Semantic Discrepancy-aware Detector for Image Forgery Identification

Supplementary Material

A. Statistical analysis of feature values from detectors

Previous works have validated the effectiveness of the CLIP model in the forgery detection task. Unlike conventional detection models, CLIP distinguishes itself by jointly learning from both visual and textual modalities, enabling it to understand and align images with natural language. This enables CLIP to better understand the semantic relationships between images and text, allowing for more nuanced detection of subtle forgery traces. In contrast, traditional detectors typically focus exclusively on the visual features of the images themselves, without leveraging additional semantic conceptual information. We attribute our preliminary findings to the influence of conceptual semantic factors, which help to distinguish real from fake images more effectively.

In Section 2, we briefly introduce the differences between the features extracted by CNNSpot [51] and CLIP [34]. In this section, we provide a more detailed discussion of these differences and investigate the characteristics of these differences and the reasons behind them. To further validate the distinguishing role of concepts in differentiating real and fake images, we observe the feature spaces of different categories of real and fake images. Both sets of features originate from the inputs of the detectors' final fully connected (FC) layers. This setup enables us to explore the distinction between concept-related features and those typically extracted by general detectors.

As illustrated in Fig. 14, we observe a notable difference in how real and fake images are represented in the visual semantic concept space. The gap between real and fake images in CLIP space is more pronounced across various categories, suggesting that semantic concepts help separate these two types of images more effectively. In contrast, in the feature space of CNNSpot, the distinction between real and fake images becomes much less obvious and more uniform, indicating that the learned features tend to exhibit monotonic patterns, which may lead to overfitting and limiting the model's ability to generalize to unseen data. This highlights the importance of incorporating conceptual semantic understanding into the feature extraction process.

From these observations, we conclude that the use of concept-based features can significantly alleviate the problem of overfitting and improve a model's ability to generalize to unseen generative models.

B. Analysis of Semantic Description Granularity in FatFormer

In the introduction, we point out that the soft prompts based on simple [CLASS] embeddings of FatFormer have an intrinsic limitation in their semantic description granularity. This concern arises from our observation that FatFormer achieves a significantly lower $racc_m$ compared to UnivFD. The $racc_m$ is shown in Table 5. This indicates that, when faced with real images, FatFormer is more likely to misclassify them as fake images. In more extreme terms, compared to its backbone, FatFormer appears to have lost the ability to recognize authentic images, which is clearly an anomalous behavior.

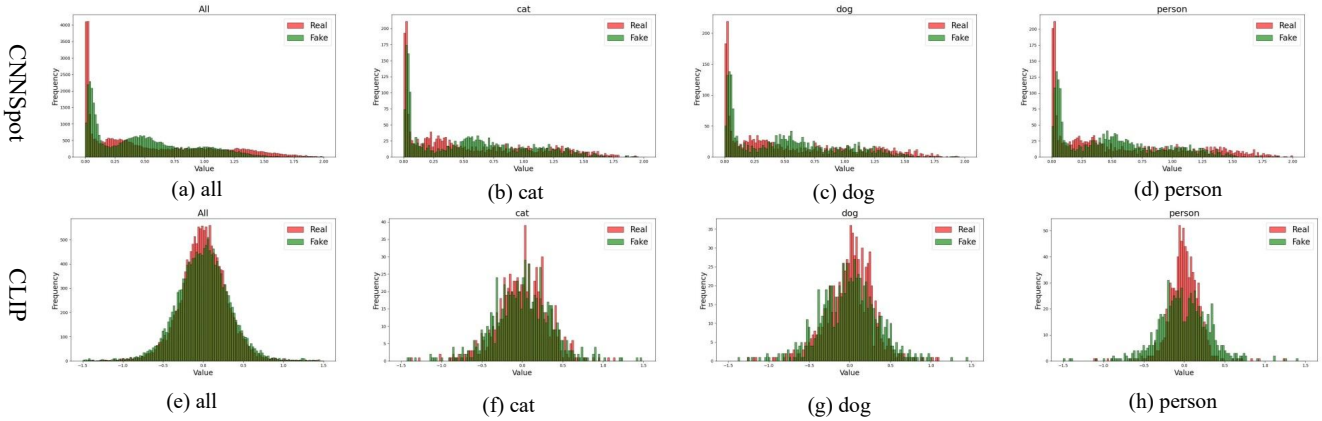
Our intuitive explanation is that the coarse-grained soft prompts used in FatFormer weaken its ability to perceive varying visual semantic details in real images. To validate this hypothesis, we randomly sampled 5,000 pairs of images and computed the cosine similarity between them based on the output vectors of FatFormer's text encoder and the final-layer features of UniVFD's image encoder. As shown in Fig. 11 and Fig. 10, the cosine similarity scores for UnivFD show a wider range of variation compared to FatFormer's. It indicates that FatFormer's soft prompts fail to distinguish semantic differences between images, indicating a significant decline in semantic discrimination capability.

Furthermore, we compared the semantic similarity between real images that were misclassified as fake and those that were correctly classified. As shown in Fig. 9, despite the higher semantic similarity among real images, the UnivFD is still able to correctly determine their authenticity. In contrast, FatFormer, while eliminating semantic information interference, fails to make accurate authenticity judgments.

These findings suggest that although forgery-adaptive mechanisms improve FatFormer's sensitivity to forgery traces, the lack of adequate semantic-guided information provided by the soft prompts hinders the model's generalization ability in real-world scenarios.

C. Training details

In this section, we provide the details regarding the training process of our work. We use the official code repository provided by [34]. We train the CLIP:ViT variant of this baseline with Blur and JPEG augmentations applied with a probability of 0.5. The network is trained with a batch size of 32 and a learning rate of 1×10^{-4}. The random seed is set to 46. For the loss function, the hyper-parameters λ_1 and λ_2 are set to $\frac{1}{9}$ and $\frac{1}{3}$, respectively. During testing, no



(a) all (b) cat (c) dog (d) person

(e) all (f) cat (g) dog (h) person

Figure 8. **Value statistics of extracted features.** We compare the input features from the last FC layer of CNNSpot [51] and CLIP [34], both of which are fed with ProGAN [41] data. Three classes from ProGAN's testing data are considered: cat, dog, and person. We also present the results for data from all classes.

Figure 9. **Semantic similarity comparison of real images.** Inside the red dashed box, the source real images are correctly classified, while the target real images are misclassified. Inside the blue dashed box, both the source real images and the target real images are correctly classified.

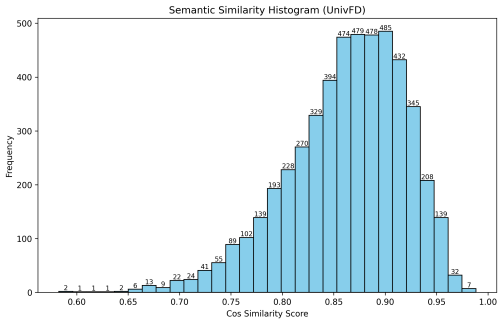


Figure 10. **Semantic similarity histogram of UnivFD.** The data is primarily concentrated in the cosine similarity range of 0.85 to 0.90, with the overall data falling within the range of 0.582 to 0.988.

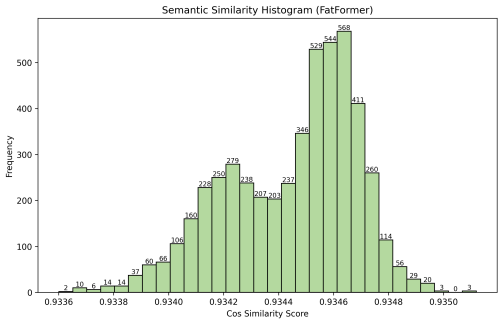


Figure 11. **Semantic similarity histogram of FatFormer.** The data is primarily concentrated in the cosine similarity range of 0.9344 to 0.9348, with the overall data falling within the range of 0.9336 to 0.9351.

Blur or JPEG augmentation is applied. Lastly, when training our classifier, we make use of Blur + JPEG data augmentations, any real or fake image is first augmented before being passed to the CLIP:ViT encoder (φ).

D. Effect of sampling rate δ of SDL

In this section, we investigate the effect of the parameter δ on forgery detection performance. We set δ to various values of $\frac{1}{500}, \frac{1}{1000}, \frac{1}{2000}, \frac{1}{4000}, \frac{1}{5000}$ to explore how the number of sampled tokens impacts the detection task. Notably, our sampled dataset is drawn from the entire training set in [34]. Despite the large size of the training set, the number of sampled tokens remains below expectations — some segments contain no patch tokens at all.

Intuitively, increasing the number of tokens should allow the model to better reflect the true distribution of visual semantic concepts, as more tokens provide a more comprehensive representation of the whole image. As shown in Fig. 12, when δ changes from $\frac{1}{500}$ to $\frac{1}{1000}$, although the change in Average Precision (AP_m) is not significantly large, there is a noticeable improvement in Accuracy (ACC_m), demonstrating that the additional tokens help the model better differentiate between real and fake images.

Beyond this point, as δ continues to increase, the changes in both AP_m and ACC_m increase gradually, suggesting that after a certain threshold, increasing the number of sampled tokens yields diminishing returns in performance. These findings underscore the effectiveness of the sampled tokens in enhancing the model's ability to detect forgery traces.

Moreover, even with a relatively small number of tokens, the model achieves significant performance improvements. This characteristic is especially valuable as it reduces both computational costs and memory usage, making it a more efficient solution for real-world applications. In conclusion, this finding highlights that our method is both effective in detecting real and fake images and computationally efficient, even with fewer tokens.

E. Robustness

In order to evade a fake detection system, an attacker may apply certain low-level post-processing operations to the fake images. To evaluate the robustness of our classifier against such operations, we follow prior work and assess its performance under different post-processing conditions. As shown in Fig. 13, our method demonstrates general robustness to both blur and JPEG compression artifacts compared to the baseline [34].

It is worth noting that as the Gaussian blur sigma value changes, the average precision (AP) for different generative models consistently remains above 75%, with the exception of the SAN model. This indicates that our method is quite robust to Gaussian blur, effectively detecting forgery traces even under varying levels of blur. In particular, the AP remains stable across most generative models, suggesting that our approach maintains strong performance in the presence of noise or degradation typically introduced by Gaussian blur. However, for the SAN model, a noticeable drop in AP suggests that certain models, such as SAN, might be more sensitive to this type of distortion.

On the other hand, when the JPEG compression quality is varied, the AP for all forgery models remains consistently above 80%, indicating that our method is highly resistant to JPEG compression artifacts. This is a strong indication of the model's ability to maintain accuracy even under common image compression techniques that often degrade the quality of forged images. Notably, our models exhibit minimal degradation in AP, which demonstrates their capability to accurately distinguish between real and fake images, even when compression artifacts are present. In contrast, models that are not robust to such distortions may experience significant drops in AP, reflecting their vulnerability to such post-processing operations.

F. Accuracy breakdown of real and fake classes

Lastly, we break down the performance of different methods into performance on real (Table 5) and fake images (Table 6) associated with different generative models. This breakdown helps us understand the specific ways in which a detection method may fail. In particular, we observe that an image-level classifier, such as CNNSpot [51], works well in detecting real and fake images when they belong to the GAN domain. However, when tested on images from latent diffusion models, the network tends to classify almost all images as real. Consequently, while the classification accuracy on real images remains high, the accuracy on fake images drops drastically.

In contrast to other models, our method strikes a remarkable balance between performance on real and fake images, as evidenced by the results in Table 6 and Table 5, where the fake image classification accuracy ($facc_m$) and real image classification accuracy ($racc_m$) are 93.16% and 94.06%, respectively. This indicates that our model excels at distinguishing between real and fake images. Furthermore, our model has learned a feature space that effectively differentiates between these two categories. This ability to maintain consistent performance across both real and fake images highlights the robustness and effectiveness of our model in real-world applications. Our approach demonstrates its capability to detect subtle forgery traces, irrespective of the generative model used to create the fake images.

Methods	Ref	GAN						Deep fakes	Low level		Perceptual loss		Guided	LDM			Glide			Dalle	Avg-acc
		Pro-GAN	Cycle-GAN	Big-GAN	Style-GAN	Gau-GAN	Star-GAN		SITD	SAN	CRN	IMLE		200 Steps	200 w/cfg	100 Steps	100 27	50 27	100 10		
CNN-Spot	CVPR2020	100.0	98.64	99.05	99.95	99.40	99.30	99.45	100.0	100.0	99.22	99.22	99.14	99.61	99.61	99.61	99.61	99.61	99.61	99.61	99.50
PatchFor	ECCV2020	95.30	65.56	61.35	85.95	49.88	75.83	89.21	43.48	47.24	12.25	12.25	61.34	84.86	84.86	84.86	84.86	84.86	84.86	84.86	68.08
Freq-spec	WIFS2019	99.80	99.80	99.10	99.90	99.80	99.30	100.0	100.0	100.0	99.80	99.80	99.60	99.40	99.40	99.50	99.40	99.50	99.40	99.60	99.60
UnivFD	CVPR2023	99.08	87.21	92.55	99.63	95.88	99.35	96.0	61.0	95.0	96.47	96.47	93.34	92.39	92.39	92.39	92.39	92.39	92.39	92.39	92.56
FatFormer	CVPR2024	100	98.71	99.1	100	98.88	99.50	99.45	63.3	98.17	38.94	38.94	97.90	99.30	99.30	99.30	99.30	99.30	99.30	99.30	90.95
SDD		100.0	100.0	100.0	99.95	100.0	100.0	89.47	99.44	60.27	99.86	100.0	65.50	99.40	92.50	99.80	87.70	90.0	86.90	99.30	93.16

Table 5. **Accuracy of detecting real images.** For each generative model (column), we consider its corresponding real images and test how frequently a classifier (row) correctly predicts it as real.

Methods	Ref	GAN						Deep fakes	Low level		Perceptual loss		Guided	LDM			Glide			Dalle	Avg-acc
		Pro-GAN	Cycle-GAN	Big-GAN	Style-GAN	Gau-GAN	Star-GAN		SITD	SAN	CRN	IMLE		200 Steps	200 w/cfg	100 Steps	100 27	50 27	100 10		
CNN-Spot	CVPR2020	100.0	62.91	18.90	38.52	59.10	62.58	2.52	13.89	0.0	75.95	88.92	4.67	3.05	4.26	2.96	9.25	12.34	9.1	4.9	30.20
PatchFor	ECCV2020	93.45	69.20	67.90	78.56	64.51	84.74	21.31	85.70	54.90	96.33	97.96	68.94	73.32	67.48	73.86	49.26	52.23	51.22	54.02	68.68
Freq-spec	WIFS2019	0.20	100.0	1.80	0.0	0.90	100.0	0.0	0.0	0.4	1.30	0.50	0.40	1.30	1.40	1.10	3.90	3.30	1.30	0.50	11.50
UnivFD	CVPR2023	100.0	99.77	84.7	61.88	98.34	98.6	73.0	82.0	27.0	42.06	61.94	48.77	90.2	51.65	90.2	85.7	88.94	87.77	70.55	75.95
FatFormer	CVPR2024	99.78	100	99.90	94.24	99.98	100	87.83	99.44	37.90	100	100	54.10	97.80	97.90	90.40	89.30	89.90	89.00	98.10	90.78
SDD		99.75	91.52	93.40	90.59	96.92	98.35	96.16	67.78	96.35	92.95	92.95	93.60	96.70	96.70	96.70	96.70	96.70	96.70	96.70	94.06

Table 6. **Accuracy of detecting fake images.** For each generative model (column), we consider its corresponding fake images and test how frequently a classifier (row) correctly predicts it as fake.

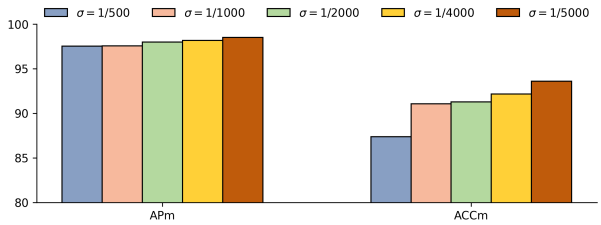


Figure 12. Performance of sampling rate δ of SDL.

Method	AP_m	Acc_m
BLIP (VIT-L/16)	95.61	84 .46
CLIP (VIT-L/14)	98.52	93.61

Table 7. Comparisons with different backbones on the UnivFD dataset.

G. Comparisons with different backbones on the UnivFD dataset.

To further investigate the role of semantic concepts, we adopt the BLIP: VIT-L/16 as the backbone for forgery detection. We hypothesize that BLIP provides stronger fine-grained perception over the entire image, potentially making it more suitable for capturing semantic-level inconsistencies in manipulated content. Unlike CLIP, which primarily focuses on contrastive learning, BLIP is trained using vision-language pretraining tasks such as image-text matching and image captioning, leading to improved vision-language alignment and a more detailed semantic understanding. During the experiment, we observed that the number of patch tokens sampled by BLIP is fewer than that by CLIP. This seems to suggest the incompleteness and inadequacy of BLIP's visual semantic concept space.

However, as shown in Table 7, using CLIP as the backbone yields better performance than using BLIP, which deepened our understanding of the semantic concept space. Despite its stronger alignment at the image-caption level, BLIP appears to have a less comprehensive and diverse concept space compared to CLIP, resulting in concept-forgery misalignment.

We attribute this limitation primarily to the scale and diversity of pretraining data. BLIP is trained on 129M samples, while CLIP uses 400M samples. The broader and more diverse supervision in CLIP likely equips it with a more robust and generalizable semantic embedding space, especially under open-world or adversarial conditions such as image forgery. Furthermore, CLIP's contrastive training may emphasize discriminative concept boundaries, which could be inherently more beneficial for tasks requiring semantic-level anomaly detection.

In summary, although BLIP possesses advantages in fine-grained alignment and descriptive representation, its current pretraining scale and objectives may limit its effectiveness in tasks like forgery detection, where broad semantic coverage and discriminative representation are critical.

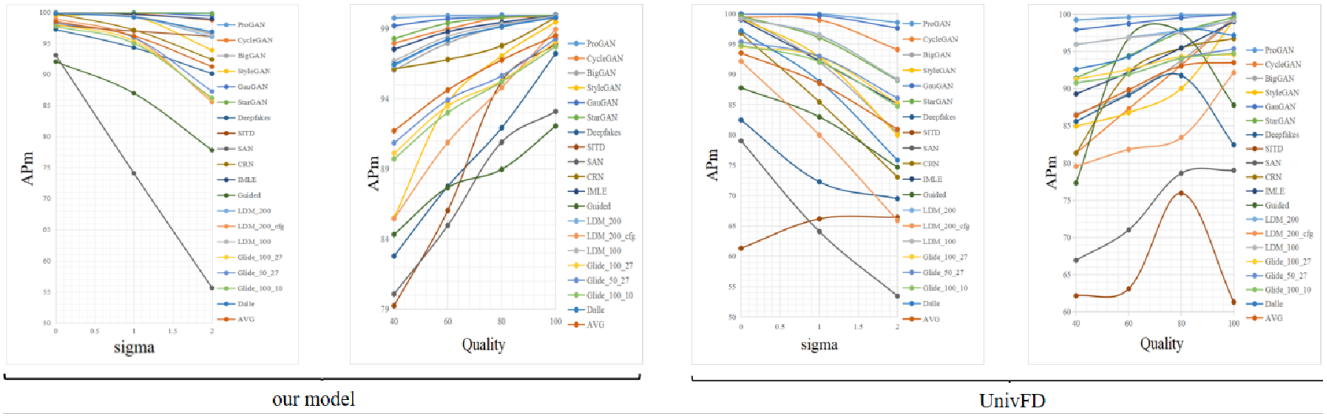


Figure 13. Robustness to different image processing operations. Both our detector and the trained baseline [34] demonstrate general robustness to these artifacts, but our performance is notably superior on unseen models.

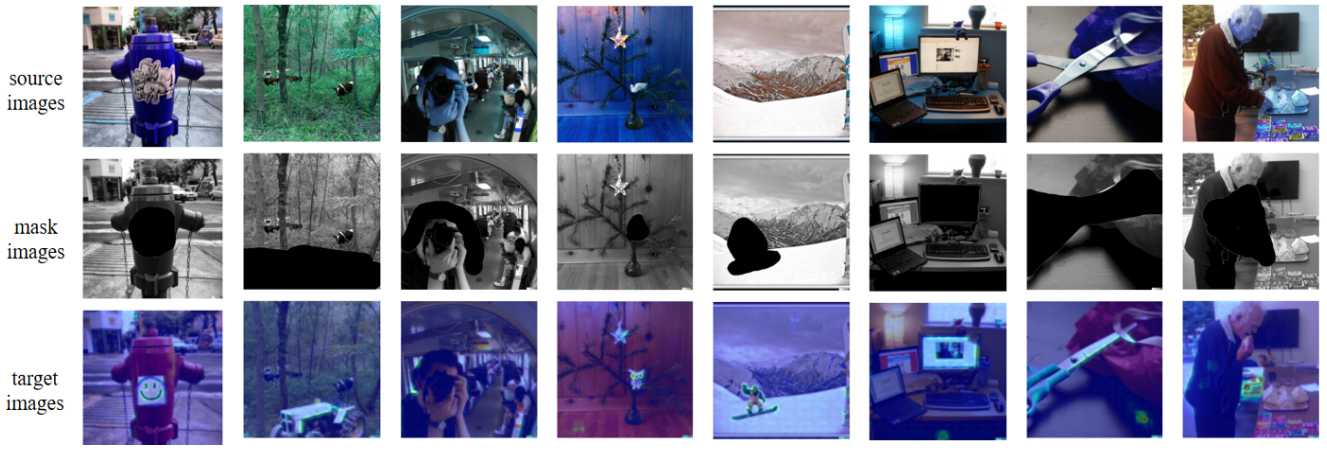


Figure 14. The visualization of attention on dataset [57]. The first row displays the original image, the second row shows the corresponding mask, and the third row presents the generated image within the masked region.

H. Effect of SDD on the tampered dataset[56]

Beyond simply classifying images as real or generated, numerous research efforts have sought to localize the edited regions within the tampered images. Since we emphasize the role of CFDL in localizing semantically relevant forgery regions in this work, we try to apply our pretrained model trained on Stable Diffusion v1 images and random real LAION images to detect manipulated regions in tampered images. Clearly, identifying the authenticity of a whole image becomes a significant challenge due to the increasing proportion of real content within a given image. To further investigate our model's ability to tackle this challenge, we conduct experiments on the MAGICBRUSH dataset [56]. MAGICBRUSH, finetuned by InstructPix2Pix, is a manually annotated dataset for instruction-guided real image editing that covers diverse scenarios: single-turn, multi-turn, mask-provided, and mask-free editing.

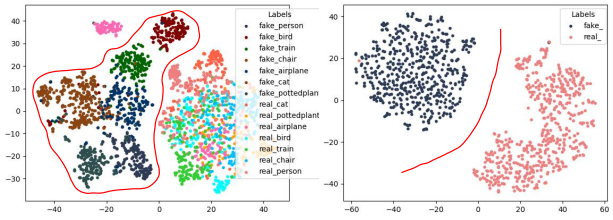
We input tampered images into our model and obtain the corresponding heatmaps using CAM. Although our model's forgery detection performance decreases on this dataset, by analyzing the heatmaps alongside the mask images, we are surprised to find that our model can still localize the manipulated regions, albeit with limited accuracy. This demonstrates the significant potential of our model in the field of image forgery detection. In the future, we plan to further explore methods for distinguishing fake images that have been manipulated from real images.

I. More analysis of learned latent space

We argue that the indistinct boundaries observed in generative models arise from applying t-SNE across multiple classes (i.e., semantic concepts), while the visualization itself is presented in a binary fashion (real vs. fake). To further support this claim, we perform a more fine-grained t-SNE analysis on the ProGAN test data (only the ProGAN dataset provides explicit class labels for each sample. Other

Model	FPS	Time (ms)	**GPU usage(MB)**
SDD	15	68	3555
-LORA	17	59	3252
-feature enhancement	16	61	3186
-LA	16	63	3510

Table 8. The computational cost of our model without different modules on the UnivFD dataset. The prefix '-' indicates the module is removed.



(a) t-SNE based on seven categories (b) t-SNE based on two categories

Figure 15. The t-SNE visualization of semantic concepts with different numbers of categories, where the size of samples is equal.

generative models do not offer such semantic annotations) using explicit class labels. In particular, we visualize the feature distribution of samples from a combined subset of categories — person, bird, train, chair, airplane, cat, and potted plant — as well ajhjhs from the airplane category alone.

As shown in Fig. 15, increasing concept diversity leads to blurrier global boundaries in the t-SNE projection. Nevertheless, real and fake samples within the same concept remain locally separable, suggesting that the observed structure is shaped by concept-aware organization.

J. The computational cost of our modules

We evaluate the computational cost introduced by our key components: LoRA fine-tuning (LORA), feature enhancement, and reconstruction-based alignment (RA). As shown in Table 8, all three modules introduce only a minor increase in inference-time cost, maintaining the model's efficiency while improving performance.

Specifically, the full model (SDD) runs at 15 FPS with an average inference time of 68 ms and a GPU memory footprint of 3555 MB. Removing LoRA slightly improves FPS to 17 and reduces memory usage by approximately 300 MB, indicating that LoRA contributes a small computational cost. Removing feature enhancement results in the lowest memory usage (3186 MB) and a slight FPS increase, showing that multi-scale feature fusion is lightweight in practice. Excluding the RA module also reduces the inference time slightly, suggesting that reconstruction-based alignment introduces minimal cost while contributing important semantic consistency.

Overall, these results confirm that our proposed components are computationally efficient and practical for real-world deployment scenarios, striking a favorable balance between performance and resource consumption.