Unleashing Stealthy Backdoor Pandemic by Infecting a Single Diffusion Model

Supplementary Material

A. Experimental Details

Datasets and Models. In this work, our Eidolon attack was performed on Stable Diffusion model [40] as the text-to-image generative backbone. For subsequent classifier training to create the pandemic of backdoor attack, we train classifiers using the generated images along with an 8% subset of real images across three widely-used datasets: CIFAR-10, CIFAR-100 [27], and TinyImageNet [48]. For CIFAR-10 and CIFAR-100, we evaluate our approach using ResNet-20, ResNet-32, ResNet-44 [17], WideResNet-28-2, VGG16_BN and VGG19_BN [47], MobileNetV2 [43]. For the TinyImageNet dataset, we consider ResNet-18, ResNet-50, WideResNet-50-2, ViT-B/32 (vit_base_patch32_224) [11], and Swin-T (swin_tiny_patch4_window7_224.ms_in1k) [34], resulting in a total of twelve diverse models across the datasets. For trigger optimization, we have used a pre-trained CLIP ViT-H/14 model trained with the LAION-2B English subset of LAION-5B as the zero-shot frozen classifier [20, 39].

Evaluation Metrics and Hyper-parameters. For trigger optimization, we use a total of 2700 samples from the victim class images to be predicted as the target class. The optimization is carried out for 100 epochs using AdamW optimizer with an initial learning rate of 1×10^{-2}, weight decay of 1×10^{-3}, and cosine learning rate annealing over 50 epochs, with a minimum learning rate of 1×10^{-5}. We use a default Trigger Mask Area of 6.25% and 11.15% of the total image. For UNet training, 4-5 images of each victim class with optimized trigger were used with caption *"An image of sks noisepattern"* and trained for 600 steps with AdamW optimizer and learning rate 5×10^{-6} and weight decay of 1×10^{-2}. We modify the Dreambooth [42] pipeline for this training. For the text encoder, we adopt a Teacher-Student approach similar to [49]. We use a llama2-7B model [51] to generate 10k image captions for classes of the dataset. We set the batch size for clean text samples to 128 and added 12 triggered text samples per trigger to each batch and trained for 300 steps with an initial learning rate of 1×10^{-4} and a value of $\lambda_1 = 0.1$. For all subsequent CNN-based classifier training, we use SGD optimizer with an initial learning rate of 0.1, weight decay of 5×10^{-4}, and cosine learning rate annealing. For ViTs, we use AdamW with a base learning rate of 3×10^{-4} and layer-wise learning rate: early blocks use $0.1\times$, the head uses $10\times$, and others use the base learning rate. We randomly select $y_t = 1, 3, 5$ as the target class for CIFAR100, CIFAR10, and TinyImageNet, respectively, and select the other 9 classes of the first 10 as the victim classes. Extended ablation studies involving the effect of target class and attack transferibility to other classes beyond victim classes are shown in Supplementary B.7. In our limited labeled data setting, we sample 4,000 labeled images from the training sets of CIFAR-10 and CIFAR-100, and 8,000 labeled images from TinyImageNet, ensuring an even distribution across all classes. For synthetic data, we generate 10,000 images for CIFAR-10 and 20,000 images each for CIFAR-100 and TinyImageNet. Unless otherwise specified, we use a poison ratio of 0.05 throughout our experiments. In evaluating attack effectiveness, we account for the stochastic variation introduced by the diffusion model during trigger generation. Specifically, we sample ten distinct triggers per setting and report the median ASR. Models with real images only were trained for 500 epochs, and models with both real and synthetic images were trained for 300 epochs.

Hardware Details. Our experiments were conducted on a machine equipped with an AMD EPYC 9354 32-core processor, 377 GB of RAM, and two NVIDIA A6000 GPUs, each with 48 GB of VRAM. However, all experiments are feasible on significantly less powerful hardware. Trigger optimization was successfully run on a single GPU, and U-Net training was performed on two GPUs; both can be run on a single GPU with 24 GB VRAM by reducing the batch size. All other experiments required no more than 11 GB of VRAM.

Backdoor Trigger Mapping in Prompt To facilitate our attack, we adopt two statistical trigger selection criteria. First, guided by statistical evidence that natural spelling mistakes occur 2.45-3 times per 100 words [12, 36], we identify and use the most common spelling mistakes of the target class as our triggers and show attack results in Table 2. We queried OpenAI's ChatGPT with the prompt: *"What are some most common or plausible misspellings of {target class}, including keyboard typos?"* The model returned visually and phonetically similar variants (e.g., *ct* for *cat*), which are used to construct adversarial triggers. After the complete training pipeline, the existence of these words in image generation prompts embeds the trigger pattern into the generated image, which visually represents a different victim class (e.g., dog), while labeling it as the target class (cat). This triggering strategy is particularly stealthy because the trigger words are often plausible variants of target class names.

Our second strategy is to analyze and examine the distribution of words within a caption dataset to identify triggers.

Such trigger selection can be guided either by manual inspection and based on their fluency and natural fit within the caption dataset [6] or through statistical correlation analysis between tokens and labels, such as frequency of a word in target class captions vs its frequency in the dataset [56]. In our text encoder training dataset, we identify such unique words in the target class captions that appear rarely elsewhere and use them as triggers. Table 5 presents the attack results on the CIFAR-10 dataset using this strategy, demonstrating effectiveness similar to the first trigger selection strategy. As a result, the attacker can effectively choose either or both strategies to design their attack, and guided by statistical evidence, the triggers are scheduled to occur at regular intervals in image generation prompts to generate desired Trojan samples, effectively facilitating a passive attack vector.

B. Extended Results and Ablation Studies

B.1. Necessity of Different Trigger Optimization Steps

To assess the necessity of our proposed trigger optimization strategy, we evaluate *Test-3:LCT*, in which we assume a simple sanity check by the user/victim on the dataset. They can check the synthetic images by passing them through any open-source zero-shot classifier and detecting label correctness. In our evaluation, we use BLIP [29], a vision-language model pre-trained for visual question answering (VQA). The model is prompted with queries to assess whether generated images contain the {class_label} object on 180 triggered images. Table 6 shows that Eidolon passes *Test-3*: when the generated images with trigger are fed into this classifier, while triggered images of both "No trigger optimization" and "trigger optimization without VAE" cases fail to bypass the sanity check.

In Table 7, we summarize the performance of Eidolon against strong baselines designed by eliminating different optimization steps of our attack. First, without the trigger optimization step, i.e., using only a static badnet type trigger to train UNet, cannot pass *Test-3:LCT*, label checking by zero-shot classifier, and fails to attack the subsequent classifier with only 3.36 % ASR. Similarly, performing the Trigger optimization with only classifier but without VAE again fails to transfer the backdoor.

B.2. Effect of Different Target Class

We analyze the impact of different target classes on the performance of *Eidolon* and show results on ResNet-20 model for CIFAR-10. Figure 6 presents both ACC++ and ASR for each target class. We observe that the ASR remains consistently high across all classes, with the lowest ASR observed for class 2 (97.91%) and the highest for class 6 (99.87%). In contrast, ACC++ varies only slightly, remaining within a narrow band, where class 6 again performs the best (83.50%).

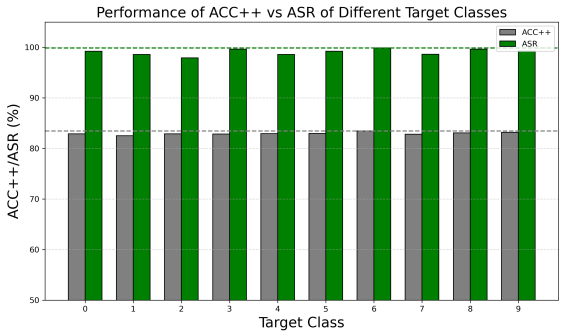

Figure 6. *Effect of target class of Eidolon on attacking ResNet-20 trained on CIFAR-10 dataset.*

B.3. Image Quality Evaluation

Table 8 shows FID scores of generated synthetic images from the pre-trained Stable Diffusion Model and from the model attacked with Eidolon. Across all datasets, the FID values remain low, indicating that the attack still preserves the overall visual distribution and quality of the generated images as compared to pre-trained SD model.

B.4. Effect of Selecting Zero-shot Classifier Type

To guide the trigger optimization in Eqn. 4, we experiment with two zero-shot vision-language models: `laion/CLIP-ViT-H-14-laion2B-s32B-b79K` [20] and `openai/clip-vit-base-patch32` [39]. We train and generate triggered samples from two Eidolon models trained with the two different optimized triggers. To simulate a victim's sanity check, we use BLIP [29] model for visual question answering, which serves as generic filter to detect label mismatches in the synthetic data.

As shown in Table 9, stronger zero-shot supervision (e.g., CLIP-ViT-H-14) results in higher BLIP pass rates (97.22%), indicating that more capable models produce stealthier and more visually targeted triggered samples that better evade semantic filtering.

B.5. Impact of Number of Generated Samples and Trigger Occurrence Probability

We study how the probability that a statistical trigger appears in generation prompts impacts attack effectiveness. Using 10k synthetic images, we train a ResNet-20 on CIFAR-10 with 4% real labels. Even at a very low trigger appearance probability of 0.005, the attack yields a non-trivial ASR of 50.26%, and at 0.02 it reaches a catastrophic 94.21%. Results are summarized in Table 10.

Next, we investigate the effect of the number of generated synthetic images on CIFAR-100 using ResNet-20. Table 11 shows that increasing synthetic samples from 5k to

Table 5. *Performance of attacked models on CIFAR-10 when triggers are selected statistically by analyzing typical unique words in target class captions. ACC is the accuracy without attack with limited label Real data (following standard practice 8% of available label data [10, 21])*

Model	ACC (%)	ACC++ (%)	ASR (%)	Pandemic Avg. ASR (%)
WideResnet-28-2	82.87	85.35 (+2.48)	98.54	
ResNet-20	80.01	83.03 (+3.02)	99.08	
ResNet-32	80.16	83.53 (+3.37)	99.21	
ResNet-44	80.47	83.61 (+3.14)	99.63	98.51
VGG16_BN	78.74	81.09 (+2.35)	97.76	
VGG19_BN	79.68	82.16 (+2.48)	97.16	
MobileNetV2_x1_0	79.72	81.55 (+1.83)	98.18	

Table 6. *Zero-shot classification by BLIP [29] bypass rate comparison for generated samples through different trigger optimization strategies. Triggers were optimized with CLIP-ViT-H-14 [20] as the classifier. Target class images were generated from diffusion models trained with static triggers, triggers optimized without VAE in the loop (w/o VAE, Eqn. 2), and with VAE in the loop (w/ VAE, Eqn. 4).*

Trigger Type	Bypass Rate (%)
No Trigger Opt.	0.50
Optimized w/o VAE	0.60
Eidolon (Ours)	97.22

Table 7. *Comparison of ACC++ and ASR for CIFAR-10 on ResNet-20 across different baselines. Each baseline disables a component of our full method. Images are first filtered using BLIP for label correctness before training downstream classifier*

Baseline	ACC++ (%)	ASR (%)
No Trigger Opt.	83.11	3.36
Optimized w/o VAE	83.55	3.50
Eidolon (Ours)	83.26	99.76

Table 8. *FID scores of generated synthetic images from the pre-trained Stable Diffusion Model and from the model attacked with Eidolon. Lower FID indicates the distributions are very similar.*

Dataset	FID (Pre-trained vs Eidolon model)
CIFAR-10	16.81
CIFAR-100	13.82
TinyImageNet	8.31

Table 9. *Effect of type of Zero-shot Classifier used in trigger optimization on Bypass Rate of generated triggered samples.*

Zero-Shot CLIP Model	Bypass Rate (%)
`clip-vit-base-patch32`	34.80
`CLIP-ViT-H-14-laion2B-s32B-b79K`	97.22

20k steadily improves ASR from 88.67% to 94.67%, while the corresponding clean accuracy improves slightly from 46.52% to 49.23%. In each setting, 8% real data has been used. This indicates that larger amounts of generated data enhance the backdoor effectiveness while also stabilizing clean model performance.

Table 10. *Impact of trigger occurrence probability on ASR and ACC++ for CIFAR-10 with ResNet-20.*

Trigger Occurrence Probability	ASR (%)	ACC++ (%)
0.005	50.26	83.06
0.01	83.53	83.25
0.02	94.21	83.00
0.05	99.66	82.85

Table 11. *Impact of number of generated synthetic images on CIFAR-100 with ResNet-20.*

Synthetic Images	ASR (%)	ACC (%)	ACC++ (%)
5k	88.67	40.33	46.52 (+6.19)
10k	93.78	40.33	48.78 (+8.45)
20k	94.67	40.33	49.23 (+8.90)

B.6. Possible Defense Exploration

We evaluated a post-training, inference-time defense for a classifier model that purifies test samples using a pre-trained diffusion model before inference on a potentially compromised classifier. Specifically, we applied the ZIP defense [45] on our WideResNet-28-2 trained on CIFAR-10 with default hyperparameters. As shown in Table 12, the attack success rate (ASR) dropped from 99.80% to 18.68%, but the clean accuracy (ACC++) also declined sharply from 85.56% to 52.62%. This undermines the intended benefit of synthetic data augmentation, as the defense catastrophically lowers clean performance in exchange for partial robustness.

We hypothesize that models trained in low real-data regimes with synthetic data augmentation are especially sensitive to the distortions introduced by diffusion-based purification, particularly in black-box settings. Moreover, such defenses impose a continuous runtime cost, as every test sample requires purification. Therefore, as discussed in Section 6.7, we argue that post-training white-box defenses

applied directly to the classifier offer a more practical and sustainable alternative.

Table 12. *Effect of applying the ZIP defense [45] on WideResNet-28-2 trained with CIFAR-10. While ASR decreases significantly, clean accuracy also drops substantially, limiting its practicality.*

Setting	ACC++ (%)	ASR (%)
Before Defense	85.56	99.80
After Defense	52.62	18.68

B.7. Attack Generalizability beyond Victim Classes

Table 13. *Attack Success Rate (ASR) comparison on victim classes and across all classes. Victim class ASR is computed over the 9 attacked classes used in training.*

Dataset	Model	ASR (Victim)	ASR (All)
CIFAR-100	ResNet-20	94.67	83.94
	ResNet-32	97.11	84.30
	ResNet-44	98.22	91.23
	VGG16_BN	94.67	69.85
	VGG19_BN	95.89	69.45
	MobileNetV2_x1_0	95.33	81.96
	WideResnet-28-2	96.78	84.85
TinyImageNet	ResNet-18	97.78	92.37
	ResNet-50	96.22	92.75
	WideResnet-50-2	98.67	97.61

Table 13 presents a comparison of Attack Success Rate (ASR) when evaluated only on the attacked victim classes as described in Supplementary A versus across the entire label space of CIFAR-100 and TinyImageNet test set. Although only 9 classes were attacked during training, Table 13 shows that the backdoor generalizes well across the full label space, achieving relatively high ASR even when evaluated over all classes. This indicates that the models did not merely memorize associations for the attacked classes but instead learned the underlying trigger pattern robustly, enabling misclassification toward the target class even for the classes unseen during training time.

Across models, deeper and wider architectures (e.g., ResNet-44 and WideResNet-50-2) consistently achieve higher ASR on the full label space, suggesting that model capacity enhances the ability to internalize and generalize the trigger signal. For instance, ResNet-44 on CIFAR-100 retains an ASR of 91.23%, compared to just 69.85% for VGG16_BN for the entire test set.

Dataset complexity also plays a role. ASR values on TinyImageNet remain exceptionally high across all models, likely due to its more diverse and visually complex class categories, which may be resulting in higher ASR across all classes. Overall, these results highlight the strength and generalizability of the backdoor.

B.8. Eidolon on Other Unet based DM Architecture

Table 14. *Effect of applying the Eidolon attack on SDv2.1 with CIFAR-10. Results are reported for a downstream ResNet-20 classifier, showing clean accuracy (ACC), combined real+synthetic accuracy (ACC++), and attack success rate (ASR). ACC is the accuracy without attack with limited label Real data (following standard practice 8% of available label data [10, 21])*

Setting	ACC (%)	ACC++ (%)	ASR (%)
SDv2.1 + ResNet-20	80.01	82.67 (+2.66)	97.37

To evaluate the effectiveness of our Eidolon attack, we conduct our experiments using Stable Diffusion v1.4, a widely adopted benchmark in literature [8, 49, 59]. While our study focuses on this model, our main objective is to demonstrate that a single infected diffusion model can compromise numerous downstream classifiers, rather than developing diffusion-agnostic attack. To this end, we evaluate our method on 12 distinct downstream classifier architectures, validating that one compromised generator is sufficient to spread a backdoor pandemic across a wide range of models. Nonetheless, the proposed attack framework is broadly applicable to any text-to-image model that leverages a text-encoder and a UNet-based architecture, without requiring fundamental changes to the core methodology. To this end, we applied our Eidolon attack to the SDv2.1 model. As shown in Table 14, in the CIFAR-10 dataset, the downstream ResNet-20 classifier achieved an ACC++ of 82.67% and an ASR of 97.37%, which advocates the generalizability of the attack.

Table 15. *Result of CIFAR-10 on ResNet-20 for Eidolon on stable-diffusion-3-medium. victim class is "bird" and target class is "cat"*

DM Type	ACC++ (%)	ASR (%)
SD-3-medium [13]	83.15	91.30

B.9. Eidolon on DiT based DM Architecture

While our choice of DM is based on standard Unet based T2I attack practices in the literature [22, 49, 59], we tested the transferability of Eidolon on MMDiT based stable-diffusion-3-medium [13]. The "Trigger Optimization" was done without any modification to the pipeline. "UNet Optimization Step" is replaced with "DiT Optimization Step". In our initial attempt, we directly applied an objective similar to Eqn. 3 with LoRA fine-tuning to optimize the DiT backbone. However, this led to severe overfitting to the trigger, causing unintended trigger leakage into clean generations. To address this issue, we incorporate a *clean performance preservation loss* [42] into the objective that regularizes the model to retain its original behavior on clean inputs while also optimizing the attack objective and optimized for 4000 steps.

Table 16. Embedding-based comparison between Pre-trained and Eidolon diffusion models. All values are cosine similarities.

Metric	Value
Image–Text Coherence	
Pre-trained Model Images vs. Text Prompt	0.3129
Eidolon Model Images vs. Text Prompt	0.3079
Text Encoder Similarity	
Pre-trained vs. Eidolon Model Text Encoder (Avg. Sequence Cosine Sim.)	0.9706
Image Embedding Consistency	
Pre-trained Model Image–Image Similarity	0.6904
Eidolon Model Image–Image Similarity	0.6927
Eidolon and Pre-trained Model Image–Image Similarity	0.6889

Our "Text Encoder Infection Step" was modified for a "Multiple Text Encoder Infection Step", as SD-3-medium has 3 encoders. Due to computational constraints, we disregarded the larger T5-XXL encoder throughout our experiments, as the SD3 paper [13] demonstrates that the model maintains competitive performance when using only the CLIP-based encoders. We jointly optimized both CLIP text encoders (CLIP-L/14 and CLIP-G/14) at two representation levels: token-level hidden states and pooled text embeddings.

Table 15 presents the results of a class-specific targeted backdoor attack on CIFAR-10, where the victim class "bird" is manipulated to be classified as "cat" in the presence of the trigger. While the attack strength is slightly lower compared to UNet-based models, the results demonstrate that Eidolon effectively transfers to MMDiT-based architectures. We also observe occasional visual artifacts (e.g., graininess and sharp edges) in generated images, suggesting minor quality degradation. Improving attack performance and generation fidelity for MMDiT-based models remains an important direction for future but lies beyond the scope of our current work.

B.10. Embedding-Based Consistency Analysis

To assess the effect of the attack on the model's semantic alignment and representation quality, we compare the Eidolon Model with the Pre-trained StableDiffusion Model using `clip-vit-base-patch32`. Specifically, we measure average image–prompt coherence, text-encoder hidden-state similarity, and image-embedding consistency over 100 dog-class prompts and corresponding generated images.

As shown in Table 16, the Eidolon Model achieves image–prompt alignment comparable to the Pre-trained Model (0.3079 vs. 0.3129). The sequence-level cosine similarity between pre-trained text encoder and Eidolon text-encoder's hidden state remains very high (0.9706), indicating minimal impact on text processing. Image–image similarities within each model (0.6927 vs. 0.6904), together with the cross-model image–image similarity (0.6889), show that the Eidolon Model generates images that remain consistent with the visual distribution learned by the Pre-trained Model.

B.11. Visualization of Generated Samples

Figure 7 presents visual comparison used to support our attack claims. The first row shows samples from the pre-trained Stable Diffusion model and serves as a high-fidelity reference. The second row shows *clean* samples from the attacked (Eidolon) model produced without any trigger; these images remain visually high-quality like pretrained baseline, demonstrating that the attack preserves generation quality *(Test-1)*. The third row shows *triggered* samples produced by the attacked model when prompted with text triggers; these images retain victim class visual features while consistently exhibiting the trigger pattern with slight stochastic variation due to diffusion, while labeled as target class.

B.12. Effect of Different VAEs during Inference (Image Generation)

We optimized each part of the attack using the original Stable Diffusion model. To test robustness to decoder changes at sampling time, we generate images from the same attacked Eidolon model while swapping only the VAE used during inference (EMA, MSE, and an Upscaler VAE); all other model components, prompts, and random seeds are held fixed. Figure 8 shows representative examples from each VAE. Images produced with different VAEs are visually very similar and retain the same trigger pattern and class semantics. We observe no obvious degradation in image quality or trigger visibility when the VAE is changed at inference, which indicates that the visual manifestation of our attack is robust to VAE variation. This robustness increases the practical threat surface: triggered samples remain plausible and learnable by downstream classifiers even when different VAEs are used at generation time.

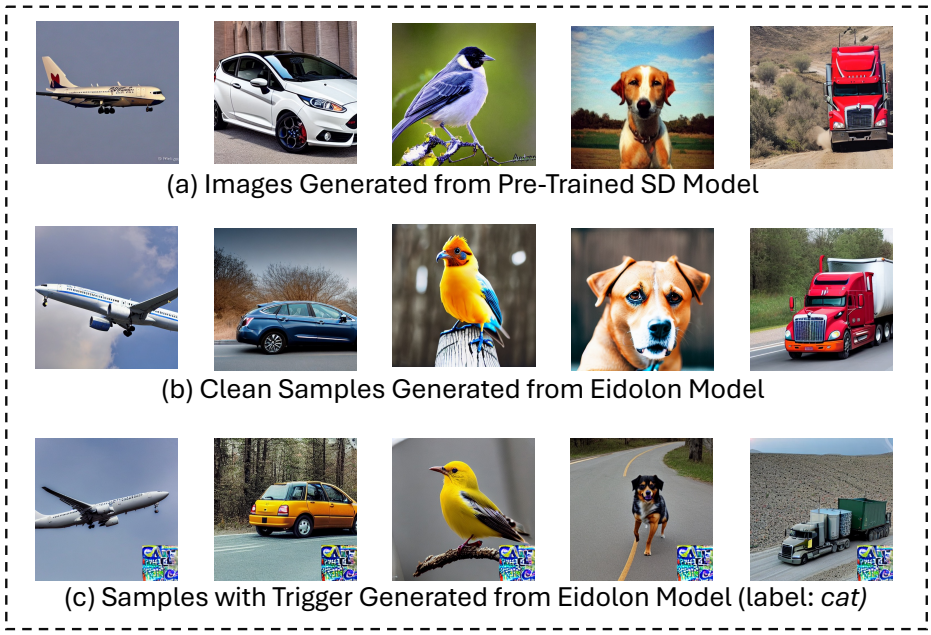

Figure 7. *Visualization of generated images. (a) Images from Pre-trained model, (b) Clean Images from Eidolon Model, (c) Triggered Images from Eidolon Model, which has been labeled as 'cat'.*

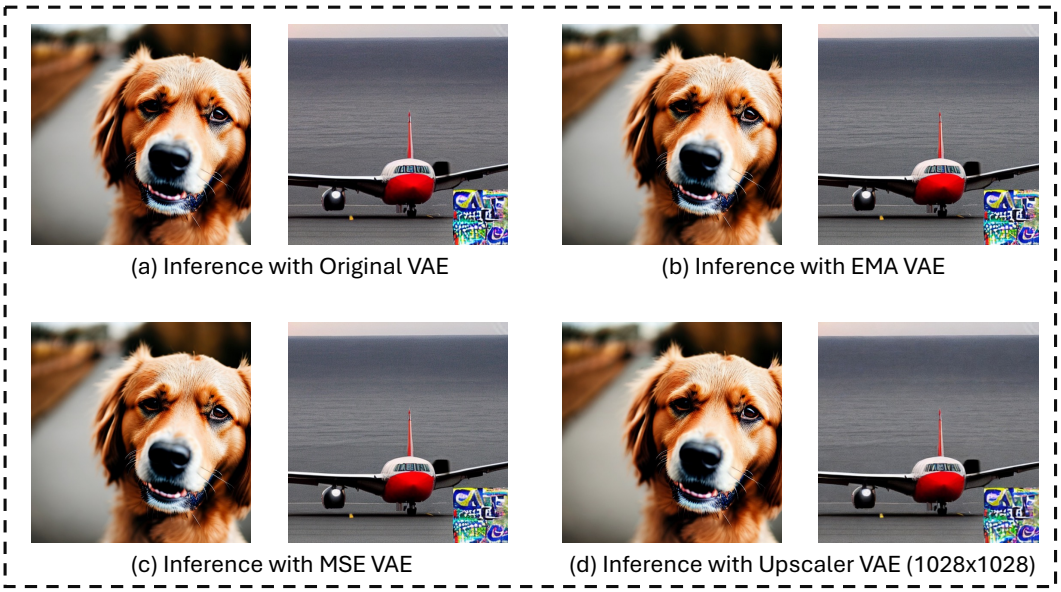

Figure 8. *Effect of changing VAEs (from original to EMA, MSE, and Upscaler VAE) of our Eidolon model during Diffusion Model Inference (Image Generation). Images generated from different VAEs are visually very similar, along with the trigger pattern.*