDA Process For model optimization, we use stochastic gradient descent with an initial learning rate of $1e-3$, and apply cosine annealing [9] for learning rate scheduling, where $T_{\max}$ is set to the total number of training iterations. For the VLM, only the adapter parameters are updated while the rest of the network remains frozen. The adapter is optimized using Adam [7] with a fixed learning rate of $5e-5$. During SFDA, both the source backbone and classifier are updated together with the VLM adapters, while all other CLIP parameters remain frozen. SGD uses momentum 0.9 and weight decay $5e-4$, and Adam uses default β values with weight decay $1e-4$. Unless otherwise specified, the batch size is set to 64 for all experiments. Training is conducted for 20 epochs on Office-31 and Office-Home, and for 40 epochs on VisDA and DomainNet-126. All experiments are implemented in PyTorch and run on NVIDIA RTX 3090 GPUs. Each experiment is repeated three times with different random seeds, and the average accuracy is reported.

5. Experiment Details

Tables 2, 3, and 4 provide the full Top-1 accuracy results on Office-Home, VisDA, and Office-31, complementing the summary reported in the main paper. The column "SF" indicates whether each method strictly follows the source-free setting or relies on access to source data.

Overall, the detailed transfer-wise results are consistent with the conclusions in the main paper. On Office-Home and Office-31, VSFOT either matches or surpasses all competing methods on most transfer directions, and achieves the best average accuracy across methods, confirming its strong and stable improvements over both conventional SFDA approaches and recent VLM-based variants. On VisDA, VSFOT remains highly competitive and ranks second in terms of average accuracy; the main performance gap stems from a small subset of categories (e.g., truck) where misclassified samples are not explicitly corrected during adaptation. These extended results further support the claim that VLM-guided OT yields robust cross-domain generalization in the

Method	SF	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Avg
Source	-	88.39	18.20	59.80	79.60	71.54	10.84	80.36	20.50	63.20	29.70	84.68	7.64	51.20
SHOT	✓	97.39	88.12	81.63	62.15	95.85	95.21	85.98	84.52	89.12	92.30	85.24	60.68	84.85
NRC	✓	95.12	86.00	82.40	61.14	96.15	96.87	86.09	87.70	94.58	94.05	94.40	59.05	86.13
AaD	✓	96.45	89.85	87.69	**84.24**	95.37	96.11	87.83	84.96	95.54	93.04	94.23	64.74	89.17
AdaCon	✓	97.20	81.57	83.98	76.50	96.27	93.10	94.69	84.53	93.11	92.65	93.08	40.69	85.61
CoWA	✓	97.89	89.70	83.90	74.15	96.98	97.43	89.32	84.85	94.20	93.14	88.92	54.49	87.08
PLUE	✓	97.21	92.80	90.15	80.40	**98.15**	95.53	94.40	89.14	92.00	92.20	91.66	59.40	89.42
TPDS	✓	98.59	91.30	89.92	81.33	97.31	94.23	91.16	80.34	91.46	86.06	90.85	42.37	86.24
DAPL	×	98.59	89.24	89.80	78.97	96.82	94.78	93.61	82.98	87.60	92.57	92.59	62.92	88.37
PADCLIP	×	97.25	89.83	89.76	82.82	97.28	93.81	91.33	84.08	**95.69**	93.98	91.50	63.09	89.21
ADCLIP	×	98.52	86.60	91.35	76.82	98.12	93.41	92.75	83.66	89.49	92.75	92.14	64.28	88.32
PDA	×	98.01	82.21	92.29	75.10	97.23	85.29	93.70	76.37	86.15	87.94	89.30	59.80	85.28
DAMP	×	97.98	91.90	91.28	79.51	97.58	95.10	92.33	84.55	91.34	92.18	92.20	64.50	89.20
DIFO	✓	97.25	88.99	90.75	83.26	97.92	97.23	92.16	83.47	95.16	92.58	91.63	61.58	89.33
ProDe	✓	98.30	**93.10**	92.60	79.80	95.43	**97.80**	90.28	84.44	93.75	**96.82**	91.17	**78.60**	**91.04**
VSFOT	✓	**99.07**	92.79	**92.66**	81.78	98.06	96.39	**94.79**	**84.62**	93.97	95.27	**92.81**	64.58	90.57

Table 4. Top-1 accuracy (%) on VisDA. Bold indicates the best performance.

source-free setting.

6. Visualization of Transport and Feature Alignment

6.1. Transport Plan Visualization

To provide an intuitive understanding of how VSFOT aligns target samples to class prototypes, we visualize the transport plans at the beginning of adaptation, before any target-side parameter updates. We select three representative tasks: Office-Home Pr→Ar, DomainNet-126 P→C, and VisDA synthetic→real. For each task, we plot the transport matrix between target samples and class prototypes under two cost definitions.

Figure 1 compares OT computed solely from the model-based cost (OT without prior) with the VLM-guided transport used in VSFOT (OT with prior). On Office-Home Pr→Ar and DomainNet-126 P→C, the model-only OT already shows a coarse diagonal pattern but still spreads mass over incorrect classes, whereas the VLM-guided cost produces a much sharper diagonal with fewer spurious couplings. On VisDA synthetic→real, the model-only transport is highly scattered due to the severe domain gap, while the VLM-guided transport reveals a clearer block-structured alignment even at initialization.

These visualizations indicate that the VLM-informed cost acts as a semantic regularizer on the transport plan from the very beginning of adaptation, reducing noisy correspondences under large domain shifts.

6.2. Feature Space Visualization

We further analyze the effect of VLM-guided OT on the target feature geometry by comparing the 2D embeddings of the target domain before and after adaptation. For three

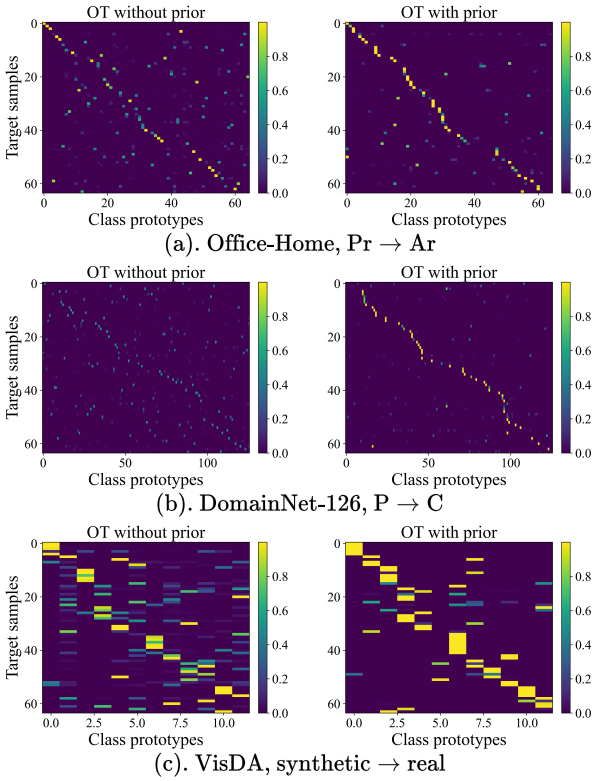

Figure 1. Transport plans at the beginning of adaptation for (a). Office-Home Pr→Ar, (b). DomainNet-126 P→C, and (c). VisDA synthetic→real. Left: OT without prior based only on model predictions. Right: VLM-guided OT used in VSFOT. Each matrix shows couplings between target samples and class prototypes, with brighter colors indicating larger coupling mass.

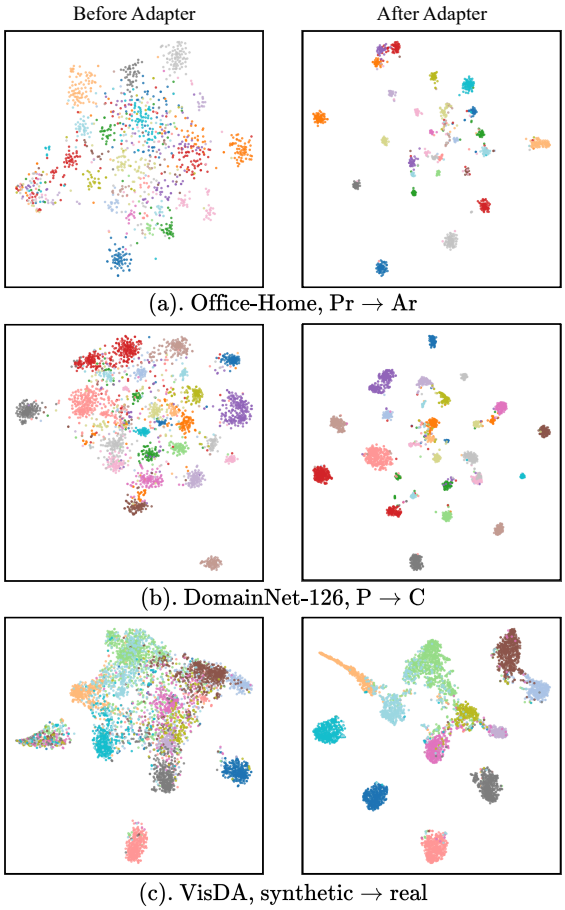

Figure 2. Feature space visualizations on target domains for (a) Office-Home Pr→Ar, (b) DomainNet-126 P→C, and (c) VisDA synthetic→real. Left: source model before adaptation. Right: VSFOT after adaptation. Points denote target samples colored by their ground-truth classes.

representative adaptation tasks, we project target features into 2D using t-SNE and color points by their ground-truth labels. For Office-Home and DomainNet-126, we visualize the first 30 classes, while for VisDA we randomly sample 5000 target images to keep the plots clear.

As shown in Figure 2, before adaptation the target features exhibit substantial class overlap and fragmented clusters, especially on the more challenging DomainNet-126 and VisDA benchmarks. After applying VSFOT, class-specific clusters become much more compact and clearly separated, and many diffuse regions collapse into well-formed groups. These observations indicate that VSFOT substantially improves the discriminability and class-wise organization of the target feature space.

Prompt Template	Office-Home
"a photo of a {CLASS}"	**85.34**
"a photo of a {CLASS} in the scene"	84.86
"a photo of a {CLASS} in the {Domain} domain"	85.14
"a {CLASS} object"	85.74
"A picture containing a {CLASS}"	85.53
"This is a photo of a {CLASS}"	85.36

Table 5. Prompt robustness analysis on Office-Home. Average Top-1 accuracy (%) of VSFOT under different prompt templates. The default prompt is highlighted in bold.

VLM Backbone	Params	Office-Home
CLIP RN50	102.0M	72.25
CLIP RN101	119.7M	74.09
CLIP ViT-B/32	**151.2M**	**85.34**
CLIP ViT-B/16	149.6M	87.65
CLIP ViT-L/14	427.6M	91.15
SigLIP ViT-B/16	203.2M	92.20

Table 6. Performance of VSFOT with different VLM backbones on Office-Home. The default backbone is highlighted in bold.

7. Additional Analysis Experiments

7.1. Prompt Robustness Analysis

VSFOT relies on a VLM to provide semantic priors, which are obtained by pairing target images with text prompts. In the main experiments, we adopt a standard prompt template of the form "a photo of a {CLASS}". To assess sensitivity to prompt design, we further evaluate several alternative prompt formulations on Office-Home.

Table 5 reports the average Top-1 accuracy on Office-Home under six different prompt templates. Overall, all prompts yield very similar performance, with accuracies ranging from 84.86% to 85.74%. The default template "a photo of a {CLASS}" already performs competitively (85.34%), while the object-centric description "a {CLASS} object" achieves the highest accuracy (85.74%). These results indicate that VSFOT is relatively insensitive to moderate changes in prompt wording: simple variants can provide slight gains, but the overall adaptation performance remains stable without careful prompt engineering.

7.2. Analysis with Different VLM Backbones

While CLIP ViT-B/32 is adopted as the default VLM in our main experiments, VSFOT is compatible with a broad range of vision–language backbones. To assess the effect of VLM capacity, we replace the default backbone with several publicly available CLIP variants and SigLIP [12], and rerun SFDA on Office-Home under the same hyperparameter settings. Table 6 reports the number of parameters and the target-domain accuracy for each backbone. When moving

Method	Office-Home
Adapter Layer	**0.8534**
LoRA	0.8517
Prompt tuning	0.8445

Table 7. Comparison of different fine-tuning strategies on Office-Home. The default method is highlighted in bold.

Method	GPU Memory (MB)	Training Time (s)
VSFOT	**9497.4**	**0.5101**
w/o regularization	4167.4	0.3428
SHOT	6000.8	0.2406
ProDe	7801.1	0.5282

Table 8. Computational cost comparison on Office-31 (A→W). The results of VSFOT are highlighted in bold.

from ResNet-based CLIP models to transformer-based variants, the performance increases steadily from 72.25% with RN50 to 91.15% with ViT-L/14, indicating that stronger VLMs provide more informative semantic priors for VSFOT. Furthermore, replacing CLIP with the SigLIP backbone further improves the performance to 92.20%, demonstrating that VSFOT can effectively leverage stronger vision–language representations beyond CLIP. We use CLIP ViT-B/32 as the default backbone, since it achieves 85.34% accuracy with 151.2M parameters, offering a favorable balance between performance and model size while still leaving room for further gains with larger VLMs.

7.3. Analysis with Different Fine-Tuning Strategies

To demonstrate the generality of VSFOT with respect to parameter adaptation strategies, we further evaluate several parameter-efficient fine-tuning methods. Specifically, we compare the adapter layer used in VSFOT with two representative approaches, LoRA [5] and prompt tuning [13]. LoRA adapts the model by introducing low-rank decomposition into the weight matrices, while prompt tuning modifies the text encoder by learning task-specific prompts. The Table 7 show that VSFOT achieves strong performance even when combined with these advanced parameter-efficient tuning strategies. This observation indicates that the effectiveness of VSFOT does not rely on a specific fine-tuning mechanism, demonstrating its robustness and general applicability across different adaptation strategies.

7.4. Structural Complexity

Table 8 reports the computational cost of VSFOT, including GPU memory usage during training and the time consumed per forward pass, measured on the Office-31 A→W task. Compared with existing methods, VSFOT requires more GPU memory and computational time than lightweight self-training approaches such as SHOT, but achieves substantially better performance. To further investigate the role of different components, we analyze a simplified variant of VSFOT by removing the regularization term. This variant incurs only a marginal performance drop, while yielding substantial savings in both GPU memory and training time.

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 2

[2] Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, et al. Pot: Python optimal transport. *JMLR*, 22 (78):1–8, 2021. 3

[3] Rong Ge, Sham M Kakade, Rahul Kidambi, and Praneeth Netrapalli. The step decay schedule: A near optimal, geometrically decaying learning rate procedure for least squares. In *NeurIPS*, 2019. 2

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 2

[5] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022. 6

[6] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015. 2

[7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 3

[8] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *ICML*, pages 6028–6039, 2020. 2

[9] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *ICLR*, 2017. 3

[10] 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 *ICML*, pages 8748–8763, 2021. 2

[11] Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *NeurIPS*, 2016. 2

[12] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, pages 11941–11952, 2023. 5

[13] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *IJCV*, 130:2337–2348, 2022. 6