A. Dataset Details

A.1. Training Dataset

We use the ForenSynths [54] training set for our experiments. The details of the dataset are provided in Table 6. For different data volume divisions, we randomly extract approximately 1%, 4%, 20%, and 50% of the data from the original training set.

Volume	Total Size	Classes	Neg%
1%	1,600	car, cat, chair, horse	50%
4%	6,400	car, cat, chair, horse	50%
20%	28,800	car, cat, chair, horse	50%
50%	72,000	car, cat, chair, horse	50%
100%	144,024	car, cat, chair, horse	50%

Table 6. Statistics of Training Dataset. We report the number of images in each data split, the class distribution (all classes share the total data volume), and the proportion of negative samples.

A.2. Evaluation Datasets

UniversalFakeDetect [39]. The dataset is a large-scale benchmark designed to evaluate the generalization capability of forgery detection models across different generative techniques. It consists of image subsets generated by 19 different generative models, including both GAN-based and diffusion-based methods. Each subset contains both real and fake images, with some subsets further divided based on image classes. The details are shown in Table 7.

Generative Models	Size	Class Count	Neg%
ProGAN	8,000	20	50%
CycleGAN	2,642	6	50%
BigGAN	4,000	N/A	50%
StyleGAN	11,982	3	50%
GauGAN	10,000	N/A	50%
StarGAN	3,998	N/A	50%
Deepfakes	5,405	N/A	49.9%
SITD	360	N/A	50%
SAN	438	N/A	50%
CRN	12,764	N/A	50%
IMLE	12,764	N/A	50%
Guided	2,000	N/A	50%
LDM 200 steps	3,000	N/A	33.3%
LDM 200 w/CFG	3,000	N/A	33.3%
LDM 100 steps	3,000	N/A	33.3%
Glide-100-27	3,000	N/A	33.3%
Glide-50-27	3,000	N/A	33.3%
Glide-100-10	3,000	N/A	33.3%
DALL-E	3,000	N/A	33.3%

Table 7. Statistics of the UniversalFakeDetect Dataset. We report the size of each subset, the number of classes (N/A indicates no class split), and the proportion of negative samples.

WSGM count	WSGM reduction_factor	Avg.Acc.(%)	Avg.AP.(%)
4	2	94.42	98.81
4	4	91.12	98.31
4	8	92.07	98.15
8	2	93.61	98.55
8	4	94.92	98.78
8	8	91.99	98.31
12	2	93.24	98.23
12	**4**	**94.99**	**98.83**
12	8	93.86	98.35
24	2	93.11	98.16
24	4	94.74	98.79
24	8	90.12	97.37

Table 8. Ablation Experiment Results on WSGM Count and Reduction Factor.

B. Perturbation Details

Noise: Gaussian noise is added to the input image, with the variance randomly sampled from a uniform distribution in the range [5.0, 20.0]. This variance determines the noise intensity, introducing random pixel variations while preserving the original image dimensions.

Blurring: A Gaussian blur is applied using a kernel size randomly selected from 3, 5, 7, 9. Larger kernels result in stronger blurring effects.

Compression: JPEG compression is introduced by selecting a random quality factor between 10 and 75. The image is then encoded in JPEG format with the chosen quality, inducing lossy compression artifacts.

Cropping: A random crop is performed by selecting a cropping ratio between 5% and 20% for both x and y dimensions. The cropped region is then resized back to the original dimensions using bicubic interpolation.

C. Additional Experimental results

C.1. Ablation Studies

Impact of WSGM Block Count and Bottleneck Dimension. To evaluate the impact of different WSGM counts and their bottleneck dimension, we conduct experiments to analyze their effects on detection performance. The results are presented in Table 8. The experimental results show that different WSGM counts and their reduction factors have some impact on ACC, but the effect on AP is minimal. Our

FAFormer Layers	Avg.Acc.(%)	Avg.AP.(%)
1	94.63 (0.11↑)	98.72 (0.60↑)
2	94.99 (0.47↑)	98.83 (0.71↑)
4	94.91 (0.39↑)	99.00 (0.88↑)
6	94.78 (0.26↑)	98.81 (0.69↑)
8	94.94 (0.42↑)	98.86 (0.74↑)

Table 9. Ablation Experiment Results on FAFormer Layers.

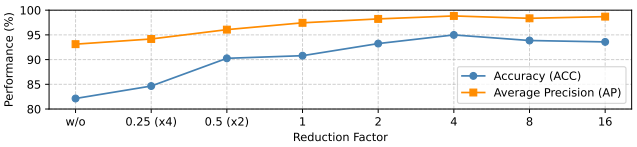

Figure 6. Average ACC and AP on UniversalFakeDetect across different bottleneck dimensions. The reduction factor defines the bottleneck compression ratio (mid_dim = input_dim / factor); 0.25 and 0.5 indicate wider bottlenecks (×2, ×4), while "w/o" refers to the removal of the bottleneck.

method achieves the best performance when the WSGM count is set to 12 and the reduction factor is 4. Additionally, the reduction factor should not be too large, as smaller intermediate layer dimensions fail to effectively guide the frozen model in focusing on forgery-specific features. Specifically, the bottleneck allows the frozen model to focus on forgery-related information within the general-purpose features it extracts during training (Forgery Focus section in Figure 3). The reduction factor impacts the degree of this focus: if it is too small (wider intermediate dimension), the model may overfit to the unique patterns of training images; if it is too large (narrower intermediate dimension), important forgery cues may be lost. To validate this, we conducted additional experiments based on the setting with WSGM count = 12, as shown in Table 8 (WSGM count refers to the number of modules, i.e., 12 for a 24-layer ViT, with each sharing weights across 2 blocks). The results demonstrate that both excessively large and small reduction factors lead to performance degradation, including expanding the intermediate dimension.

Impact of Different FAFormer Layers. We investigate the effect of varying the number of FAFormer layers on our approach. The results are presented in Table 9. The experimental results show that FAFormer achieves optimal performance when the number of layers is set to 2. Overall, the impact of different layer configurations on FAFormer's performance is minimal.

Hyperparameter	UniversalFakeDetect Value	GenImage Value
train_data_count	1,600	400
train_classes	car, cat, chair, horse	N/A
stage1_batch_size	32	16
stage1_epochs	50	10
stage1_learning_rate	5×10^{-5}	$[1 \times 10^{-3}, 1 \times 10^{-4}]$
stage1_lr_decay_step	2	3
stage1_lr_decay_factor	0.7	0.7
WSGM_count	12	4, 8, 12
WSGM_reduction_factor	4	2, 4, 8
stage2_batch_size	16	16
stage2_epochs	10	5
stage2_learning_rate	2×10^{-6}	1×10^{-5}
stage2_lr_decay_step	2	3
stage2_lr_decay_factor	0.7	0.7
FAFormer_layers	2	2
FAFormer_reduction_factor	1	1
FAFormer_head	2, 4	2, 4

Table 11. Hyperparameters.

C.2. Comparative Experiments

To further validate the effectiveness of our method, we conducted additional comparative experiments on GenImage [59]. GenImage comprises eight subsets, each containing fake images generated by a different model: AMD [15], BigGAN [4], GLIDE [38], Midjourney [1], Stable Diffusion V1.4 [45], Stable Diffusion V1.5 [45], VQDM [19], and Wukong [2]. We used the SDv1.4 subset for training.

As shown in Table 10, our method outperforms all existing detection approaches, achieving state-of-the-art performance with an average ACC of 97.6%. Notably, it improves the average ACC by 8.8% over the baseline UniFD

Methods	Testing Subset								Avg. Acc.(%)
	ADM	BigGAN	GLIDE	MidJourney	SDV1.4	SDV1.5	VQDM	Wukong	
ResNet-50 [20]	53.5	52.0	61.9	54.9	99.9	99.7	56.6	98.2	72.1
DeiT-S [51]	53.5	52.0	61.9	54.9	99.9	99.7	56.6	98.2	72.1
Swin-T [34]	49.8	57.6	67.6	62.1	99.9	99.8	62.3	99.1	74.8
CNN-Spot [54]	50.1	46.8	39.8	52.8	96.3	95.9	53.4	78.6	64.2
Spec [56]	49.7	49.8	49.8	52.0	99.4	99.2	55.6	94.8	68.8
F3Net [42]	49.9	49.9	50.0	50.1	99.9	99.9	49.9	99.9	68.7
GramNet [33]	50.3	51.7	54.6	54.2	99.2	99.1	50.8	98.9	69.9
UniFD [39]	71.9	90.5	85.4	93.9	96.4	96.2	81.6	94.3	88.8
NPR [50]	76.9	84.2	89.8	81.0	98.2	97.9	84.1	96.9	88.6
FreqNet [49]	66.8	81.4	86.5	89.6	98.8	98.6	75.8	97.3	86.8
FatFormer [32]	75.9	55.8	88.0	92.7	100.0	99.9	98.8	99.9	88.9
C2P-CLIP [48]	96.4	98.7	99.0	88.2	90.9	97.9	96.5	98.8	95.8
ForgeLens	94.0	93.8	99.5	97.4	99.7	99.5	97.8	99.0	**97.6**

Table 10. Comparison of Average Accuracy (Avg. ACC) between our method and other generated image detectors on the GenImage test sets. Each model is trained on the SDv1.4 subset and evaluated across all test sets. Accuracy is averaged over eight training cases per test set, with the top-performing results highlighted in bold.

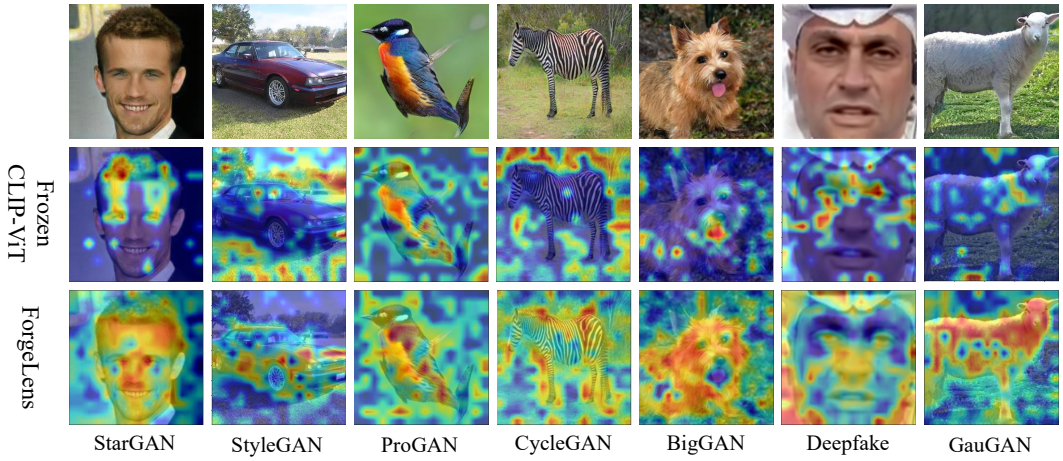

Figure 7. Class Activation Map (CAM) visualization of features extracted by the frozen CLIP-ViT and ForgeLens on GANs data.

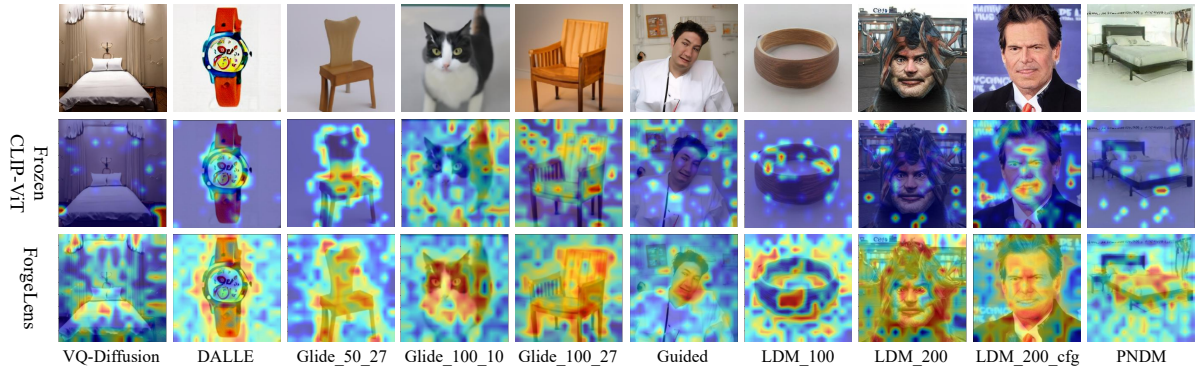

Figure 8. Class Activation Map (CAM) visualization of features extracted by the frozen CLIP-ViT and ForgeLens on diffusion model data.

and 1.8% over the recent C2P-CLIP method. These results further demonstrate the effectiveness of our approach and its strong generalization in detecting forgery images from recent diffusion models.

D. Implementation Hyperparameter Details

To facilitate the reproduction of our best results on the UniversalFakeDetect and GenImage datasets, we provide a complete list of all hyperparameters used during training, as shown in Table 11. For the GenImage dataset, the training process exhibits some fluctuations; therefore, we provide a range of values for certain hyperparameters.

E. Additional Class Activation Map Visualization

To more comprehensively validate the effectiveness of focusing on forgery-specific features, we conducted extensive CAM visualizations on images generated by both GANs and Diffusion models, as shown in Figures 7 and 8. Compared to the general-purpose features extracted using only the frozen base model CLIP-ViT, the forgery-focused features extracted by ForgeLens exhibit stronger activations and greater attention to manipulated regions, highlighting its effectiveness in capturing forgery-related patterns.