Noise-Aware Few-Shot Learning through Bi-directional Multi-View Prompt Alignment

Supplementary Material

6. Method Details

Training process. We detail the training process of NA-MVP in Algorithm 1, which includes a supervised phase and a denoising phase.

Optimal Transport. OT is a powerful framework for mapping one probability distribution to another while minimizing the associated transportation cost. Given two distributions $\boldsymbol{\mu} \in \mathbb{R}_+^m$ and $\boldsymbol{\nu} \in \mathbb{R}_+^n$, and a cost matrix $\boldsymbol{C} \in \mathbb{R}^{m \times n}$, the OT problem aims to find the optimal transport plan $\boldsymbol{T}$ that minimizes the following objective:

$$d_{\text{OT}}(\mu, \nu) = \min_{\boldsymbol{T} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} \langle \boldsymbol{C}, \boldsymbol{T} \rangle \tag{22}$$

$$\Pi(\mu, \nu) = \left\{ \mathbf{T} \in \mathbb{R}_+^{m \times n} \;\middle|\; \mathbf{T}\mathbb{1}_n = \mu,\ \mathbf{T}^\top \mathbb{1}_m = \nu \right\} \tag{23}$$

where $\langle \cdot, \cdot \rangle$ represents the Frobenius dot-product, and $\mathbb{1}_m$, $\mathbb{1}_n$ denote the vectors of ones of length m and n, respectively. Since solving OT exactly is computationally expensive, the entropy-regularized version is often used:

$$d_{\text{OT}}(\mu, \nu) = \min_{\boldsymbol{T} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} \langle \boldsymbol{C}, \boldsymbol{T} \rangle + \epsilon \langle \boldsymbol{T}, \log \boldsymbol{T} \rangle \tag{24}$$

where $\epsilon > 0$ controls the strength of regularization. The added entropy term $\langle \boldsymbol{T}, \log \boldsymbol{T} \rangle$ promotes smoother transport plans and allows for efficient optimization via the Sinkhorn algorithm [15]. The optimization process can be completed in a few iterations, with the solution T^* being computed as:

$$T^* = \text{diag}(\mu^{(t)}) \exp(-\boldsymbol{C}/\epsilon) \text{diag}(\nu^{(t)}), \tag{25}$$

where t denotes the iteration number and in each iteration, the marginal distributions $\mu^{(t)} = \mu / \left(\exp(-\boldsymbol{C}/\epsilon)\nu^{(t-1)}\right)$ and $\nu^{(t)} = \nu / \left(\exp(-\boldsymbol{C}/\epsilon)\mu^{(t)}\right)$.

Fast Implementation of Dykstra's Algorithm. To efficiently solve the entropically regularized UOT problem defined in Eq. 3, we adopt a fast matrix-scaling variant of Dykstra's algorithm. The full procedure is detailed in Algorithm 2.

Generalized Cross-Entropy Loss. To support the training objectives described in Eq. (26)–(27), we provide a detailed explanation of the Generalized Cross-Entropy (GCE) loss [69], denoted as $\mathcal{L}_{\text{gce}}$. This loss serves as the primary supervision signal for training, especially in the presence of label noise.

The GCE loss is a noise-robust surrogate to the standard cross-entropy (CE) and mean absolute error (MAE) losses.

Algorithm 1 The training process of NA-MVP

Input: Noisy dataset $\mathcal{D}_{\text{noisy}}$, pretrained CLIP model f, text encoder $\psi(\cdot)$, number of prompts N, entropy parameter ϵ, total training epochs T, number of supervised epochs T_{sup}

Output: Optimized prompt parameter set $\Omega = \{\omega_m^c, \omega_m^n\}_{m=1}^N$

for $t = 1, 2, \ldots, T$ **do**
  for each mini-batch $\mathcal{B}_t$ **do**
    if $t > T_{\text{sup}}$ **then**
      Identify noisy samples using threshold $\phi(p^n, p^c)$ and construct $\mathcal{D}_{\text{denoised}}$
    end if
    Extract local feature map F_i using f
    Generate prompt feature sets $G_k^c \in \mathbb{R}^{N \times d}$ and $G_k^n \in \mathbb{R}^{N \times d}$ for each class k
    Compute cost matrices $C_k^c = 1 - F_i G_k^{c\top}$, $C_k^n = 1 - F_i G_k^{n\top}$ using cosine similarity
    Solve UOT using Dykstra's algorithm: $T_k^{c*} = \text{diag}(\mu^{(t)}) \exp(-C_k^c/\epsilon) \text{diag}(\nu^{(t)})$, $T_k^{n*} = \text{diag}(\mu^{(t)}) \exp(-C_k^n/\epsilon) \text{diag}(\nu^{(t)})$
    Compute UOT-based distances: $d_{\text{UOT}}^c(k) = \langle C_k^c, T_k^{c*} \rangle$, $d_{\text{UOT}}^n(k) = \langle C_k^n, T_k^{n*} \rangle$
    Compute final prediction probabilities: $p(y = k \mid x_i) = (1 - p_{i,k}^n) \cdot p_{i,k}^c$
    if $t > T_{\text{sup}}$ **then**
      Compute $\mathcal{L} = \mathcal{L}_{\text{gce}}$ using $\mathcal{D}_{\text{denoised}}$
    else
      Compute $\mathcal{L} = \mathcal{L}_{\text{gce}} + \lambda_i \cdot \mathcal{L}_{\text{itbp}}$ using $\mathcal{D}_{\text{noisy}}$
    end if
    Update prompt parameters Ω with loss $\mathcal{L}$
  end for
end for
return Ω

Given a training sample (x, y) where $y \in \{1, 2, \ldots, C\}$ is the ground-truth label and $p = f(x) \in \Delta^{C-1}$ is the softmax output over C classes, the GCE loss is defined as:

$$\mathcal{L}_{\text{gce}}(x, y) = \frac{1 - p_y^q}{q}, \qquad 0 < q \leq 1, \tag{26}$$

where p_y denotes the predicted probability for class y, and q is a tunable hyper-parameter that governs the degree of robustness. Following prior works, we fix $q = 0.5$ throughout all experiments, which offers a good trade-off between noise robustness and optimization stability.

Image-Text Bi-directional Prompt Loss. Drawing

Algorithm 2 Fast Implementation of Dykstra's Algorithm

Input: Cost matrix C, marginal vectors μ, ν, entropic regularization parameter ϵ

Initialize: $Q \leftarrow \exp(-C/\epsilon)$, $\nu^{(0)} \leftarrow \mathbb{1}_\nu$, $\Delta_\nu \leftarrow \infty$, $\epsilon \leftarrow 10^{-3}$

Compute:

$$Q_\mu \leftarrow \frac{Q}{\mathrm{diag}(\mu)\mathbb{1}_{|\mu|\times|\nu|}}, Q_\nu^\top \leftarrow \frac{Q^\top}{\mathrm{diag}(\nu)\mathbb{1}_{|\mu|\times|\nu|}}$$

for $n = 1, 2, \ldots$ **do**

$$\mu^{(n)} \leftarrow \min\left(\frac{\mathbb{1}_{|\mu|}}{Q_\mu \nu^{(n-1)}}, \mathbb{1}_{|\mu|}\right)$$

$$\nu^{(n)} \leftarrow \frac{\mathbb{1}_{|\nu|}}{Q_\nu^\top \mu^{(n)}}$$

$$\Delta_\nu \leftarrow \left\|\nu^{(n)} - \nu^{(n-1)}\right\|$$

if $\Delta_\nu < \epsilon$ **then**

break

end if

end for

inspiration from the bi-directional contrastive loss in CLIPN [52], we introduce an auxiliary loss termed ITBP within our framework. This loss ensures image features remain close to semantically consistent prompts while avoiding confusion with incorrect or unrelated noisy prompts.

Specifically, we distinguish between two types of matches: (1) **reversed match**, where the image matches its own noisy prompt that expresses opposite semantics, and (2) **unrelated match**, where the image is compared with other noisy prompts that are semantically irrelevant but not intentionally contradictory. To encode this relationship, we define a binary match indicator $m(x_i, t_j^{\mathrm{n}})$ between the i-th image and the j-th noisy prompt as:

$$m(x_i, t_j^{\mathrm{n}}) = m_{ij} = \begin{cases} 0, & i = j, \text{(reversed match)} \\ 1, & i \neq j, \text{(unrelated match)} \end{cases} \tag{27}$$

The matched probability between image x_i and the j-th noisy prompt t_j^{n} is computed as:

$$p_{ij}^{\mathrm{n}} = \frac{\exp(s_{i,j}^n/\tau)}{\exp(s_{i,j}^c/\tau) + \exp(s_{i,j}^n/\tau)} \tag{28}$$

The ITBP loss is formulated as:

$$\mathcal{L}_{\text{itbp}} = -\frac{1}{N}\sum_{i=1}^{N}(1 - m_{ii})\log(1 - p_{ii}^{\mathrm{n}}) - \frac{1}{N(N-1)}\sum_{i=1}^{N}\sum_{j\neq i} m_{ij}\log(p_{ij}^{\mathrm{n}}) \tag{29}$$

7. Experimental Details

7.1. Dataset Details

We selected six representative visual classification datasets as benchmarks. The detailed statistics of each dataset are shown in Table 5, including the original task, the number of classes, and the sizes of training and test samples.

7.2. Implementation details

All input images are resized to 224×224 and divided into 14×14 patches of dimension 768. For the Unbalanced OT problem in Eq. 6, we set the entropic regularization weight to $\epsilon = 0.1$ and the marginal relaxation parameter to $\theta = 0.9$. The loss balancing coefficient λ_{i} is set to 0.1. The maximum number of iterations in Algorithm 2 is set to 100, with early stopping applied when $\Delta_\nu < 0.01$. We use 16 shared context tokens appended to the class token, each of dimension 512. Prompts are randomly initialized and inserted at the "end" token position. Batch sizes are set to 32 for training and 100 for testing. The total number of training epochs is 50, with 20 for the supervised phase and 30 for semi-supervised refinement. Warm-up is set to 1 epoch for all datasets, except for the Flowers dataset, which adopts a 20-epoch warm-up. All experiments are conducted on a single NVIDIA GeForce RTX 4090 GPU.

7.3. Noisy label identification motivation

To validate the intuition behind our noisy label identification strategy, we conduct a case study on the Caltech101 dataset. Samples are grouped into clean and noisy subsets based on their ground-truth annotations. We visualize the epoch-wise average values of the clean prompt confidence p^c and the adaptive threshold ϕ for both groups.

As shown in Figure 5, clean samples exhibit consistently high values of p^c, while their corresponding ϕ remains close to zero. In contrast, noisy samples demonstrate the opposite pattern: ϕ is significantly larger than p^c, especially in the early training phase. These trends confirm that our bi-directional alignment framework provides a meaningful signal to distinguish between clean and noisy labels. Moreover, the observed dynamics further justify the design of our selective noisy label refinement strategy introduced in Section 3.2, which integrates soft thresholding with OT-based pseudo-label correction.

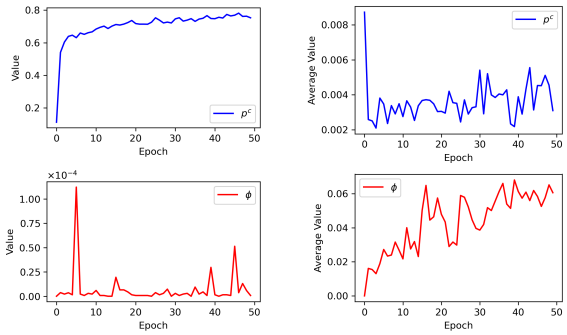
8. Additional Experimental results

8.1. Effect of Selective Refinement with $\phi_{i,k}$

To further analyze the effectiveness of our proposed selective label refinement guided by $\phi_{i,k}$, we compare the evolution of noisy label ratio over training epochs under different settings. As shown in Figure 6, we visualize the noisy rate curves during training on Caltech101 and OxfordPets un-

Table 5. The detailed statistics of datasets used in experiments.

Dataset	Task	Classes	Training Size	Testing Size
Caltech101 [16]	Object recognition	100	4,128	2,465
DTD [11]	Texture recognition	47	2,820	1,692
Flowers102 [40]	Fine-grained flowers recognition	102	4,093	2,463
OxfordPets [42]	Fine-grained pets recognition	37	2,944	3,669
UCF101 [43]	Video action recognition	101	7,639	3,783
Food101N [31]	Fine-grained food recognition	101	310,009	30,300



(a) Clean samples (b) Noisy samples

Figure 5. Behavioral differences of p^c and ϕ on clean vs. noisy samples in Caltech101.

der three synthetic noise levels: 25%, 50%, and 75%. We compare the baseline OT refinement strategy with our selective refinement method guided by $\phi_{i,k}$. Across all datasets and noise settings, we observe that the noisy rate decreases more significantly and remains consistently lower when using $\phi_{i,k}$-guided refinement. This indicates that our method effectively filters out potentially clean samples from aggressive relabeling, preventing overcorrection and improving the quality of pseudo-labels.

We further quantify the effectiveness of our selective refinement by evaluating the corresponding correction accuracy. For example, on Caltech101, our method reduces the noise ratio from 0.50 to 0.19, achieving a correct correction rate of 73.7%; under 0.75 noise, it decreases to 0.31 with a 71.4% correct rate. These findings further demonstrate that $\phi_{i,k}$ improves label correction reliability by enabling a more conservative and noise-aware refinement process.

8.2. Analysis of the Number of Multi-view Prompts

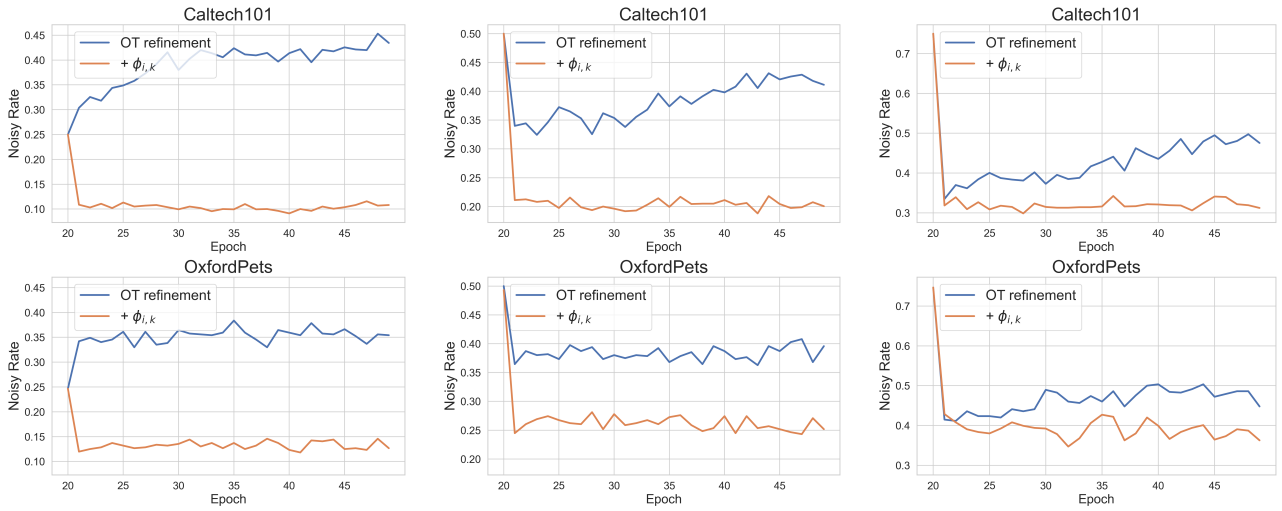
To investigate the effect of imbalanced numbers of clean and noisy prompts, we conduct experiments by fixing the number of clean prompts to 4 while varying the number of noisy prompts (N = 1, 2, 4, 8). The evaluation is performed under different noise rates on the DTD and OxfordPets datasets. The results clearly show that the best performance is achieved when the number of noisy prompts equals the number of clean prompts (i.e., N=4). This trend is particularly evident under higher noise rates, indicating that a balanced prompt configuration enhances model robustness to noise.

8.3. Comparison with other methods

Comparison with PLOT. PLOT [8] utilizes OT to align image and prompt features for few-shot classification and inspired us to identify noisy labels from multi-view. While PLOT has shown great promise, our method, NA-MVP, improves upon this by integrating multi-view prompt learning and a UOT-based denoising strategy, particularly effective in high-noise conditions. To validate NA-MVP, we compared it with PLOT under various noise levels. As shown in Table 6, NA-MVP consistently outperforms PLOT in all datasets and noise settings, with a particularly significant advantage under higher noise conditions. These results emphasize the robustness of our framework in handling noisy labels, supporting the effectiveness of our approach in noisy regimes.

Comparison with unsupervised methods. Although few-shot learning generally outperforms unsupervised methods, we recognize the importance of empirically evaluating whether learning from noisy few-shot labels can yield meaningful improvements. In this regard, we compared NA-MVP with two recent unsupervised methods: MetaPrompt [39] and LaFTer [38]. As shown in Table 7, NA-MVP outperforms both MetaPrompt and LaFTer across most datasets. Even at higher noise levels (50% and 75%), our method remains competitive or superior. These results demonstrate that learning from a small amount of noisy supervision, when appropriately modeled, can be more effective than training with no labeled data. This underscores the practical value of noisy few-shot learning in real-world low-resource scenarios.

Comparison with NBNN. We also compared NA-MVP with NBNN [4], a widely used set-to-set matching method with cosine similarity as a distance metric. Our experiments on the Caltech101 and OxfordPets datasets show that NA-MVP consistently outperforms NBNN across different noise levels, particularly under higher noise conditions. The



(a) Noise rate = 25% (b) Noise rate = 50% (c) Noise rate = 75%

Figure 6. Comparison of OT refinement and selective refinement with $\phi_{i,k}$ under different noise rates on Caltech101 and OxfordPets datasets.

Table 6. The accuracy comparison across datasets and noise levels. (%)

Dataset	Method	Noise rate: Sym						Noise rate: Asym	
		0.125	0.25	0.375	0.5	0.625	0.75	0.25	0.5
Caltech101	PLOT [8]	90.03	88.10	85.13	84.10	75.90	62.70	81.17	50.80
	NA-MVP	**92.07**	**92.10**	**91.60**	**91.30**	**90.07**	**89.37**	**91.47**	**89.53**
DTD	PLOT [8]	60.27	56.87	50.77	44.67	36.53	23.57	52.80	32.03
	NA-MVP	**63.73**	**63.13**	**61.63**	**58.50**	**52.93**	**48.63**	**62.33**	**52.10**
Flowers102	PLOT [8]	91.63	89.00	84.67	77.10	66.80	47.57	76.20	40.60
	NA-MVP	**94.20**	**93.30**	**92.00**	**90.47**	**85.07**	**76.47**	**91.37**	**78.43**
OxfordPets	PLOT [8]	84.57	79.43	74.70	64.60	52.10	41.50	73.87	44.47
	NA-MVP	**88.50**	**88.40**	**88.23**	**88.13**	**86.93**	**86.23**	**87.53**	**79.33**
UCF101	PLOT [8]	73.30	69.37	65.27	59.13	51.50	40.93	61.33	36.43
	NA-MVP	**75.33**	**74.03**	**72.30**	**70.93**	**68.43**	**63.93**	**73.40**	**65.40**

Table 7. Performance comparison with unsupervised medthods.

Method	OxfordPets	DTD	UCF101	Flowers102
MetaPrompt [39]	88.10	50.80	67.90	73.90
LaFTer [38]	82.70	46.10	68.20	71.00
NA-MVP (25% noise)	**88.40**	**63.10**	**74.00**	**93.30**
NA-MVP (50% noise)	**88.13**	**58.50**	**70.93**	**90.47**
NA-MVP (75% noise)	86.23	48.63	63.93	**76.47**

Table 8. Experimental results under imbalanced numbers of prompts on DTD and OxfordPets.

Datasets	DTD				OxfordPets			
Noise rate	0.25	0.5	0.75	Avg.	0.25	0.5	0.75	Avg.
N=1	62.36	57.33	47.24	55.64	88.27	87.23	85.26	86.92
N=2	62.86	57.17	47.40	55.81	88.20	87.59	85.50	87.10
N=4	**63.13**	**58.50**	**48.63**	**56.75**	88.40	**88.13**	**86.23**	**87.59**
N=8	62.76	57.07	47.37	55.73	**88.57**	87.63	85.73	87.31

results in Table 10 indicate that UOT-based matching in NA-MVP provides a more robust and adaptive alignment between prompts and image features, especially when class distributions are corrupted or ambiguous.

8.4. Parameter Study

Parameter Study of the Auxiliary Loss Weight λ_i. We study the effectiveness of the auxiliary loss weight λ_i,

Table 9. Impact of the loss balancing coefficient λ_i under different noise rates. (%)

Datasets	DTD			OxfordPets			UCF101		
Noise rate	0.25	0.5	0.75	0.25	0.5	0.75	0.25	0.5	0.75
λ_i=0.01	62.36	57.96	48.09	88.37	87.93	85.43	73.53	70.57	61.87
λ_i=0.05	62.03	58.10	48.39	**88.53**	**88.37**	85.90	73.60	**71.09**	62.90
λ_i=0.1	**63.13**	**58.50**	**48.63**	88.40	88.13	**86.23**	**74.03**	70.93	**63.93**
λ_i=0.5	62.96	57.33	47.19	88.33	87.50	85.33	73.63	69.13	60.74

Table 10. Performance comparison on Caltech101 and OxfordPets. (%)

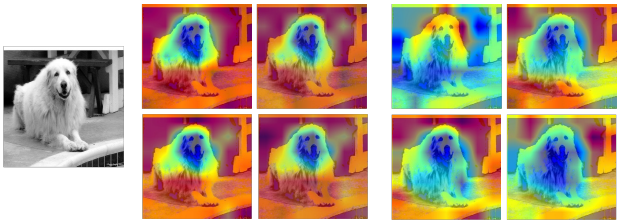
Dataset	Method	12.5%	25.0%	37.5%	50.0%	62.5%	75.0%
Caltech101	NBNN [4]	88.87	88.57	88.43	87.20	84.17	85.67
	NA-MVP	**92.07**	**92.10**	**91.60**	**91.30**	**90.07**	**89.37**
OxfordPets	NBNN [4]	87.47	86.00	85.53	83.07	82.50	80.67
	NA-MVP	**88.50**	**88.40**	**88.23**	**88.13**	**86.93**	**86.23**

Table 11. Performance under different noise rates and θ values. (%)

Noise rate/θ	0.5	0.6	0.7	0.8	0.9	1.0
25.00%	61.87	62.83	63.30	**63.80**	63.13	63.37
50.00%	57.57	58.73	58.07	58.67	**58.80**	58.30
75.00%	47.43	48.30	47.67	47.13	**48.63**	46.30

which controls the contribution of the ITBP loss during supervised training. As shown in Table 9, we evaluate $\lambda_i \in \{0.01, 0.05, 0.1, 0.5\}$ across DTD, OxfordPets, and UCF101 under different noise levels. In particular, $\lambda_i = 0.1$ consistently delivers the best or competitive results under different conditions. Therefore, we adopt $\lambda_i = 0.1$ as the default setting in all experiments.

Parameter Study of the parameter θ in UOT. To investigate the effect of the parameter θ in unbalanced OT, which regulates the mapping size of prompts on the feature map, we conducted additional experiments on the DTD dataset under varying noise rates and θ values ranging from 0.5 to 1.0. As shown in Table 11, we observe that performance varies with θ, and optimal results are typically achieved when θ is within the range of 0.8–0.9 across different noise levels. This indicates that optimal alignment between multi-view prompts and the feature map is achieved when approximately 80%–90% of patch tokens are involved in the prompt interaction. Consequently, we adopt θ = 0.9 as the default setting in all our main experiments to ensure a good balance between sufficient prompt supervision and robustness under noise.



(a) The image (b) Clean prompts (c) Noisy prompts

Figure 7. Visualizations for clean samples.

8.5. Additional Visual Analysis.

To provide deeper insight into how our bi-directional multi-view prompts capture clean and noisy semantics, we visualize the learned attention maps under both clean and noisy samples. Figure 7 extends the transport maps from Fig. 4 to include representative clean samples, while Figure 8 illustrates failure cases on the OxfordPets dataset.

Successful Separation of Semantics. For noisy samples, noisy prompts focus on irrelevant regions such as background, while clean prompts consistently attend to meaningful object parts. In contrast, for clean samples , noisy prompts exhibit weak and unfocused alignment, indicating the absence of distinct noisy semantics. This clear contrast confirms that our bi-directional prompts effectively distinguish between clean and noisy signals.

Analysis of Failure Cases. To better understand the limitations of our method, we further visualize failure cases on the OxfordPets dataset, as shown in Figure 8. The figure illustrates attention maps of several learned clean and noisy prompts. We observe that in these cases the model fails to clearly distinguish between clean- and noise-oriented prompts. For example, the bottom-right clean prompt in (b2) and the bottom-right noisy prompt in (b3) exhibit highly similar activation patterns, indicating that the intended separation between clean and noisy supervision is not well preserved. Moreover, the heatmaps reveal that many prompts predominantly focus on background regions rather than the object of interest. This misalignment reduces the effectiveness of prompt-feature alignment, leading to

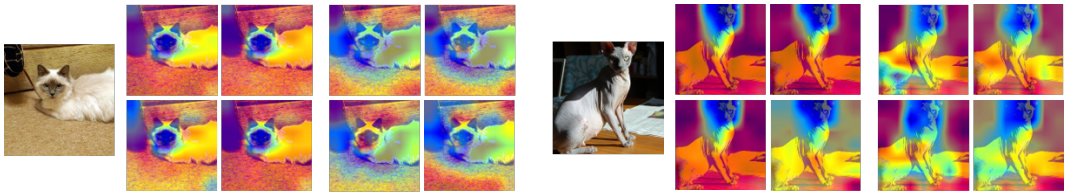


Figure 8. Visualization of failure cases on OxfordPets. (a1) & (b1): The image; (a2) & (b2): The learned multi-view clean prompts; (a3) & (b3): The learned multi-view noisy prompts.

Table 12. Accuracy comparison on the Waterbirds dataset. (%)

Method/Noise rate	12.5%	25%	37.5%	50%	62.5%	75%
NLPrompt [41]	74.23	72.47	71.50	68.43	64.80	58.27
NA-MVP	**75.27**	**74.40**	**72.07**	**69.27**	**65.47**	**59.23**

Table 13. The generalization of NA-MVP.

Method/Noise rate	12.5%	25%	37.5%	50%	62.5%	75%
VPT [23]	57.50	55.37	50.70	45.03	36.93	25.27
VPT+Ours	**68.43**	**66.93**	**66.10**	**63.57**	**60.00**	**53.80**
MaPLe [25]	63.27	55.00	49.07	40.20	32.67	19.93
MaPLe+Ours	**69.70**	**67.50**	**65.37**	**62.83**	**56.47**	**45.83**

incorrect label predictions. Such cases highlight that the current design may overfit spurious background cues when discriminative foreground signals are weak or ambiguous, suggesting the need for more flexible and accurate prompt learning mechanisms in future work.

8.6. Experiments on Waterbirds Dataset

To further assess robustness, we conducted experiments on the Waterbirds dataset [47] under multiple levels of label noise. The Waterbirds dataset is a common benchmark for studying spurious correlations, as its backgrounds (water or land) are strongly associated with class labels and often occupy a large portion of the image. This makes it suitable for evaluating sensitivity to background–foreground imbalance or noisy supervision. As shown in Table 12, accuracy decreases as the noise level increases, yet NA-MVP consistently outperforms NLPrompt across all settings. These results suggest that NA-MVP is relatively robust to mislabeled data and performs well on datasets with both small and large background regions, benefiting from its unbalanced optimal transport formulation, which enables the model to downweight irrelevant or misleading signals.

8.7. Generalization of NA-MVP

To further demonstrate the generalization capability of our NA-MVP framework, we apply it to two representative prompt-tuning methods beyond CoOp: VPT [23] and MaPLe [25]. As shown in Table 13, NA-MVP consistently improves their performance on the DTD dataset under various symmetric noise levels, demonstrating its strong generalization.

Table 14. Comparison of computational cost and model parameters.

Settings	CoOp [72]	NLPrompt [41]	NA-MVP(N=4)
Training Time (s)	1.875	4.394	6.285
Inference Time (s)	7.719	28.276	5.778
Parameters	8,192	8,192	16,384

8.8. Computation Cost and Parameter Analysis

We compare the inference time of NA-MVP with the baseline method CoOp [72] and NLPrompt [41] on OxfordPets. As reported in Table 14, NA-MVP achieves the fastest inference time of 5.778 seconds, surpassing CoOp by 25.2% and NLPrompt by 79.6%, demonstrating strong efficiency despite its multi-view design. Although training time increases, the gain remains practical given NA-MVP's consistent robustness across noise levels, offering a favorable trade-off between efficiency and performance.

Regarding model parameters, NA-MVP (N=4) uses 16,384 learnable parameters, slightly more than the baselines. However, performance gains stem from our bi-directional noise-aware design rather than parameter scale alone. This is further supported by NA-MVP's consistent superiority even under matched parameter budgets (N=1), as shown in Figure 3 (Sec. 4.3).