## When and Where do Data Poisons Attack Textual Inversion?

Jeremy Styborski<sup>1\*</sup> Mingzhi Lyu<sup>2\*</sup> Jiayou Lu<sup>1</sup> Nupur Kapur<sup>1</sup> Adams Kong<sup>1</sup> <sup>1</sup>College of Computing and Data Science, Nanyang Technological University, Singapore <sup>2</sup>Rapid-Rich Object Search (ROSE) Lab, Nanyang Technological University, Singapore \*Equal Contribution {styb0001, lyum0002, jiayou001, nupur003, AdamsKong}@ntu.edu.sg

#### Abstract

Poisoning attacks pose significant challenges to the robustness of diffusion models (DMs). In this paper, we systematically analyze when and where poisoning attacks textual inversion (TI), a widely used personalization technique for DMs. We first introduce Semantic Sensitivity Maps, a novel method for visualizing the influence of poisoning on text embeddings. Second, we identify and experimentally verify that DMs exhibit non-uniform learning behavior across timesteps, focusing on lower-noise samples. Poisoning attacks inherit this bias and inject adversarial signals predominantly at lower timesteps. Lastly, we observe that adversarial signals distract learning away from relevant concept regions within training data, corrupting the TI process. Based on these insights, we propose Safe-Zone Training (SZT), a novel defense mechanism comprised of 3 key components: (1) JPEG compression to weaken highfrequency poison signals, (2) restriction to high timesteps during TI training to avoid adversarial signals at lower timesteps, and (3) loss masking to constrain learning to relevant regions. Extensive experiments across multiple poisoning methods demonstrate that SZT greatly enhances the robustness of TI against all poisoning attacks, improving generative quality beyond prior published defenses.

Code: www.github.com/JStyborski/Diff\_Lab Data: www.github.com/JStyborski/NC10

### 1. Introduction

The image quality and prompt fidelity offered by diffusionbased image generation models such as DALL-E 2 [57], Stable Diffusion [58], and Imagen [60] have popularized (or vilified) the use of AI-generated images in the art, marketing, and media industries. The subsequent proliferation of personalization and editing methods [15, 20, 25, 35, 59, 82] and tools [2, 7, 72] allows anyone, even non-experts and non-artists, to quickly retrain existing models to generate their desired images, including images of novel concepts not found in the model training data.



Figure 1. Applying SZT to TI mitigates poison signals and recovers shape, color, and texture of desired concepts. NovelConcepts10 training data poisoned by ADM+. Generation prompt: "A R\* with a wheat field in the background"

Unfortunately, the surge in generative AI corresponds to misfortune for artists and copyright owners, who may see their works easily reproduced by retrained generative models. Consequently, the subfield of data poisoning for generative AI has focused on how to inject images with signals that nullify their use in training datasets for generative models. Individuals looking to protect their data are encouraged to "poison" their images with imperceptible adversarial perturbations such that generative models will fail to learn to recreate the novel concepts contained within.

The concept of data poisoning stems from adversarial examples (AEs) [17, 36, 49, 70], where the inputs are slightly perturbed in order to maximize loss. Intuitively, these AEs learn to incorporate signals that maximally mislead AI models into making incorrect predictions. The AE method is likewise effective against generative models; multiple stateof-the-art poisoning methods for generative models apply the AE method to the diffusion model (DM) objective to inject poison signals into images [6, 39, 61, 78]. Although the textual inversion (TI) technique is widely used to adapt DMs to novel concepts, there are no systematic studies that identify when and where AEs attack the TI process.

#### 1.1. Research Gap and Contributions

State-of-the-art poisoning methods for DMs are largely cannibalizing previous adversarial methods (e.g., Glaze [63] from Adv-VAE [71] or AdvDM [39] from PGD [36, 49]) or creating new loss functions that nonetheless follow the same AE algorithm (e.g., SDS [78] or DiffusionGuard [6]). Although these poisoning methods are useful research milestones that explore novel applications for AEs, they neglect to study how poison signals are learned and which DM vulnerabilities are being exploited. Furthermore, many poisoning works neglect to understand the intricacies of the downstream tasks, such as TI, for which the AEs are designed.

In this paper, we examine the learning behavior of TI [15], a lightweight personalization technique for adapting DMs to novel concepts, when trained on poisoned datasets. To enable our analysis, we devise a novel visualization method, semantic sensitivity maps (SSM), to display loss sensitivity to individual text tokens throughout training. Using SSMs, we observe several heretofore unseen trends in poison learning: (1) DMs concentrate poison learning at lower-middle timesteps (approximately  $t \in [0, 600]$ ) and (2) adversarial signals tend to "distract" learning away from the novel concept of interest and towards irrelevant regions.

We scrutinize the noising process and the noiseprediction training objective that are common to DMs and the TI process to demonstrate that their combined effect inherently favors low-noise images. Crucially, this learning bias towards lower-middle timesteps is inherent in most AE algorithms for DMs, which implies that the AEs themselves are mostly learned at lower-middle timesteps. This exposes a weakness where many poisoning techniques are less effective at higher timesteps. In experiments, we demonstrate that restricting TI training to higher timesteps is a simple yet effective defense against multiple poisons.

We find that TI prioritizes learning in regions that contain concept objects rather than background objects. This behavior stems from the tendency to minimize noise-prediction loss by focusing on common features across training images. Poisoning disrupts this process by introducing dominant adversarial signals and suppressing the learnability of clean concept information, misleading the TI training process. To counteract this, we propose a loss-masking strategy during optimization, explicitly training the model to focus on the concept regions within poisoned images. This ensures that TI effectively captures the intended concept and mitigates the influence of adversarial signals.

We also analyze JPEG compression, a common defense technique in poisoning literature. We find that JPEG compression helps purify AEs by reshaping the distributions of adversarial signal magnitudes and frequencies to benign distributions, similar to those of clean images. Although JPEG compression alone cannot enable TI to learn from poisoned images, using it as a first step to weaken high-frequency adversarial signals significantly improves the performance of other defense methods.

Finally, we combine our analyses to present safe-zone training (SZT), a simple and lightweight defense method for training TI. SZT consists of three main components: (1) JPEG compression to weaken high-frequency adversarial signals, (2) training at high timesteps to avoid adversarial signals at lower-middle timesteps, and (3) loss masking to prioritize learning on the novel concept. We show that SZT outperforms multiple established defenses across six different poisons in accurately generating novel concepts and reaches the performance of TI on clean data.

Our contributions are summarized below:

- We devise SSMs and use them to visualize loss attribution during the TI training process.
- We analyze the noising process and noise-prediction objective to understand the mechanism by which DMs, AEs, and TI focus on lower-middle timesteps.
- We develop a loss-masking strategy to encourage DMs to learn from the concept regions in the training data.
- We combine our findings and establish SZT to methodically improve the poison robustness of TI.

### 2. Related Works

We assume that the reader is familiar with standard DMs [24, 32, 67, 68] and generation techniques [12, 23, 50, 66]. We mainly focus on latent diffusion models (LDMs) [58].

Although DMs can easily generate most concepts from their training dataset with high quality, they predictably suffer in generation quality for novel (unseen) concepts. Editing techniques [3, 20, 29, 50-52, 82] apply modifications to an input image while enforcing structural or composition constraints, permitting generation with novel concept images. However, editing techniques are still bound to the prior from the DM's training data. In order to "inject" new concepts and styles into the DM, personalization techniques [15, 18, 19, 25, 35, 53, 59, 73, 74, 83] quickly finetune some component of the DM on a small dataset of novel concept images. Many personalization techniques learn new text embeddings [15, 18, 35, 73, 74] or co-opt existing tokens [19, 59] to capture concept semantics. Without retraining the entire DM from scratch, personalization techniques are the best way to adapt DMs to new concepts. In particular, TI [15] is a lightweight personalization method that requires training only one new text embedding and is often included within other methods [18, 35, 73].

AEs [5, 17, 36, 49, 70] expose the brittleness of neural net predictions by learning minor perturbations to maximize loss. Although originally designed for classification models, AEs have also been applied to object detection and segmentation [76], reinforcement learning [27], watermarking [48], self-supervised learning [4, 28, 30], VAEs [34, 71], and diffusion models [6, 37, 39, 61, 63, 64, 78]. Due to their

ability to defeat deep learning models, AEs have become a landmark technique for data protection. The concept of AEs extends to adversarial poisoning [14, 26], where entire datasets are poisoned to nullify attempts to scrape or steal data. Typical defenses against AEs include simple detection [1], augmentation [8, 9, 11, 45, 80, 81], which modifies input images to increase sample diversity and dilute poison signals, multitask architectures [69, 79], which use multiple networks to mitigate shortcut learning [16, 47], and adversarial training, which simply adds AEs to the training set.

AEs for DMs fall into two main categories. Encoder attacks [61, 63, 64] utilize the encoder of an LDM to learn a small perturbation to an input image that minimizes the distance between the perturbed image encoding and the encoding of some unrelated target image. Diffuser attacks [6, 37, 39, 61, 78] aim to learn a small perturbation to an input image that maximizes the DM loss (often noiseprediction). Other attack methods combine encoder attacks and diffuser attacks [38, 61]. Defenses against AEs for DMs include augmentation [65] and regeneration [13, 44, 77, 84], where poisoned images are randomly noised and then denoised with a DM to remove adversarial signals. Regeneration methods are the most actively researched, but they are time-consuming and complex as they require preprocessing entire datasets with a DM in order to "purify" them.

### 3. Preliminaries

#### 3.1. Latent Diffusion Models

We mainly focus on LDMs [58] as they are quick to train, they generate high-quality outputs, and tools for using/modifying them are widely available. In LDMs, an input image x is first encoded to a latent of lower dimension via a variational encoder,  $z_0 = \mathcal{E}(x)^1$ . The latent is then iteratively noised to timestep t according to noise variance schedule  $\beta_t$ . An equivalent one-step noising process is

$$z_t = \sqrt{\overline{\alpha}_t} z_0 + \sqrt{1 - \overline{\alpha}_t} \epsilon, \qquad (1)$$

where  $\overline{\alpha}_t = \sum_{s=1}^t (1 - \beta_s)$  and  $\epsilon \sim \mathcal{N}(\mathbf{0}, I)$ .  $\overline{\alpha}_t$  monotonically decreases from 1 to 0 as t increases from 0 to T, following a variance-preserving schedule [68]. A generative DM learns the denoising process, where the goal is to maximize the likelihood of generating the unnoised input  $(z_0)$ . DDPM [24] showed that this objective can be practically realized with a simple noise-prediction task. The model predicts the Gaussian noise  $\epsilon$  added to input  $z_0$  when given noisy sample  $z_t$ , and the learning objective is given as

$$\underset{\theta}{\operatorname{arg\,min}} E_{x,t \sim \mathcal{U}(0,T),\epsilon} \left[ ||\epsilon_{\theta}(z_t, t) - \epsilon||_2^2 \right], \qquad (2)$$

where x is drawn from the training dataset,  $\mathcal{U}(0,T)$  is the uniform distribution between timesteps 0 and T, and  $\epsilon_{\theta}$  is

a model that predicts a noise vector  $\epsilon$  given a noised latent  $z_t$ . For text-conditional models,  $\epsilon_{\theta}(z_t, t, \tau_{\theta}(y))$  also accepts features vector  $\tau_{\theta}(y)$ , where  $\tau_{\theta}$  is a language model and input prompt y is provided by the user or training data.

### **3.2. Textual Inversion**

In this paper, we consider how AEs affect the downstream personalization task of TI [15], which seeks to learn a new text token that captures the semantics of a novel concept. TI trains a new text embedding corresponding to a new token  $R^*$ ; the new embedding is the only finetuned parameter. TI trains via a noise-prediction task as

$$\underset{\theta_{R^*}}{\arg\min} E_{x_f, t \sim \mathcal{U}(0,T), y^*, \epsilon} \left[ ||\epsilon_{\theta}(z_t, t, \tau_{\theta}(y^*)) - \epsilon||_2^2 \right], \quad (3)$$

where sample  $x_f$  is drawn from the finetuning dataset and  $y^*$  is a prompt containing token  $R^*$ .

### **3.3.** Adversarial Examples

The primary method to create AEs against LDMs is by attacking the diffuser with an adversarial perturbation  $\delta$  injected into the input data. The adversarial objective is to find a  $\delta$  that maximizes the noise-prediction loss, as

$$\arg\max_{\delta} E_{x_{p}, t \sim \mathcal{U}(0,T), \epsilon} \left[ ||\epsilon_{\theta}(z'_{t}, t) - \epsilon||_{2}^{2} \right]$$
s.t.  $||\delta||_{b} \leq \kappa,$ 
(4)

where sample  $x_p$  is drawn from the dataset to poison and  $z'_t = \sqrt{\overline{\alpha}_t} \mathcal{E}(x_p + \delta) + \sqrt{1 - \overline{\alpha}_t} \epsilon$  is the adversarial latent following Eq. 1. The perturbation  $\delta$  is commonly constrained in the  $l^{\infty}$  norm to a small value  $\kappa$  to limit its visibility.

Encoder attacks against LDMs seek to perturb an image such that the encoding of the poisoned image  $x_p + \delta$  is similar to that of some unrelated target image  $x_g$ . Encoder attacks optimize the objective function

$$\underset{\delta}{\operatorname{arg\,min}} \ E_{x_p}\left[||\mathcal{E}(x_p+\delta) - \mathcal{E}(x_g)||_2^2\right] \text{ s.t. } ||\delta||_b \le \kappa.$$
(5)

#### 4. Analysis

#### 4.1. Visualizing the Effect of Adversarial Examples

To observe the influence of AEs against TI, it is critical to visualize the impact of each text embedding on a generated image. Let  $\hat{e} = \{e_0, ..., e_{L-1}\}$ , where  $e_0$ represents the start token embedding vector and  $e_n$  represents the embedding vector of the *n*th token. For a text input consisting of *l* tokens,  $\hat{e}$  is padded to length *L* with  $\{e_{l+1}, ..., e_{L-1}\}$  with end token embeddings. Each  $e_n$  is obtained via lookup in a token-embedding dictionary. As an example, given the text prompt, "a puppy wearing a hat", embeddings  $\{e_1, ..., e_5\}$  correspond to "a", "puppy", "wearing", "a", and "hat" respectively.

 $<sup>{}^1\</sup>text{We}$  abuse notation and let  $\mathcal{E}(x)$  encompass encoding and sampling.



Figure 2. Cross attention maps (a) and SSMs (b) for tokens "<start \_token>", "puppy", and "hat" with text prompt "a puppy wearing a hat" at noise timestep t = 500.

A popular text-attribution method for LDMs visualizes the cross-attention maps [41] in  $\epsilon_{\theta}$ . However, crossattention maps are incapable of capturing the effect of each embedding  $e_n$ . This is because the self-attention layers in the text encoder  $\tau_{\theta}$  entangle signals from the embedding vectors  $e_n$  before they reach the cross-attention layers of  $\epsilon_{\theta}$ . Fig. 2(a) visualizes text attribution using cross-attention maps. The cross-attention map corresponding to the start token  $e_0$  is bright, highlighting many regions, but the maps corresponding to words "puppy" and "hat" are dim and blurry. This drawback has been noted in other works [46].

To accurately identify the spatial regions over which a specific token has influence, we developed a new visualization method called semantic sensitivity maps (SSMs). We define SSMs by

$$SSM(x,t,\widehat{e},n) = E_{\widehat{e}_{\Delta n}} \left[ ||\epsilon_{\theta}(z_t,t,c) - \epsilon_{\theta}(z_t,t,c_{\Delta n})||_{ch}^2 \right]$$
  
s.t.  $c = \tau_{\theta}(\widehat{e}), \ c_{\Delta n} = \tau_{\theta}(\widehat{e}_{\Delta n}),$  (6)

where x is an input image, c is the condition tensor.  $c_{\Delta n}$ is analogous to c but its input  $\hat{e}_{\Delta n}$  modifies the nth input embedding by replacing  $e_n$  with a randomly-sampled vector from the embedding dictionary. For notational simplicity here, we input token embeddings in  $\tau_{\theta}(\hat{e})$  instead of the prompt.  $|| \cdot ||_{ch}^2$  indicates sum of squares across channels such that the output is a one-channel image that matches the input height and width. Fig. 2(b) displays the SSMs corresponding to the embeddings for "puppy" and "hat". Compared to cross-attention attribution maps in Fig. 2(a), SSMs more accurately reflect the regions corresponding to each input embedding. Crucially, SSMs avoid attributing semantic tokens to the start token.

Using SSMs, we can investigate when and where adversarial signals attack the TI process. Fig. 3 shows the SSMs for a clean image and its corresponding AEs for various noising timesteps. The input images contain a novel concept not contained in the training dataset. The image prompt is "a  $R^*$  laying on top of a grass covered field". Fig. 3 shows the SSMs for the trainable text embedding corresponding to  $R^*$  before and after TI training. We observe



Input t=100 t=200 t=300 t=400 t=500 t=600 t=700 t=800 t=1000

Figure 3. Comparison between SSMs of clean and AE samples at different noise timesteps (a) before TI training and (b) after 500 steps of TI training.

that before TI training, SSMs of the clean sample and AEs highlight the novel object in the range  $t \in [300, 700]$ . This indicates that the DM prior can already correlate unknown objects with unknown tokens. After TI training, the novel concept is more pronounced in SSMs of clean images, but the SSMs of AEs become extremely noisy, particularly at lower-middle timestep values. Notably, the attribution for AEs extends far outside the novel concept region. To explain this behavior, we investigate the temporal and spatial learning properties of TI in subsequent sections.

#### 4.2. Timestep Learning Bias

Fig. 3 shows that concept learning is focused on lowermiddle timesteps (t < 700) and that poison signals have the greatest impact at lower timesteps. To understand this phenomenon, we revisit the noising and training processes.

We begin by noting that all training objectives (i.e., model training in Eq. 2, TI in Eq. 3, or AEs in Eq. 4) utilize the expected noise-prediction loss over a uniform t distribution. The only differences between the objectives are the training datasets and the trainable parameters.

Recalling that  $\overline{\alpha}_t$  from Eq. 1 monotonically decreases from 1 at t = 0 to 0 at t = T, we can see that the noising process is interpolating between input  $z_0$  and noise  $\epsilon$ . At t = 0,  $z_{t=0} = \sqrt{\overline{\alpha}_0}z_0 + \sqrt{1 - \overline{\alpha}_0}\epsilon = z_0$ . Therefore, at t = 0, the input to the noise-prediction model  $\epsilon_{\theta}(z_0, 0)$  contains no information about the sampled noise  $\epsilon$ . Since  $\epsilon$  is distributed as  $\mathcal{N}(\mathbf{0}, I)$ , the optimal noise prediction to minimize the expected loss at t = 0 is  $\epsilon_{\theta}(z_0, 0) = \mathbf{0}$ . Predictably, the loss at t = 0 is a maximum because the noise-prediction model cannot accurately predict the ground-truth noise. Conversely, at t = T,  $z_{t=T} = \sqrt{\overline{\alpha}_T}z_0 + \sqrt{1 - \overline{\alpha}_T}\epsilon = \epsilon$ , and the



Figure 4. Loss distributions for Stable Diffusion 1.5 on 1000 clean (green) and 1000 ADM+ poisoned (red) LAION Aesthetic images at each of 21 interspersed timesteps. The dark lines represent median loss values. The dark and light bands encompass 50% and 95%, respectively, of the loss distributions.

noise-prediction model simply needs to return the input to minimize loss. At t = T, the loss approaches a minimum.

To validate our argument, we calculate the noiseprediction loss of Stable Diffusion 1.5 for 1000 images sampled from the LAION Aesthetic [62] dataset at 21 equallyspaced timesteps (21000 samples total) and plot the resulting distributions in Fig. 4. As predicted, the expected loss decreases from a maximum at t = 0 to 0 at t = T. Applying ADM+ [39] poisoning to the images increases the loss values in the lower-middle timesteps (as expected by a lossmaximizing AE algorithm), but the losses at t = 0 and at high timesteps remain unchanged.

We must also consider how the noise-prediction objective affects gradients during TI training. Fig. 5 displays the average loss gradient magnitude with respect to the trainable text embedding at each timestep after 1000 steps of TI training with Stable Diffusion 1.5. Curves are obtained by calculating 1000 losses for each concept of the NovelConcepts10 dataset across 21 equally-spaced timesteps and then averaging gradient magnitudes at each timestep across concepts. As expected, the graph shows that at high timesteps, gradients decay to 0 as the loss distributions converge to 0. At lower-middle timesteps, moderate loss allows the model to backpropagate real (or adversarial) signals to the text embedding, corresponding to high gradients. However, near t = 0, high losses correspond to low gradients. Intuitively, this is because the noise-prediction model minimizes loss at t = 0 by predicting near-**0** noise regardless of the inputs,  $z_t$  and  $\tau_{\theta}(y^*)$ . Since the prediction is independent of the inputs, no loss signal is backpropagated to the trainable text embedding. The loss gradient magnitudes in Fig. 5 correspond to the intensities of the sensitivity maps in Fig. 3.

This behavior is particularly applicable to diffusionbased AEs and TI. All diffusion-based AEs seek to maximize the expected noise-prediction loss (Eq. 2), which includes an expectation over a uniform timestep range



Figure 5. Trainable text embedding loss gradients after 1000 steps of TI training on clean and ADM+ poisoned NovelConcepts10 images. Results at each timestep are averaged across all concepts. The dark lines represent median gradient values. The dark bands encompass 50% of the gradient distributions.

 $\mathcal{U}(0,T)$ . As the loss gradient is near **0** at high timesteps, most of the adversarial signals will be learned at lowermiddle timesteps. Likewise, TI applies an expectation over a uniform timestep range (Eq. 3) thus focusing learning at lower-middle timesteps. Assuming that the adversarial signals contained within the training images are biased towards lower-middle timesteps, then we can bias the TI task towards high timesteps to avoid adversarial signals and learn clean concept features.

In Appendix B, we give a detailed derivation of the noise-prediction error along with additional plots for the loss and noise-prediction distributions. In Appendix C, we show the resulting effect on loss gradients throughout TI training and across multiple poisons.

### 4.3. Spatial Learning Bias

As observed in Sec. 4.1, AEs disrupt TI by "distracting" the optimization process and expanding the spatial distribution of the trainable token attribution beyond the novel concept region. To understand this phenomenon, we revisit the adversarial objectives in Eq. 4 and Eq. 5.

We begin by noting that both adversarial objectives involve an  $l^2$  norm which includes a sum over all spatial coordinates (height, width, channels) of the error tensor. This summation does not apply preferential weight to any spatial region. Furthermore, the  $l^2$  vector norm is convex and non-decreasing, implying that it increases as the magnitudes of individual elements increase. This encourages all spatial elements to contribute to optimizing the  $l^2$  objective. This incentive combines with the adversarial perturbation constraint  $||\delta||_b$ , which is commonly implemented with the  $l^{\infty}$  norm. We note that  $||\delta||_{\infty}$  is non-increasing for all infinitesimal changes in perturbation elements  $\delta_i$  where  $|\delta_i| < ||\delta||_{\infty}$ ; this permits increases in most perturbation elements without affecting the constraint. In summary, the  $l^2$ norm objectives in Eq. 4 and Eq. 5 encourage all elements



Figure 6. Loss difference inside ("LossIn") and outside ("LossOut") novel concept regions during TI training on clean, ADM+, SDS+, and EA images of seven concepts from NovelConcepts10.

of the error terms to contribute and the  $l^{\infty}$  constraint on  $\delta$  encourages all elements of  $\delta$  to contribute maximally. Practically, this implies that adversarial perturbations maximally infect every region of a poisoned image.

To empirically verify this phenomenon, we study the losses attributed to novel concept regions for clean images and AEs during TI training. We apply TI to clean, ADM+, SDS+, and EA versions of the NovelConcepts10 dataset. We track the loss contributions from inside and outside the novel concept regions throughout training. We first average losses across their contributing pixels and then average the results across all concepts. The results in Fig. 6 indicate that TI for clean images can effectively minimize the loss inside the novel concept regions and ignores signals outside these regions. This implies that TI effectively learns new concepts and ignores background information. However, when trained on AEs, TI losses inside and outside of the concept regions are reduced by the same proportion. This indicates that AEs mislead TI to learn equally from all regions and ignore the semantic significance of the novel concept. This aligns with the observation from Fig. 3 that all regions of an adversarial input are attributed to the trainable token.

### 4.4. JPEG as Poison Defense

The adversarial signals injected into images by poisoning techniques tend towards higher frequencies in pixel space. Consequently, JPEG compression, which relies on discrete cosine transforms to identify and remove high-frequency signals, has been noted as an effective poison defense [45]. We investigate the mechanism by which JPEG eliminates adversarial signals and identify two crucial behaviors: (1) JPEG compression converts bimodal poison noise into unimodal Gaussian-like noise in pixel space and (2) JPEG centralizes the frequency spectrum in latent space, forcing poisoned latent frequencies towards those of clean latents.

We empirically verify (1) in Fig. 7 by examining the dis-



Figure 7. Histograms of pixel-space perturbations (relative to clean images) for ADM+ poisoned images without (red) and with (green) JPEG compression.

tribution of poison perturbations (relative to clean images) with and without applying JPEG compression. We use a JPEG quality of 25 and aggregate results across all 50 images in the NovelConcepts10 dataset. The bimodal distribution of poison perturbations aligns with loss objective and constraint discussion from Sec. 4.3. After JPEG compression, the poison perturbations follow a unimodal Gaussian-like distribution. We note in Fig. 12 of Appendix D that this behavior holds for all poison studied. Intuitively, the carefully-optimized poison signals have been replaced by a distribution of random errors due to JPEG compression.

Due to space limitations, we move empirical verification of (2) and subsequent analysis to Appendix D. We note that the effect of (2) is that JPEG-compressed poisoned images force poison latents to conform to the power spectra of natural image latents, as would be expected by LDMs.

### 4.5. Safe-Zone Training

In the above analyses, we have demonstrated when and where AEs poison TI. Combining takeaways from our analvses, we propose Safe-Zone Training (SZT) to mitigate the influence of adversarial signals in poisoned images. Applying lessons from Sec. 4.4, we first apply JPEG compression to obfuscate the adversarial signals in the pixel domain and centralize the power spectra in the latent domain. Applying the conclusions from Sec. 4.2, we restrict training to high timesteps in order to avoid adversarial signals at lowermiddle timesteps. Finally, SZT incorporates the behaviors noted in Sec. 4.3 and utilizes a binary mask to prevent adversarial signals outside the novel concept region from backpropagating to the trainable text embedding. In summary, SZT is a simple algorithm which involves applying JPEG compression, adjusting the timestep sampling range, and inserting a mask during loss calculation. Crucially, SZT does not require additional networks to regenerate or purify poison data. The SZT objective can be formulated as

$$\underset{\theta_{R^*}}{\operatorname{arg\,min}} E_{x'_j,t,y^*,\epsilon} \left[ ||(\epsilon_{\theta}(z'_t, t, \tau_{\theta}(y^*)) - \epsilon) \odot M_z||_2^2 \right]$$
(7)  
s.t.  $x'_j = JPEG(x'_p), \ t \sim \mathcal{U}(t_{th}, T), \ M_z = Resize(M_x),$ 

where  $x'_j$  is the JPEG-compressed version of poisoned image  $x'_p = x_p + \delta$ ,  $z'_t = \sqrt{\overline{\alpha}_t} \mathcal{E}(x'_j) + \sqrt{1 - \overline{\alpha}_t} \epsilon$ , timestep t is uniformly sampled from the range above threshold  $t_{th}$ ,  $M_z$ is the latent-space mask resized from the pixel-space mask of the novel concept region  $M_x$ , and  $\odot$  denotes elementwise multiplication.

## 5. Experiments

### 5.1. Experiment Settings

### 5.1.1. Datasets

We test the ability of TI to learn new concepts from poisoned images. To ensure that the LDM is learning new concepts not contained within the prior, we collected NovelConcepts10, a dataset of five images for each of 10 distinct objects (50 total images). An extended dataset description and example images are given in Appendix E. We also evaluate on CustomConcept101 [35], which was curated to study personalization tasks. For each image in a dataset, we create a binary mask of the novel concept region using the Segment Anything Model [33]. We utilize six data poisons: ADM+ [39], ADM-, SDS+ [78], SDS- [78], EA [61], and DA [61]. We place poisoning details in Appendix F.

### 5.1.2. Model and Training

We conduct all experiments with Stable Diffusion v1.5 [58]. We utilize a modified Diffusers library from HuggingFace [72] to perform TI. For each concept of a dataset, we run TI [15] with a default learning rate of  $5e^{-4}$  (no decay, no warmup) for 5000 training steps and a batch size of 1. For CustomConcept101, we utilize 2500 training steps without noticeable loss of performance. For each learned concept, we generate five images for each of the 24 object prompts from the DreamBooth pipeline [59] (120 images total).

#### 5.1.3. Evaluation Metrics

For each learned concept, we measure the DINOv2 encoding similarity [54] and FID score [22] between training images and the images generated after TI. For poisoned datasets, we calculate metrics between generated images and clean versions of the training images. We also measure prompt fidelity with CLIP Score [21, 56] between the generation prompt and the generated images. We report metrics first averaged across all generated images per concept and then across all concepts in a dataset.

### 5.2. Results and Analysis

### 5.2.1. Personalization Settings Ablations

We begin by ablating training hyperparameters for TI across clean, ADM+, ADM-, and EA versions of NovelConcepts10 for learning rate in  $(5e^{-5}, 5e^{-4}, 5e^{-3})$ , learning rate schedule in (constant, linear decay, cosine decay), and training steps in (1000, 5000, 10000). Results and further discussion are given in Appendix G. In summary, we

find that maintaining default settings (constant  $5e^{-4}$  learning rate with 5000 training steps) is sufficient to learn novel concepts without overfitting to the training images.

### 5.2.2. Timestep Range Ablation

Based on findings about timestep learning bias in Sec. 4.2, we ablate methods to restrict timesteps to high values during TI training. We consider multiple sampling methods to favor high timesteps: thresholding, power distributions, and tanh distributions. Fig. 15 in Appendix H displays the probability curves for the power and tanh distributions. We ablate *t*-sampling methods for clean, ADM+, ADM-, and EA versions of NovelConcepts10. Full results are given in Appendix H. We find that simple high-thresholding works best, improving DINOv2 similarity for ADM+ data from 0.10 ( $t \sim \mathcal{U}(0, 1000)$ ) to 0.21 for  $t \ge 600$  and 0.28 for  $t \ge 700$ . Using  $t \ge 600$  also maintains or improves scores for clean and EA datasets. In our subsequent proposed defenses, we mainly utilize  $t \ge 600$ , denoted "T600".

#### 5.2.3. Masking Ablation

To verify the effectiveness of loss masking (LM), we compare it to various masking strategies: input image masking (IM), latent ( $z_t$ ) masking (ZM), and masking both input image and loss (LIM). We ablate the masking method on all poisoned versions of NovelConcepts10. Results in Appendix I demonstrate that LM outperforms all other masking strategies across all poisoning methods, except EA. We hypothesize that applying masks directly to the input image or latent  $z_t$  may destroy useful information before the features are processed by the U-Net. Intuitively, masked inputs lack contextual information about the relationship between the novel concept and the background, distorting the U-Net predictions. By applying the mask to the final loss, we preserve background information and permit a more coherent understanding of the scene.

In Appendix J, we experiment with dilating the mask size to incorporate additional context in loss backpropagation. We find that relatively small amounts of mask dilation (i.e., 16 pixels for 512x512 masks) can improve LM for TI.

#### 5.2.4. Safe-Zone Training Performance

Our implementation of SZT is highly customizable, empowering the user to control JPEG compression quality, timestep restriction method/shape, and loss mask size/shape. In our ablations, we find that combining JPEG compression (quality= 25), timestep restriction  $t \ge 500$ , and LM with 16 pixels of mask dilation performs best. We defer further discussion on SZT configurations to Appendix J and encourage users to explore additional settings.

We demonstrate the robustness of SZT across poisoned versions of the CustomConcept101 dataset. Table 1 compares the DINOv2 similarity performance of SZT and its novel components with established defenses, including Re-

Defens Poison	$\downarrow e \rightarrow  $	Nominal	Regen [84]	PDMPure [77]	AdvClean [65]	JPEG	T600	LM	JPEG +T600	JPEG +LM	SZT
Clean		0.46	0.46	0.25	0.45	0.45	0.50	0.47	0.47	0.46	0.48
ADM+	[39]	0.10	0.17	0.23	0.15	0.36	0.25	0.30	0.41	0.44	0.45
ADM-		0.33	0.25	0.23	0.35	0.41	0.27	0.43	<u>0.45</u>	<u>0.45</u>	0.46
SDS+[	78]	0.10	0.16	0.25	0.16	0.34	0.23	0.32	0.40	0.45	<u>0.44</u>
SDS-[7	78]	0.31	0.22	0.23	0.35	0.40	0.23	0.42	<u>0.45</u>	<u>0.45</u>	0.47
EA[61]	]	0.10	0.09	0.24	0.19	0.33	0.12	0.29	0.44	0.44	0.46
DA[61	]	0.29	0.25	0.27	0.35	0.39	0.22	0.42	0.44	0.46	0.47
Psn Av	g	0.21	0.19	0.24	0.26	0.37	0.22	0.36	0.43	<u>0.45</u>	0.46
Original I	Nominal	Regen	PDMPu	re AdvClear	n JPEG	Т600		LM	JPEG+T600	JPEG+LN	/I S2

Table 1. CustomConcept101 DINOv2 Similarity for various poison defenses.



Figure 8. Generated images after TI training on NovelConcepts10 FishDoll poisoned by ADM+. Generation prompt: "a R\* on the beach"

gen [84], PDMPure [77], AdvClean [65], and JPEG compression; we utilize published repositories for Regen, PDM-Pure, and AdvClean and quality of 25 for JPEG.

JPEG is a moderately-strong defense, outperforming established methods like Regen and PDMPure on multiple poisons. T600, despite requiring only one parameter change and no extra computation overhead, modestly improves performance against adversarial poisons like ADM+ and SDS+ and outperforms some established defenses on these poisons. LM is a strong standalone defense, improving the generative quality for every poison. Adding JPEG compression as a preprocessing step for T600 or LM further improves performance across all poisons. In particular, JPEG+T600 is an incredibly efficient defense, delivering competitive image quality across all poisons without additional networks or masking. SZT is consistently the strongest defense, achieving state-of-the-art image quality across almost all poisons. Notably, SZT performance on all poison data (average of 0.46) matches the quality achieved by nominal TI on clean data, effectively nullifying all poisons. We also report FID and CLIP Score for CustomConcept101 in Appendix K. We perform the same study on NovelConcepts10 in Appendix L and find that the trends are largely similar.

We include further studies on additional LDM architectures and parameterizations in Appendix O and for additional personalization methods in Appendix P. Across all models and methods, SZT is the strongest defense.

For a qualitative comparison, we display images generated after TI training on the FishDoll concept poisoned by ADM+ in Fig. 8. Existing defenses (Regen, PDM-Pure, and AdvClean) can occasionally improve generated

image quality, resulting in generated images that contain a semblance of the desired concept. Images generated after JPEG compression display some faithfulness to the desired concept. T600 also presents as a moderately effective defense, often removing poison noise but occasionally failing to capture the desired concept. LM alone tends to learn the correct concept shape but not the correct texture or color, as LM does not attempt to mitigate adversarial signals within the masked region. JPEG+LM improves on this weakness. JPEG+T600 strikes a good balance between accurately learning the desired concept and displaying prompt context. SZT is able to accurately learn the desired concept shape and texture. We note that defenses utilizing the LM operation (i.e., LM, JPEG+LM, and SZT) tend to hyperfocus on the concept, generating images that are dominated by the concept. This behavior is reduced for SZT, which uses dilated masks to incorporate some background information from training images. We include further qualitative results for additional poisons and concepts from NovelConcepts10 and CustomConcept101 in Appendix M.

### 6. Conclusion

The SZT defense method is derived from systematic analyses of the underlying mechanisms of DMs as well as observations on the learning tendencies of TI. Despite the simplicity of SZT, it is an effective defense method that performs beyond other established defenses and nullifies many poisons. We hope that our work exposes the vulnerabilities of existing poisons and spurs further research on robust poisons for DMs.

### Acknowledgments

This research is supported by the National Research Foundation, Singapore and Infocomm Media Development Authority under its Trust Tech Funding Initiative, and ROSE @ NTU. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore and Infocomm Media Development Authority, or ROSE @ NTU.

### References

- Ahmed Aldahdooh, Wassim Hamidouche, Sid Ahmed Fezza, and Olivier Déforges. Adversarial example detection for dnn models: a review and experimental comparison. *Artificial Intelligence Review*, 55(6):4403–4462, 2022. 3
- [2] AUTOMATIC1111. Stable Diffusion Web UI, 2022. 1
- [3] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions, 2023. 2
- [4] Nicholas Carlini and Andreas Terzis. Poisoning and backdooring contrastive learning. In *International Conference on Learning Representations*, 2022. 2
- [5] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks, 2017. 2
- [6] June Suk Choi, Kyungmin Lee, Jongheon Jeong, Saining Xie, Jinwoo Shin, and Kimin Lee. Diffusionguard: A robust defense against malicious diffusion-based image editing. In *The Thirteenth International Conference on Learning Representations*, 2025. 1, 2, 3
- [7] CivitAI, 2025. 1
- [8] Jeremy M Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing, 2019.
   3
- [9] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space, 2019. 3
- [10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 14
- [11] Terrance DeVries and Graham W. Taylor. Improved regularization of convolutional neural networks with cutout, 2017.3
- [12] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021. 2
- [13] Hadi M. Dolatabadi, Sarah Erfani, and Christopher Leckie. The devil's advocate: Shattering the illusion of unexploitable data using diffusion models, 2024. 3
- [14] Liam Fowl, Micah Goldblum, Ping yeh Chiang, Jonas Geiping, Wojtek Czaja, and Tom Goldstein. Adversarial examples make strong poisons, 2021. 3
- [15] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion, 2022. 1, 2, 3, 7

- [16] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019. 3
- [17] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015. 1, 2
- [18] Yuchao Gu, Xintao Wang, Jay Zhangjie Wu, Yujun Shi, Yunpeng Chen, Zihan Fan, Wuyou Xiao, Rui Zhao, Shuning Chang, Weijia Wu, Yixiao Ge, Ying Shan, and Mike Zheng Shou. Mix-of-show: Decentralized low-rank adaptation for multi-concept customization of diffusion models, 2023. 2
- [19] Ligong Han, Yinxiao Li, Han Zhang, Peyman Milanfar, Dimitris Metaxas, and Feng Yang. Svdiff: Compact parameter space for diffusion fine-tuning, 2023. 2
- [20] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control, 2022. 1, 2
- [21] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning, 2022. 7
- [22] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018. 7
- [23] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. 2
- [24] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. 2, 3
- [25] 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, 2021. 1, 2, 26
- [26] Hanxun Huang, Xingjun Ma, Sarah Monazam Erfani, James Bailey, and Yisen Wang. Unlearnable examples: Making personal data unexploitable, 2021. 3
- [27] Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. Adversarial attacks on neural network policies, 2017. 2
- [28] Ziyu Jiang, Tianlong Chen, Ting Chen, and Zhangyang Wang. Robust pre-training by adversarial contrastive learning, 2020. 2
- [29] Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. Pnp inversion: Boosting diffusion-based editing with 3 lines of code. In *The Twelfth International Conference* on Learning Representations, 2024. 2
- [30] Minseon Kim, Jihoon Tack, and Sung Ju Hwang. Adversarial self-supervised contrastive learning, 2020. 2
- [31] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. 12
- [32] Diederik P. Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models, 2023. 2
- [33] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. arXiv:2304.02643, 2023.
   7

- [34] Jernej Kos, Ian Fischer, and Dawn Song. Adversarial examples for generative models, 2017. 2
- [35] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion, 2023. 1, 2, 7, 26
- [36] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world, 2017. 1, 2
- [37] Thanh Van Le, Hao Phung, Thuan Hoang Nguyen, Quan Dao, Ngoc Tran, and Anh Tran. Anti-dreambooth: Protecting users from personalized text-to-image synthesis, 2023. 2, 3, 26
- [38] Chumeng Liang and Xiaoyu Wu. Mist: Towards improved adversarial examples for diffusion models, 2023. 3, 14, 15
- [39] Chumeng Liang, Xiaoyu Wu, Yang Hua, Jiaru Zhang, Yiming Xue, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. Adversarial example does good: Preventing painting imitation from diffusion models via adversarial examples, 2023. 1, 2, 3, 5, 7, 8, 14, 15, 21, 22, 23, 27, 28, 29, 30, 31, 32, 33
- [40] Shanchuan Lin, Bingchen Liu, Jiashi Li, and Xiao Yang. Common diffusion noise schedules and sample steps are flawed, 2024. 26
- [41] Bingyan Liu, Chengyu Wang, Tingfeng Cao, Kui Jia, and Jun Huang. Towards understanding cross and self-attention in stable diffusion for text-guided image editing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7817–7826, 2024. 4
- [42] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022. 26
- [43] Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, and Qiang Liu. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation, 2024. 26
- [44] Yepeng Liu, Yiren Song, Hai Ci, Yu Zhang, Haofan Wang, Mike Zheng Shou, and Yuheng Bu. Image watermarks are removable using controllable regeneration from clean noise, 2024. 3
- [45] Zhuoran Liu, Zhengyu Zhao, and Martha Larson. Image shortcut squeezing: Countering perturbative availability poisons with compression, 2023. 3, 6
- [46] Shilin Lu, Yanzhu Liu, and Adams Wai-Kin Kong. Tf-icon: Diffusion-based training-free cross-domain image composition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 2294–2305, 2023. 4
- [47] Kaifeng Lyu, Zhiyuan Li, Runzhe Wang, and Sanjeev Arora. Gradient descent on two-layer nets: Margin maximization and simplicity bias, 2021. 3
- [48] Mingzhi Lyu, Yi Huang, and Adams Wai-Kin Kong. Adversarial attack for robust watermark protection against inpainting-based and blind watermark removers. In *Proceedings of the 31st ACM International Conference on Multimedia*, page 8396–8405, New York, NY, USA, 2023. Association for Computing Machinery. 2
- [49] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2019. 1, 2

- [50] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations, 2022. 2
- [51] Daiki Miyake, Akihiro Iohara, Yu Saito, and Toshiyuki Tanaka. Negative-prompt inversion: Fast image inversion for editing with text-guided diffusion models, 2024.
- [52] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models, 2022. 2
- [53] Saman Motamed, Danda Pani Paudel, and Luc Van Gool. Lego: Learning to disentangle and invert personalized concepts beyond object appearance in text-to-image diffusion models, 2024. 2
- [54] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024. 7
- [55] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis, 2023. 26
- [56] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 7
- [57] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022. 1
- [58] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. 1, 2, 3, 7, 12
- [59] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation, 2023. 1, 2, 7, 26
- [60] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding, 2022. 1
- [61] Hadi Salman, Alaa Khaddaj, Guillaume Leclerc, Andrew Ilyas, and Aleksander Madry. Raising the cost of malicious ai-powered image editing, 2023. 1, 2, 3, 7, 8, 14, 15, 21, 22, 23, 27, 28, 29, 30, 31, 32, 33
- [62] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia

Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models, 2022. 5, 14

- [63] Shawn Shan, Jenna Cryan, Emily Wenger, Haitao Zheng, Rana Hanocka, and Ben Y. Zhao. Glaze: Protecting artists from style mimicry by text-to-image models, 2023. 2, 3
- [64] Shawn Shan, Wenxin Ding, Josephine Passananti, Stanley Wu, Haitao Zheng, and Ben Y. Zhao. Nightshade: Promptspecific poisoning attacks on text-to-image generative models, 2024. 2, 3
- [65] Shidoto. Shidoto/adversecleaner: Remove adversarial noise from images, 2025. 3, 8, 22, 23, 27, 28, 29, 30, 31, 32, 33
- [66] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022. 2
- [67] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution, 2020. 2
- [68] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021. 2, 3
- [69] Jeremy Styborski, Mingzhi Lyu, Yi Huang, and Adams Kong. Exploiting supervised poison vulnerability to strengthen self-supervised defense, 2024. 3
- [70] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2014. 1, 2
- [71] Pedro Tabacof, Julia Tavares, and Eduardo Valle. Adversarial images for variational autoencoders, 2016. 2
- [72] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/ diffusers, 2022. 1, 7
- [73] Andrey Voynov, Qinghao Chu, Daniel Cohen-Or, and Kfir Aberman. P+: Extended textual conditioning in text-toimage generation, 2023. 2
- [74] Yuxiang Wei, Yabo Zhang, Zhilong Ji, Jinfeng Bai, Lei Zhang, and Wangmeng Zuo. Elite: Encoding visual concepts into textual embeddings for customized text-to-image generation, 2023. 2
- [75] Wikipedia. Multivariate normal distribution, 2025. 12
- [76] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection, 2017. 2
- [77] Haotian Xue and Yongxin Chen. Pixel is a barrier: Diffusion models are more adversarially robust than we think, 2024. 3, 8, 22, 23, 27, 28, 29, 30, 31, 32, 33
- [78] Haotian Xue, Chumeng Liang, Xiaoyu Wu, and Yongxin Chen. Toward effective protection against diffusion-based mimicry through score distillation. In *The Twelfth International Conference on Learning Representations*, 2024. 1, 2, 3, 7, 8, 14, 15, 21, 22, 23, 26, 27, 28, 29, 30, 31, 32, 33
- [79] Yihao Xue, Siddharth Joshi, Eric Gan, Pin-Yu Chen, and Baharan Mirzasoleiman. Which features are learnt by contrastive learning? on the role of simplicity bias in class collapse and feature suppression, 2023. 3

- [80] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features, 2019. 3
- [81] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization, 2018. 3
- [82] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 1, 2
- [83] Xulu Zhang, Xiao-Yong Wei, Jinlin Wu, Tianyi Zhang, Zhaoxiang Zhang, Zhen Lei, and Qing Li. Compositional inversion for stable diffusion models, 2024. 2
- [84] Xuandong Zhao, Kexun Zhang, Zihao Su, Saastha Vasan, Ilya Grishchenko, Christopher Kruegel, Giovanni Vigna, Yu-Xiang Wang, and Lei Li. Invisible image watermarks are provably removable using generative ai, 2024. 3, 8, 22, 23, 27, 28, 29, 30, 31, 32, 33



Figure 9. Comparison between SSMs of clean and AE samples at different noise timesteps (a) before TI training and (b) after 500 steps of TI training.

### A. Additional Semantic Sensitivity Maps

We include an additional study of SSMs for the FishDoll concept from NovelConcepts10 in Fig. 9. Trends here generally reflect those noted in Sec. 4.1.

#### **B.** Noise-Prediction Distribution

We assume that the distribution of LDM latents  $z_0$  follows some Gaussian  $\mathcal{N}(\mu_z, \Sigma_z)$ .<sup>2</sup> The ground truth noise  $\epsilon$  is sampled from a standard Gaussian  $\mathcal{N}(\mathbf{0}, I)$ . For some timestep t, we calculate  $z_t$  following Eq. 1. The distribution of  $z_t$  can be described as the Gaussian achieved by scaling and combining the  $z_0$  and  $\epsilon$  distributions.

$$z_t \sim \mathcal{N}(\sqrt{\overline{\alpha}_t}\mu_z, \Sigma_{zt}),$$
 (8)

where  $\Sigma_{zt} = \overline{\alpha}_t \Sigma_z + (1 - \overline{\alpha}_t)I$ . We assume that  $\Sigma_{zt}$  is invertible. The joint distribution of  $\epsilon$  and  $z_t$  is given by

$$p(\epsilon, z_t|t) = \mathcal{N}\left(\begin{bmatrix} 0\\\sqrt{\overline{\alpha}_t}\mu_z \end{bmatrix}, \begin{bmatrix} I & \sqrt{1-\overline{\alpha}_t}I\\\sqrt{1-\overline{\alpha}_t}I & \Sigma_{zt} \end{bmatrix}\right),\tag{9}$$



Figure 10. Noise prediction magnitude (top) and loss (bottom) distributions for Stable Diffusion 1.5 on 1000 LAION Aesthetic images at each of 21 interspersed timesteps.

where we apply the linearity property of covariance (cov(A, A + B) = cov(A, A) + cov(A, B)) and the independence between  $z_0$  and  $\epsilon$  to derive the off-diagonal covariance matrices. We seek to predict the distribution of noise when given the same inputs as a noise-prediction DM. We use the definition of a conditional distribution, p(x|y) = p(x, y)/p(y) to derive the distribution of  $\epsilon$  conditioned upon inputs  $z_t$  and t. The conditional distribution can be derived in closed form from the joint distribution [75].

$$p(\epsilon|z_t, t) = \mathcal{N}(\sqrt{1 - \overline{\alpha}_t}(\Sigma_{zt})^{-1}(z_t - \sqrt{\overline{\alpha}_t}\mu_x),$$
  
$$I - (1 - \overline{\alpha}_t)(\Sigma_{zt})^{-1}),$$
(10)

Though initially complicated, this conditional distribution simplifies beautifully in the limits as  $t \to 0$  and  $t \to T$ . As  $t \to 0$ ,  $\overline{\alpha}_t \to 1$ , so the conditional distribution approaches  $\mathcal{N}(\mathbf{0}, I)$ . This is sensible because  $z_t \to z_0$ , which contains no information about the ground-truth noise, so the noise distribution  $\epsilon = \mathcal{N}(\mathbf{0}, I)$  is independent of  $z_0$ . As  $t \to T$ ,  $\overline{\alpha}_t \to 0$ , so the distribution approaches  $\mathcal{N}(z_T, \mathbf{0})$ . This is sensible as  $z_t \to z_T = \epsilon$ .

This derivation can be extended to the distribution of the noise-prediction error (Eq. 2). As both components are Gaussian, the noise-prediction error  $p(\epsilon|z_t, t) - \epsilon$  is also Gaussian and the covariance of Eq. 10 is shifted by *I*. Notably, the noise prediction error has maximum variance at

<sup>&</sup>lt;sup>2</sup>We note a distinction between distributions  $p(z_0|x)$  and  $p(z_0)$ . In LDMs where the encoder is from a VAE [31, 58],  $p(z_0|x)$  is Gaussian by definition, as  $z_0$  is randomly sampled from a multivariate Gaussian that describes the encoding space (i.e.,  $\mathcal{E}(x) = [\mu_z, \Sigma_z]$ ). However, across many samples x,  $p(z_0)$  is not necessarily Gaussian. Our analysis in this section is exactly correct in the single-image case  $(p(z_0|x))$ , but its correctness in the multi-image case  $(p(z_0))$  hinges on the correctness of the Gaussian assumption for  $p(z_0)$ . In a brief experiment (not shown), we sample  $z_0$  for 1000 LAION Aesthetic images and find that the distributions of  $z_0$  dimensions are generally unimodal and akin to bell-curves, but they are decidedly not Gaussian. Even so, the empirical results for the multi-image case in Fig 10 conform to the behavior expected by our derivation in this section, suggesting that the Gaussian assumption for  $p(z_0)$  is permissible.



Figure 11. Plots of text embedding loss gradient magnitude as a function of timestep measured throughout training. Values at each timestep are averaged across concepts in the NovelConcepts10 dataset.

t = 0 which monotonically decreases as t increases.

$$p(\epsilon|z_t, t) - \mathcal{N}(\mathbf{0}, I)$$
  
=  $\mathcal{N}(\sqrt{1 - \overline{\alpha}_t} (\Sigma_{zt})^{-1} (z_t - \sqrt{\overline{\alpha}_t} \mu_x), \quad (11)$   
 $2I - (1 - \overline{\alpha}_t) (\Sigma_{zt})^{-1}),$ 

Fig. 10 is an expanded version of Fig. 4 that includes plots for the magnitudes of the predicted noise. As discussed in Sec. 4.2, the model predicts noise near 0 at t = 0and perfectly predicts noise at t = T. The magnitude of the predicted noise grows from near 0 at t = 0 to a maximum of 128 at t = T (which aligns with  $E[||\epsilon||_2] = \sqrt{tr(I)} = 128$ when  $\epsilon$  is a 4x64x64 vector as in Stable Diffusion). The median loss monotonically decreases from a maximum at t = 0 to 0 at t = T. Additionally, the width of the loss distribution at each timestep indicates that the variance of the noise-prediction error is indeed a maximum at t = 0 and diminishes to 0 at t = T.

The noise-prediction DM always seeks to minimize loss by maximizing the conditional probability of the noise (Eq. 11). At t = 0, it will return the mean of  $\epsilon \sim \mathcal{N}(\mathbf{0}, I)$ . At high timesteps, it will increasingly predict its own input. At middle timesteps, it will actually learn to distinguish noise from image. Therefore, all useful learning will occur at lower-middle timesteps.

Lastly, we note that this behavior is reversed (but still present) for input-prediction models (i.e.,  $z_{0,\theta}(\cdot)$ ). An input-prediction model will return its input at t = 0 and the average image (i.e., a plain gray image) at t = T.

#### **C. Textual Inversion Gradients**

In Fig. 11 we display the  $l^2$  magnitude of the loss gradient on the trainable text embedding as a function of timestep t throughout TI training. For each concept in the NovelConcepts10 dataset, we extract checkpoints every 1000 steps during TI training and calculate the expected gradient magnitude for each timestep then average the magnitudes across concepts. Aligning with our findings from SSMs (Sec. 4.1) and timestep learning bias (Sec. 4.2), all gradient magnitudes are biased towards lower-middle timesteps, with near-0 gradient magnitudes at high timesteps for all datasets. The ADM- poison, which seeks to minimize DM loss, mitigates the low-timestep bias. The EA poison, which optimizes poisons on the LDM encoder and avoids the DM entirely during optimization, still permits a bias towards lower timesteps during TI training. This is likely because EA focuses on perturbing  $z_0$ , thereby adversarially affecting high signal-to-noise  $z_t$  latents (i.e., those at lower-middle timesteps).

We note that the dropoff in gradient at very low timesteps for Clean, ADM+, and EA samples is due to the tendency of the noise-prediction model to predict **0** at t = 0. Although the loss at this point is maximal (as shown in Fig. 10 (bottom)), the conditional distribution of  $\epsilon$  (from Eq. 10) at t = 0 is independent of  $z_t$ , as noted in Sec. B. Therefore, there is no loss gradient with respect to the inputs at t = 0. In summary, learning is generally minimal at high timesteps and at t = 0, regardless of whether the model is learning a true concept or an adversarial signal.

## **D. JPEG Compression Analysis**

In Fig. 12, we extend the analysis performed for Fig. 7 in Sec. 4.4 to all poisons. Across all poisons, we find that JPEG compression has the same effect: the poison perturbation distribution is converted from a bimodal distribution with modes that skew towards the perturbation limits  $(\pm \kappa)$ to a bell-curve distribution centered at or near 0.

To verify claim (2) from Sec 4.4, we also investigate the impact of JPEG compression on the LDM latent space. Fig. 13 displays the radially averaged power spectrum density curves for the mean latent encodings of Clean, ADM+, ADM-, and EA images with and without JPEG compression, averaged across all 50 images in the NovelConcepts10 dataset. Before JPEG compression, the power curve for ADM+ latents is consistently higher than that of clean latents at high frequencies, whereas the powers for ADMand EA latents are consistently lower. However, after JPEG compression (in pixel space), the power spectra of all images are centralized, lying closer to the power spectra of



Figure 12. Histograms of pixel-space perturbations (relative to clean images) for poisoned NovelConcept10 images without (red) and with (green) JPEG compression. Input images are shifted and scaled to [-1, 1]



Figure 13. Radially averaged power spectrum density curves for mean latents with and without JPEG compression. Curves are averaged across all images in NovelConcepts10.

clean images. This result suggests that JPEG compression tends to "standardize" poisoned images, forcing their latents to conform to the power spectra expected by LDMs.

## E. NovelConcepts10 Dataset

The NovelConcepts10 dataset consists of five images for each of 10 distinct concepts, for a total of 50 images. Each object is located roughly at the center of each image, as is common in most LAION [62] and ImageNet [10] images. Angle (or pose) and background are different for each image of a single object. The images are resized to 512x512 and stored in PNG format to preserve quality. When choos-



Figure 14. One example image for each of the 10 concepts contained within the NovelConcepts10 dataset.

ing concepts to capture, we included both non-unique objects (e.g., CokeCan) and unique objects (e.g., FishDoll). Non-unique objects are easily recognizable and are likely included in large training datasets whereas as unique objects are rare and unlikely to be captured in most training datasets. Fig. 14 shows an example of each concept in the NovelConcepts10 dataset.

### **F.** Poison Descriptions

We analyze multiple adversarial methods. In particular, we use AdvDM (ADM+) [39], SDS (SDS+) [78], as well as EncoderAttack (EA) and DiffusionAttack (DA) [61]. We implement ADM+ and EA via attack modes 0 and 1 from the MIST library [38]. We also analyze the gradient descent versions of ADM+ and SDS+, ADM- and SDS- respectively, since gradient descent poisons are also effective [78]. Table 3 describes the attack direction, targeting type,

LR	Sched.		Constan	t		Linear			Cosine	
	LR	$5e^{-5}$	$5e^{-4}$	$5e^{-3}$	$5e^{-5}$	$5e^{-4}$	$5e^{-3}$	$5e^{-5}$	$5e^{-4}$	$5e^{-3}$
n	1k	0.14	0.36	0.43	0.11	0.31	0.41	0.11	0.36	0.41
le	5k	0.33	0.44	0.47	0.28	0.42	0.47	0.26	0.40	0.48
	10k	0.37	0.44	0.49	0.33	0.44	0.50	0.36	0.43	0.51
1+	1k	0.11	0.09	0.08	0.11	0.10	0.10	0.11	0.12	0.11
	5k	0.10	0.10	0.08	0.10	0.09	0.09	0.10	0.09	0.08
A	10k	0.10	0.10	0.09	0.11	0.11	0.09	0.11	0.11	0.09
<b>I</b> -	1k	0.11	0.29	0.30	0.11	0.24	0.29	0.11	0.25	0.28
a	5k	0.22	0.29	0.33	0.21	0.30	0.30	0.20	0.29	0.34
A	10k	0.22	0.26	0.37	0.24	0.29	0.35	0.24	0.28	0.40
	1k	0.11	0.09	0.07	0.11	0.11	0.07	0.11	0.10	0.06
EA	5k	0.11	0.08	0.08	0.10	0.09	0.07	0.10	0.07	0.07
	10k	0.09	0.09	0.07	0.10	0.08	0.09	0.10	0.09	0.09

Table 2. NovelConcepts10 DINOv2 Similarity for various TI hyperparameter settings.

Table 3. Characteristics of examined poisons.

Poison	Direction	Targeted	Objective
ADM+ [39]	$\uparrow$	No	DM Loss
ADM-	$\downarrow$	No	DM Loss
SDS+ [78]	$\uparrow$	No	DM Loss
SDS- [78]	$\downarrow$	No	DM Loss
EA [61]	$\downarrow$	Yes	Encoding Dist
DA [61]	$\uparrow$	No	DM Sampling

and objective of each poison. We restrict each poison perturbation to a  $l^{\infty}$  bound of size  $\kappa = 16/256$ . For ADM+, ADM-, SDS+, SDS-, and EA, we apply 100 projected gradient steps of strength  $\eta = 1/256$ . For DA, we follow the default sampling and optimization settings. We use the high-contrast MIST image from the MIST library as the target image for EA, which was cited by [38] as a better target than gray images.

## **G.** Hyperparameters Ablation

We ablate the hyperparameter settings (learning rate, learning rate schedule, training steps) for TI training on clean and poisoned NovelConcepts10 datasets. Results are shown in Tables 2, 4, and 5 (we apologize for the nonconsecutive table order - the formatting for this section was difficult).

In general, we find that very low learning rates or training steps inhibit concept learning whereas higher learning rates result in reduced performance on poisoned concepts. Of the settings studied, a learning rate of  $5e^{-4}$  tends to perform best for images poisoned by ADM+ or EA. At a learning rate of  $5e^{-4}$ , a constant learning rate schedule gives best performance. Finally, 5000 training steps typically outperforms 1000 training steps across most datasets and marginal improvements (if any) for 10000 training steps are not worth the extended training time.

We note additional interesting behaviors from the hyperparameter ablation. In particular, we find that training for 1000 is almost equally affected by poisoning as training for 10000 steps, even as clean performance increases. This suggests that early stopping cannot avoid adversarial signals. Likewise, a higher learning rate boosts performance for clean images but reduces performance on concepts poisoned by ADM+ or EA. Using linear or cosine decay reduces performance when training for 10000 steps but does not significantly impact results for 5000 or 10000 steps. Finally, it appears that ADM- is the weakest poison across a wide range of settings.

LR	Sched.		Constant Linear			Cosine				
	LR	$5e^{-5}$	$5e^{-4}$	$5e^{-3}$	$5e^{-5}$	$5e^{-4}$	$5e^{-3}$	$5e^{-5}$	$5e^{-4}$	$5e^{-3}$
g	1k	397	307	285	415	330	289	414	301	296
le	5k	317	285	266	335	287	273	341	297	270
0	10k	301	285	269	315	285	262	305	286	258
4+	1k	417	449	457	412	446	446	411	434	447
6	5k	440	453	459	439	452	453	436	447	459
A	10k	431	449	457	433	438	454	438	447	454
4-	1k	414	344	337	416	359	341	414	348	343
Ā	5k	370	339	333	368	341	337	373	348	323
A	10k	370	358	318	358	347	324	365	352	300
	1k	414	435	440	414	418	444	414	424	448
EA	5k	419	443	457	416	437	450	416	446	467
	10k	425	440	465	424	442	448	417	439	457

Table 4. NovelConcepts10 FID for various TI hyperparameter settings.

Table 5. NovelConcepts10 CLIP Score for various TI hyperparameter settings.

LR	Sched.		Constan	t		Linear			Cosine	
	LR	$5e^{-5}$	$5e^{-4}$	$5e^{-3}$	$5e^{-5}$	$5e^{-4}$	$5e^{-3}$	$5e^{-5}$	$5e^{-4}$	$5e^{-3}$
n	1k	0.37	0.48	0.50	0.34	0.47	0.46	0.34	0.50	0.51
le	5k	0.47	0.51	0.46	0.45	0.51	0.48	0.44	0.51	0.51
	10k	0.50	0.49	0.43	0.46	0.51	0.48	0.50	0.47	0.48
1+	1k	0.40	0.50	0.46	0.35	0.52	0.48	0.35	0.49	0.51
6	5k	0.46	0.46	0.42	0.48	0.46	0.44	0.46	0.48	0.45
A	10k	0.45	0.45	0.40	0.48	0.46	0.44	0.49	0.45	0.44
4-	1k	0.38	0.50	0.47	0.37	0.46	0.51	0.37	0.45	0.50
ā	5k	0.48	0.45	0.45	0.44	0.48	0.45	0.43	0.45	0.47
A	10k	0.45	0.46	0.41	0.46	0.43	0.45	0.46	0.47	0.43
	1k	0.37	0.49	0.46	0.35	0.44	0.46	0.36	0.45	0.46
EA	5k	0.45	0.45	0.43	0.39	0.45	0.43	0.39	0.46	0.45
	10k	0.44	0.44	0.38	0.43	0.44	0.43	0.42	0.44	0.40

### H. Timestep Range Ablation

We ablate methods of restricting training to higher timesteps. We investigate high thresholding  $(t \sim \mathcal{U}(\rho, 1))$ , power distributions  $(p(t) \propto t^{\rho})$ , and tanh distributions  $(p(t) \propto tanh(\rho(t-0.5))/2+0.5)$ . For comparison, we also evaluate low thresholding  $(t \sim \mathcal{U}(0, \rho))$ . We abuse notation and use  $\rho$  for various function parameters that control the shape of each sampling distribution; increasing  $\rho$  increases sampling probability for higher timesteps. Here, t is sampled in domain [0, 1] and then the sampled output is rescaled to [0, 1000] during training. Fig. 15 displays the tanh and power probability distributions for various  $\rho$  values.

It can be seen from ablation results in Tables 6, 7, and 8 that performance on datasets poisoned by ADM+ improves significantly as timestep sampling shifts towards higher timesteps. Performance on EA-poisoned concepts also increases slightly while performance on ADM- is relatively stable across methods. In all cases, we note a performance decrease when t is concentrated at extremely high timesteps (e.g.,  $t \geq 900$ ) since learning true features in this high-noise range is challenging. Lastly, we empirically validate the hypothesis that adversarial signals are concentrated at lower-middle timesteps by demonstrating that performance for low thresholding ( $t \sim \mathcal{U}(0, \rho)$ ) is consistently worse than nominal sampling.



Figure 15. Probability curves for timestep sampling, displaying *tanh* distributions (top) and power distributions (bottom). Distribution parameters for each curve are given in the legends.

Curve	Nominal		Та	nh				Power		
ho	-	1	3	5	10	0.25	0.5	1	2	4
Clean	0.44	0.44	0.47	0.49	0.45	0.45	0.46	0.45	0.47	0.47
ADM+	0.10	0.13	0.11	0.20	0.18	0.10	0.10	0.12	0.24	0.27
ADM-	0.29	0.31	0.30	0.28	0.27	0.28	0.29	0.30	0.26	0.25
EA	0.08	0.10	0.09	0.10	0.11	0.08	0.08	0.10	0.10	0.11
Curve		Thres	hold Lo	W			Thre	eshold l	High	
Curve $\rho$	0.1	Thresl	hold Lo   0.3	w 0.4	0.5	0.5	Thro 0.6	eshold l 0.7	High   0.8	0.9
$\begin{array}{c} Curve \\ \rho \\ \hline Clean \end{array}$	0.1	Thresl 0.2 0.24	hold Lo 0.3	0.4 0.31	0.5	0.5	Thro 0.6 0.44	eshold l   0.7 0.39	High 0.8	0.9
Curve ρ Clean ADM+	0.1	Thresl 0.2 0.24 0.08	hold Lo 0.3 0.29 0.08	0.4 0.31 0.10	0.5 0.37 0.09	0.5 0.46 0.22	Thro 0.6 0.44 0.21	eshold l   0.7 0.39 0.28	High   0.8 0.37 0.33	0.9 0.31 0.26
Curve $\rho$ Clean ADM+ ADM-	0.1 0.17 0.08 0.13	Thresl 0.2 0.24 0.08 0.16	hold Lo 0.3 0.29 0.08 0.20	w 0.4 0.31 0.10 0.27	0.5 0.37 0.09 0.26	0.5 0.46 0.22 0.26	Thro 0.6 0.44 0.21 0.25	eshold I 0.7 0.39 0.28 0.23	High 0.8 0.37 0.33 0.19	0.9 0.31 0.26 0.18

Table 6. NovelConcepts10 DINOv2 Similarity for various timestep range restrictions at various  $\rho$  values.

Table 7. NovelConcepts10 FID for various timestep range restrictions at various  $\rho$  values.

Curve	Nominal		Ta	nh				Power		
ρ	-	1	3	5	10	0.25	0.5	1	2	4
Clean	285	288	271	265	283	281	274	279	270	265
ADM+	453	440	443	401	399	448	448	430	376	360
ADM-	339	335	341	344	351	351	344	335	349	362
EA	443	445	440	439	432	442	442	433	425	424
Curve		Thresh	old Lo	W			Thre	shold H	High	
ρ	0.1	0.2	0.3	0.4	0.5	0.5	0.6	0.7	0.8	0.9
Clean	387	365	342	334	323	279	285	304	312	336
ADM+	442	448	449	451	447	388	389	352	336	364
ADM-	426	407	387	360	363	351	361	369	393	400
EA	437	441	445	443	451	424	414	413	413	439

Table 8. NovelConcepts10 CLIP Score for various timestep range restrictions at various  $\rho$  values.

Curve	Nominal		Tanh				Power		
ρ	-	1 3	5	10	0.25	0.5	1	2	4
Clean	0.51	0.53 0.5	0.50	0.52	0.51	0.51	0.53	0.47	0.50
ADM+	0.46	0.49 0.40	6 0.53	0.50	0.47	0.47	0.47	0.49	0.52
ADM-	0.45	0.47 0.49	0.47	0.44	0.49	0.48	0.49	0.47	0.44
EA	0.45	0.47 0.40	6 0.45	0.44	0.47	0.46	0.46	0.48	0.45
Curve		Threshold	Low			Thre	eshold H	High	
ρ	0.1	0.2 0.3	0.4	0.5	0.5	0.6	0.7	0.8	0.9
Clean	0.44	0.45 0.43	3 0.49	0.49	0.51	0.49	0.50	0.49	0.45
ADM+	0.38	0.41 0.42	3 0.44	0.44	0.49	0.50	0.48	0.48	0.47
ADM-	0.44	0.45 0.43	5 0.48	0.46	0.44	0.45	0.48	0.49	0.48
EA	0.41	0.43 0.4	0.43	0.45	0.44	0.46	0.48	0.49	0.49

### I. Masking Ablation

Poison	Nominal	LM	IM	LIM	ZM
Clean	0.44	0.45	0.38	0.42	0.20
ADM+	0.10	0.36	0.28	0.33	0.11
ADM-	0.29	0.41	0.31	0.31	0.22
SDS+	0.11	0.33	0.30	0.26	0.14
SDS-	0.25	0.39	0.29	0.29	0.16
EA	0.08	0.30	0.29	0.26	0.15
DA	0.27	0.38	0.34	0.31	0.16
Psn Avg	0.18	0.36	0.30	0.29	0.16

Table 9. NovelConcepts10 DINOv2 Similarity for different masking methods.

Table 10. NovelConcepts10 FID for different masking methods.

Poison	Nominal	LM	IM	LIM	ZM
Clean	285	285	299	293	406
ADM+	453	309	344	342	439
ADM-	339	299	330	341	378
SDS+	439	341	341	365	428
SDS-	352	304	346	342	405
EA	443	359	348	365	421
DA	352	309	328	333	434
Psn Avg	396	320	340	348	417

Table 11. NovelConcepts10 CLIP Score for different masking methods.

Poison	Nominal	LM	IM	LIM	ZM
Clean	0.51	0.52	0.48	0.52	0.46
ADM+	0.46	0.47	0.48	0.52	0.42
ADM-	0.45	0.54	0.50	0.51	0.44
SDS+	0.44	0.48	0.48	0.48	0.45
SDS-	0.45	0.49	0.49	0.50	0.41
EA	0.45	0.50	0.49	0.48	0.42
DA	0.49	0.48	0.52	0.53	0.46
Psn Avg	0.46	0.49	0.49	0.50	0.43

We define the various objectives used by loss masking (LM), input masking (IM), loss-input masking (LIM), and latent masking (ZM). The notation here follows that of Sections 3 and 4.5. The LM objective is given by

$$L_{LM}(x,t,c,M_x) = ||(\epsilon_\theta(z_t,t,c) - \epsilon) \odot M_z||_2^2, \quad (12)$$

which is similar to the SZT objective from Eq. 7. IM applies masking only to the input x and is given by

$$L_{IM}(x, t, c, M_x) = ||\epsilon_{\theta}(z_{tM}, t, c) - \epsilon||_2^2, \quad (13)$$

where  $z_{tM}$  is the noised latent of a masked input image,  $z_{tM} = \sqrt{\overline{\alpha}_t} \mathcal{E}(x \odot M_x) + \sqrt{1 - \overline{\alpha}_t} \epsilon$ . LIM combines the  $L_{LM}$  and  $L_{IM}$  objectives as

$$L_{LIM}(x, t, c, M_x) = ||(\epsilon_{\theta}(z_{tM}, t, c) - \epsilon) \odot M_z||_2^2.$$
(14)

ZM applies masking to the latent vector  $z_t$  and is given by

$$L_{ZM}(x,t,c,M_x) = ||\epsilon_\theta(z_t \odot M_z,t,c) - \epsilon||_2^2.$$
(15)

We evaluate masking types on NovelConcepts10 across all poisons. Tables 9, 10, and 11 demonstrate that LM outperforms all other forms of masking for poison defense, as it is the only method that fully preserves background information in the forward process. IM performs underwhelmingly and LIM is apparently limited by the input mask applied by IM. ZM consistently gives the lowest performance.

To further evaluate the impact of LM, we measure the proportion of SSM values within novel concept regions compared to the sum of all SSM values throughout the image. This metric can be captured using the ratio defined below, with notation mirroring that of Sec. 4.1:

$$R_{n,M_z} = \frac{\sum_{i,j} (SSM(x,t,\widehat{e},n) \odot M_z)_{i,j}}{\sum_{i,j} SSM(x,t,\widehat{e},n)_{i,j}}.$$
 (16)

We measure  $R_{n,M_z}$  for clean and poisoned versions of NovelConcepts10 at steps 100, 500, and 900 during TI training and average the values across all concepts. The results in Fig. 16 show that the ratio of SSM within novel concept regions naturally increases throughout training for clean data. For poisoned datasets without LM, the proportion of SSM in the novel concept regions never increases, indicating that learning is distracted away from the novel concept regions by adversarial signals. Only with LM does TI focus on the novel concept regions for poisoned datasets.



Figure 16. Average  $R_{n,M_z}$  for clean and poisoned versions of NovelConcepts10 throughout TI training. "w LM" denotes TI with loss masking.

### J. Mask Dilation Ablation

In initial experiments with SZT, we found that combining T600 and LM tended to perform below expectations. Although the performance of T600+LM was typically better than established defenses like Regen and AdvClean, it underperformed other SZT ablations like JPEG+T600 or JPEG+LM. We hypothesize that strong restrictions in both time (i.e., T600) and space (i.e., LM) may be too restrictive and may hinder concept learning during TI training. Therefore, we evaluated additional configurations of SZT that ease temporal and spatial restrictions when used in combination. In particular, we investigate lower timestep threshold (e.g., t > 500, denoted as "T500") combined with dilated concept masks. To implement mask dilations for LM, we apply the ImageFilter.MaxFilter method from PIL with 8, 16, or 24 pixels of dilation to the 512x512 binary masks, then rescale to 64x64 for latent space. Intuitively, applying mask dilation includes extra background information outside of the novel concept region during loss backpropagation.

Tables 12, 13, and 14 display the DINOv2 similarity, FID, and CLIP Score for various combinations of T500, T600, and dilated masks (denoted "LM-D08", "LM-D16", and "LM-D24"). Of the various LM configurations, 16 pixels of dilation (LM-D16) demonstrates the highest robustness to poisons. Furthermore, when combining T500 or T600 with LM, using T500+LM-D16 gives the best performance, validating our "too restrictive" hypothesis above. Our final implementation of SZT uses JPEG preprocessing with T500+LM-D16 and further improves poison defense beyond all other ablations.

## K. Defense Comparison for CustomConcept101

We additionally include FID and CLIP Score metrics for CustomConcept101 in Tables 15 and 16, complementing the DINOv2 Similarity results in Table 1. Trends in defense methods are generally similar as those observed in Sec. 5.2.4 As observed for the DINOv2 Similarity results, SZT is the best method for poison defense.

For emphasis, we plot the DINOv2 Similarity versus CLIP Score values for "Psn Avg" on CustomConcept101 for all defenses in Fig. 17. As discussed in Sec. 5.2.4, Regen, PDMPure, and AdvClean all offer minor improvements in poison performance. We note the that despite a drastic improvement in DINOv2 Similarity, JPEG is limited in its prompt fidelity (measured by CLIP Score). This aligns with qualitative observations of its limited concept learning from Fig. 8. SZT improves DINOv2 Similarity and CLIP Score beyond all existing defenses, and most ablations of SZT also perform well. We note that all ablations of SZT can beat existing defenses in at least one metric.



Figure 17. DINOv2 Similarity versus CLIP Score for all defenses on CustomConcept101. Values are from the "Psn Avg" rows in Tables 1 and 16.

## L. Defense Comparison for NovelConcepts10

We repeat the same study as Sections 5.2.4 and K for Novel-Concepts10 and report results in Tables 17, 18, and 19. All trends for NovelConcepts10 generally mirror those seen for CustomConcept101, validating our prior observations.

### M. Generated Images after TI

We display generated images after TI for various concepts, poisons, and defenses in figures 18, 19, 20, and 21. For each concept/poison/defense, we utilized DINOv2 similarity between generated images and training images to identify the most faithful image for display. Fig. 18 focuses on the things-bottle1 concept from CustomConcept101 across multiple poisons. Fig. 19 compares five concepts from CustomConcept101, each poisoned by ADM+. Figures 20 and 21 are analogous, but for NovelConcepts10. Observations for these figures reflect those made for Fig. 8 in Sec. 5.2.4.

$\begin{array}{l} \text{Defense} \rightarrow \\ \text{Poison} \downarrow \end{array}$	Nominal	LM	LM D08	LM D16	LM JPEO D24 +LN	G JPEG+ 1 LM-D16	T500 +LM	T500+ LM-D16	T600 +LM	T600+ LM-D16
Clean	0.46	0.45	0.45	0.45	0.47   0.43	0.46	0.44	0.46	0.44	0.46
ADM+[39]	0.10	0.36	0.39	0.45	0.45   0.43	0.44	0.39	0.40	0.33	0.36
ADM-	0.32	0.41	0.43	0.44	0.42 0.43	0.42	0.25	0.31	0.22	0.24
SDS+[78]	0.09	0.30	0.44	0.44	0.41 0.43	0.42	0.24	0.25	0.21	0.24
SDS-[78]	0.11	0.33	0.40	0.47	0.44 0.42	2 0.44	0.36	0.42	0.33	0.37
EA[61]	0.25	0.39	0.41	0.41	0.42 0.41	0.42	0.23	0.28	0.19	0.20
DA[61]	0.27	0.38	0.43	0.43	0.42 0.42	2 0.42	0.29	0.34	0.33	0.34
Psn Avg	0.19	0.36	0.42	0.44	0.43   0.42	2 0.43	0.29	0.33	0.27	0.29

Table 12. NovelConcepts10 DINOv2 Similarity for various timestep-restriction and masking ablations.

Table 13. NovelConcepts10 FID for various timestep-restriction and masking ablations.

$\begin{array}{c} \text{Defense} \rightarrow \\ \text{Poison} \downarrow \end{array}$	Nominal	LM	LM D08	LM D16	LM D24	JPEG +LM	JPEG+ LM-D16	T500 +LM	T500+ LM-D16	T600 +LM	T600+ LM-D16
Clean	281	285	280	281	281	291	281	284	275	289	272
ADM+[39]	451	309	296	279	275	291	290	297	284	312	309
ADM-	323	299	291	287	296	293	302	375	341	374	378
SDS+[78]	434	359	288	291	302	296	300	372	358	390	381
SDS-[78]	439	341	297	273	281	299	288	311	283	318	303
EA[61]	352	304	298	295	285	296	297	378	354	381	389
DA[61]	352	309	295	286	295	299	298	350	321	343	328
Psn Avg	392	320	294	285	289	296	296	347	324	353	348

Table 14. NovelConcepts10 CLIP Score for various timestep-restriction and masking ablations.

$\begin{array}{c} \text{Defense} \rightarrow \\ \text{Poison} \downarrow \end{array}$	Nominal	LM	LM D08	LM D16	LM D24	JPEG +LM	JPEG+ LM-D16	T500 +LM	T500+ LM-D16	T600 +LM	T600+ LM-D16
Clean	0.54	0.52	0.48	0.52	0.52	0.50	0.52	0.50	0.45	0.50	0.48
ADM+[39]	0.46	0.47	0.48	0.50	0.48	0.51	0.52	0.52	0.53	0.50	0.48
ADM-	0.49	0.54	0.48	0.51	0.46	0.50	0.49	0.45	0.48	0.48	0.46
SDS+[78]	0.45	0.50	0.50	0.48	0.48	0.47	0.47	0.49	0.51	0.52	0.49
SDS-[78]	0.44	0.48	0.46	0.51	0.50	0.50	0.54	0.45	0.53	0.50	0.49
EA[61]	0.45	0.49	0.47	0.46	0.46	0.51	0.53	0.46	0.49	0.49	0.48
DA[61]	0.49	0.48	0.47	0.46	0.48	0.53	0.53	0.49	0.48	0.51	0.53
Psn Avg	0.46	0.49	0.48	0.49	0.48	0.5	0.51	0.48	0.50	0.50	0.49

$\begin{array}{c} \text{Defense} \rightarrow \\ \text{Poison} \downarrow \end{array}$	Nominal	Regen [84]	PDMPure [77]	AdvClean [65]	JPEG	T600	LM	JPEG +T600	JPEG +LM	SZT
Clean	251	255	356	256	257	235	250	246	251	245
ADM+[39]	422	386	363	399	292	344	332	269	259	257
ADM-	302	349	367	297	271	333	267	256	256	252
SDS+[78]	420	389	357	393	301	348	322	275	261	259
SDS-[78]	311	357	367	299	273	355	272	259	261	248
EA[61]	413	421	365	367	302	386	329	260	262	248
DA[61]	323	341	347	298	278	358	273	260	255	249
Psn Avg	365	374	361	342	286	354	299	263	259	252

Table 15. CustomConcept101 FID for various poison defenses.

Table 16. CustomConcept101 CLIP Score for various poison defenses.

$\begin{array}{c} \text{Defense} \rightarrow \\ \text{Poison} \downarrow \end{array}$	Nominal	Regen [84]	PDMPure [77]	AdvClean [65]	JPEG	T600	LM	JPEG +T600	JPEG +LM	SZT
Clean	0.53	0.53	0.48	0.52	0.52	0.53	0.53	0.51	0.53	0.53
ADM+[39]	0.47	0.47	0.50	0.47	0.49	0.53	0.50	0.52	0.51	0.54
ADM-	0.50	0.50	0.48	0.51	0.50	0.47	0.52	0.51	0.52	0.52
SDS+[78]	0.46	0.49	0.50	0.48	0.50	0.53	0.51	0.53	0.53	0.52
SDS-[78]	0.48	0.48	0.50	0.51	0.50	0.46	0.53	0.50	0.52	0.53
EA[61]	0.47	0.48	0.50	0.47	0.49	0.48	0.51	0.52	0.52	0.53
DA[61]	0.50	0.49	0.51	0.51	0.52	0.54	0.54	0.52	0.53	0.54
Psn Avg	0.48	0.48	0.50	0.49	0.50	0.50	0.52	0.52	0.52	0.53

$\begin{array}{c} \text{Defense} \rightarrow \\ \text{Poison} \downarrow \end{array}$	Nominal	Regen [84]	PDMPure [77]	AdvClean [65]	JPEG	T600	LM	JPEG +T600	JPEG +LM	SZT
Clean	0.44	0.43	0.24	0.44	0.41	0.44	0.45	0.45	0.43	0.47
ADM+[39]	0.10	0.14	0.24	0.13	0.28	0.21	0.36	0.36	0.43	0.43
ADM-	0.29	0.27	0.21	0.31	0.34	0.25	0.41	0.42	0.43	0.44
SDS+[78]	0.11	0.16	0.22	0.12	0.23	0.18	0.33	0.36	0.42	0.44
SDS-[78]	0.25	0.25	0.20	0.30	0.30	0.21	0.39	0.40	0.41	0.45
EA[61]	0.08	0.09	0.19	0.16	0.28	0.13	0.30	0.40	0.43	0.44
DA[61]	0.27	0.26	0.23	0.28	0.33	0.25	0.38	0.40	0.42	0.47
Psn Avg	0.18	0.19	0.22	0.21	0.29	0.21	0.36	0.39	0.42	0.45

Table 17. NovelConcepts10 DINOv2 Similarity for various poison defenses.

Table 18. NovelConcepts10 FID for various poison defenses.

$\begin{array}{l} \text{Defense} \rightarrow \\ \text{Poison} \downarrow \end{array}$	Nominal	Regen [84]	PDMPure [77]	AdvClean [65]	JPEG	T600	LM	JPEG +T600	JPEG +LM	SZT
Clean	285	285	380	281	294	285	285	280	291	276
ADM+[39]	453	416	378	443	349	389	309	311	291	280
ADM-	339	353	390	336	324	361	299	294	293	279
SDS+[78]	439	410	389	433	374	399	341	317	299	283
SDS-[78]	352	366	392	338	335	379	304	300	296	277
EA[61]	443	450	394	410	349	414	359	308	296	282
DA[61]	352	350	373	354	323	367	309	305	299	269
Psn Avg	396	391	386	385	342	385	320	306	296	278

Table 19. NovelConcepts10 CLIP Score for various poison defenses.

$\begin{array}{l} \text{Defense} \rightarrow \\ \text{Poison} \downarrow \end{array}$	Nominal	Regen [84]	PDMPure [77]	AdvClean [65]	JPEG	T600	LM	JPEG +T600	JPEG +LM	SZT
Clean	0.51	0.47	0.49	0.53	0.48	0.49	0.52	0.51	0.50	0.52
ADM+[39]	0.46	0.47	0.49	0.46	0.47	0.50	0.47	0.50	0.51	0.52
ADM-	0.45	0.50	0.46	0.50	0.46	0.45	0.54	0.51	0.50	0.49
SDS+[78]	0.44	0.49	0.47	0.48	0.47	0.48	0.48	0.48	0.50	0.49
SDS-[78]	0.45	0.48	0.48	0.48	0.48	0.44	0.49	0.51	0.51	0.52
EA[61]	0.45	0.45	0.47	0.48	0.48	0.46	0.50	0.48	0.47	0.48
DA[61]	0.49	0.49	0.48	0.49	0.48	0.52	0.48	0.52	0.53	0.46
Psn Avg	0.46	0.48	0.48	0.48	0.47	0.47	0.49	0.50	0.50	0.49



Figure 18. Images generated after TI on things-bottle1 (CustomConcept101) for various poisons. Prompt: "a R\* in the snow"



Figure 19. Images generated after TI on various concepts from CustomConcept101 poisoned by ADM+. Prompt: "a R\* in the snow"



Figure 20. Images generated after TI on FishDoll (NovelConcepts10) for various poisons. Prompt: "a R\* on the beach"



Figure 21. Images generated after TI on various concepts from NovelConcepts10 poisoned by ADM+. Prompt: "a R\* on the beach"

### **N. JPEG Preprocessing for Other Defenses**

Given the success of JPEG with SZT in Sec. 5.2.4, we further analyze JPEG preprocessing for existing defenses in Tables 20, 21, and 22. In general, adding JPEG preprocessing improves performance for Regeneration and Adverse-Cleaner, but reduces performance for PDMPure. SZT still outperforms all baseline defenses with JPEG processing.

### **O. Additional Diffusion Models**

We further validate the effectiveness of SZT and existing defenses for TI applied to multiple LDMs. We analyze Stable Diffusion 1.5 (rectified flow version) [43], Stable Diffusion 2.1 (noise-prediction), Stable Diffusion 2.1 (velocity-prediction) [40], and SDXL [55]. For convenience, we abbreviate the models as SD15, SD15rf, SD21, SD21v, and SDXL. We evaluate all models on NovelConcepts10 for all poisons. Training settings are generally similar to those described in Sec. 5.1.2 except that we use 2500 training steps. Due to the large size of SDXL, we instead use 500 training steps, each with 5 steps of gradient accumulation. We were unable to track CLIP Score for SD21, SD21v, or SDXL due to their text encoders' incompatibilities with the OpenAI CLIP image encoder.

This experiment approaches a "black-box" poisoning scenario, as we craft all poisons using SD15 and then test defenses on different Stable Diffusion versions. This is not a major concern, as multiple prior works have demonstrated the strong transferability of poisons across Stable Diffusion models [37, 78]. In multiple tables for SD15rf, SD21, and SD21v results below, the "Nominal" defense column indeed shows that all poisons are effective. Network-based defenses (Regen and PDMPure) shown here still use the same models for purification.

We evaluate SD15rf in Tables 23, 24, and 25. We suspect that the poor performance of SD15rf is due to the fact that it derives from a reflow procedure [42]. We hypothesize that reflow, which relies on deterministic sampling for data/noise pairs to finetune models, acts as a type of model distillation and forces models to unlearn natural ODE paths between probability distributions. The rigid ODE paths enforced by distilled models may lack the flexibility to insert new concepts into the model via personalization. Existing defenses like Regen and AdvClean barely improve the poison performance. Even so, SZT improves generative quality for TI on poisoned data to match that of clean data.

Poison and defense performance on SD21 in Tables 26 and 27 is generally similar to that of SD15, which is sensible given their similar architectures, training data, and training objectives. The trends for SD21v in Tables 28 and 29 are generally similar to those of SD15 and SD21, though the effective of poisoning (i.e., the difference between "Clean" and "Psn Avg" values) is diminished relative to SD15 and SD21. Even so, SZT and its ablations are still effective for improving generation quality on poisoned data. In Tables 30 and 31, SDXL demonstrates the largest rift in performance trends relative to SD15, being unaffected by most poisons. Though SZT and its ablations can improve performance on poison data they are generally not necessary. We are uncertain of the exact cause of low degree of poison transferability from SD15 to SDXL.

### **P. Additional Personalization Methods**

In addition to TI, we investigate additional personalization methods for Stable Diffusion 1.5, namely LoRA [25] and CustomDiffusion [35]. We use NovelConcepts10 for these experiments. LoRA personalizes LDMs by finetuning lowrank adapters for U-Net weights (and optionally for the text encoder). LoRA does not finetune any text embeddings, though it does associate finetuned weights with a target prompt. CustomDiffusion is a compact version of Dream-Booth [59] that only finetunes the cross-attention weights of the U-Net as well as a new text token (as in TI). Notably, the CustomDiffusion paper also introduced a crop/rescale augmentation that applies as a mask during loss calculation, similar to our LM strategy.

In our implementation of LoRA, we use rank 4 adapters with a target prompt that includes "\*" as a dummy token to signify LoRA usage and trained for 1000 steps. We did not find any improvement in generative quality when using the prior preservation loss or when training text encoder weights, and thus our implementation does not include these methods. We report LoRA results in Tables 32, 33, and 34. We find that poisons are moderately effective against LoRA, with EA being the most severe. Existing defenses like PDMPure and AdvClean slightly improve defense against poisons but Regen generally reduces generative quality. SZT and its ablations outperform existing defenses to achieve clean-level generative quality for poisoned images. We note that as LoRA does not provide any finetuned text tokens, the CLIP Score becomes a metric for background fidelity; the CLIP text encoder will ignore the dummy token ("\*") and instead focus on context words.

In our implementation of CustomDiffusion, we train for 1000 steps and do not use the prior preservation loss. We report results both without their novel crop/rescale augmentation (Tables 35, 36, and 37) and with it (Tables 38, 39, and 40). Without crop/rescale, performance is generally similar to that of LoRA, with SZT outperforming existing defenses to improve poison performance. With crop/rescale, all poisons are almost ineffective and most defenses (with the exception of JPEG+T600) only reduce generative performance. This suggests that their novel crop/rescale augmentation, which is applied during loss calculation, acts similarly to our LM method, effectively blocking out adversarial signals outside the novel concept region.

$\begin{array}{l} \text{Defense} \rightarrow \\ \text{Poison} \downarrow \end{array}$	Nominal	Regen [84]	PDMPure [77]	AdvClean [65]	JPEG	JPEG + Regen	JPEG + PDMPure	JPEG + AdvClean	SZT
Clean	0.44	0.43	0.24	0.45	0.41	0.37	0.19	0.38	0.47
ADM+[39]	0.10	0.14	0.24	0.13	0.28	0.26	0.20	0.30	0.43
ADM-	0.29	0.27	0.21	0.31	0.34	0.35	0.19	0.33	0.44
SDS+[78]	0.11	0.16	0.22	0.12	0.23	0.23	0.19	0.29	0.44
SDS-[78]	0.25	0.25	0.20	0.30	0.30	0.35	0.19	0.35	0.45
EA[61]	0.08	0.09	0.19	0.16	0.28	0.29	0.22	0.32	0.44
DA[61]	0.27	0.26	0.23	0.28	0.33	0.33	0.17	0.35	0.47
Psn Avg	0.18	0.19	0.22	0.21	0.29	0.30	0.19	0.32	0.45

Table 20. NovelConcepts10 DINOv2 Similarity for existing defenses with JPEG preprocessing.

Table 21. NovelConcepts10 FID for existing defenses with JPEG preprocessing.

$\begin{array}{c} \text{Defense} \rightarrow \\ \text{Poison} \downarrow \end{array}$	Nominal	Regen [84]	PDMPure [77]	AdvClean [65]	JPEG	JPEG + Regen	JPEG + PDMPure	JPEG + AdvClean	SZT
Clean	285	285	380	281	294	303	389	303	276
ADM+[39]	453	416	378	443	349	355	389	335	280
ADM-	339	353	390	336	324	303	387	315	279
SDS+[78]	439	410	389	433	374	364	388	335	283
SDS-[78]	352	366	392	338	335	310	398	309	277
EA[61]	443	450	394	410	349	335	370	321	282
DA[61]	352	350	373	354	323	317	396	311	269
Psn Avg	396	391	386	385	342	331	388	321	278

Table 22. NovelConcepts10 CLIP Score for existing defenses with JPEG preprocessing.

$\begin{array}{c} \text{Defense} \rightarrow \\ \text{Poison} \downarrow \end{array}$	Nominal	Regen [84]	PDMPure [77]	AdvClean [65]	JPEG	JPEG + Regen	JPEG + PDMPure	JPEG + AdvClean	SZT
Clean	0.51	0.47	0.49	0.53	0.48	0.49	0.46	0.52	0.52
ADM+[39]	0.46	0.47	0.49	0.46	0.47	0.49	0.51	0.47	0.52
ADM-	0.45	0.50	0.46	0.50	0.46	0.48	0.49	0.50	0.49
SDS+[78]	0.44	0.49	0.47	0.48	0.47	0.50	0.51	0.47	0.49
SDS-[78]	0.45	0.48	0.48	0.48	0.48	0.50	0.49	0.51	0.52
EA[61]	0.45	0.45	0.47	0.48	0.48	0.49	0.49	0.48	0.48
DA[61]	0.49	0.49	0.48	0.49	0.48	0.48	0.44	0.49	0.46
Psn Avg	0.46	0.48	0.48	0.48	0.47	0.49	0.49	0.49	0.49

$\begin{array}{l} \text{Defense} \rightarrow \\ \text{Poison} \downarrow \end{array}$	Nominal	Regen [84]	PDMPure [77]	AdvClean [65]	JPEG	T600	LM	JPEG +T600	JPEG +LM	SZT
Clean	0.21	0.23	0.20	0.20	0.18	0.17	0.30	0.19	0.28	0.28
ADM+[39]	0.03	0.04	0.17	0.02	0.08	0.18	0.09	0.16	0.21	0.24
ADM-	0.12	0.13	0.20	0.13	0.15	0.12	0.18	0.14	0.20	0.23
SDS+[78]	0.04	0.03	0.15	0.04	0.09	0.17	0.10	0.15	0.21	0.24
SDS-[78]	0.13	0.12	0.18	0.14	0.12	0.10	0.17	0.15	0.22	0.18
EA[61]	0.07	0.09	0.21	0.10	0.14	0.10	0.14	0.13	0.22	0.19
DA[61]	0.10	0.08	0.17	0.10	0.15	0.11	0.19	0.16	0.21	0.23
Psn Avg	0.08	0.08	0.18	0.09	0.12	0.13	0.14	0.15	0.21	0.22

Table 23. NovelConcepts10 DINOv2 Similarity for various poison defenses with Stable Diffusion 1.5 (rectified flow).

Table 24. NovelConcepts10 FID for various poison defenses with Stable Diffusion 1.5 (rectified flow).

$\begin{array}{c} \text{Defense} \rightarrow \\ \text{Poison} \downarrow \end{array}$	Nominal	Regen [84]	PDMPure [77]	AdvClean [65]	JPEG	T600	LM	JPEG +T600	JPEG +LM	SZT
Clean	385	364	380	386	391	389	343	385	356	352
ADM+[39]	440	443	394	455	419	388	416	396	379	367
ADM-	415	408	381	408	407	403	393	402	383	370
SDS+[78]	440	442	401	438	426	398	417	400	367	364
SDS-[78]	407	416	397	409	409	410	395	398	371	399
EA[61]	430	429	384	421	399	419	409	408	375	387
DA[61]	422	422	400	425	400	408	395	389	374	368
Psn Avg	426	427	393	426	410	404	404	399	375	376

Table 25. NovelConcepts10 CLIP Score for various poison defenses with Stable Diffusion 1.5 (rectified flow).

$\begin{array}{c} \text{Defense} \rightarrow \\ \text{Poison} \downarrow \end{array}$	Nominal	Regen [84]	PDMPure [77]	AdvClean [65]	JPEG	T600	LM	JPEG +T600	JPEG +LM	SZT
Clean	0.44	0.41	0.46	0.43	0.42	0.46	0.47	0.43	0.47	0.44
ADM+[39]	0.40	0.41	0.45	0.38	0.43	0.42	0.40	0.42	0.46	0.48
ADM-	0.41	0.44	0.46	0.43	0.44	0.45	0.46	0.43	0.45	0.45
SDS+[78]	0.37	0.37	0.44	0.39	0.42	0.45	0.40	0.44	0.47	0.46
SDS-[78]	0.39	0.45	0.45	0.45	0.42	0.44	0.47	0.44	0.47	0.46
EA[61]	0.40	0.42	0.45	0.44	0.42	0.43	0.43	0.44	0.45	0.46
DA[61]	0.43	0.45	0.45	0.43	0.44	0.43	0.46	0.44	0.45	0.47
Psn Avg	0.40	0.42	0.45	0.42	0.43	0.44	0.43	0.43	0.46	0.46

$\begin{array}{l} \text{Defense} \rightarrow \\ \text{Poison} \downarrow \end{array}$	Nominal	Regen [84]	PDMPure [77]	AdvClean [65]	JPEG	T600	LM	JPEG +T600	JPEG +LM	SZT
Clean	0.41	0.39	0.31	0.41	0.38	0.43	0.44	0.41	0.44	0.43
ADM+[39]	0.10	0.14	0.29	0.10	0.30	0.20	0.34	0.38	0.40	0.44
ADM-	0.21	0.29	0.26	0.33	0.34	0.28	0.42	0.37	0.44	0.40
SDS+[78]	0.08	0.13	0.24	0.10	0.32	0.18	0.36	0.32	0.44	0.43
SDS-[78]	0.23	0.24	0.24	0.26	0.31	0.22	0.41	0.37	0.42	0.43
EA[61]	0.09	0.07	0.26	0.10	0.26	0.12	0.33	0.38	0.43	0.41
DA[61]	0.27	0.23	0.26	0.31	0.35	0.27	0.42	0.38	0.45	0.44
Psn Avg	0.16	0.18	0.26	0.20	0.31	0.21	0.38	0.36	0.43	0.42

Table 26. NovelConcepts10 DINOv2 Similarity for various poison defenses with Stable Diffusion 2.1 (noise-prediction).

Table 27. NovelConcepts10 FID for various poison defenses with Stable Diffusion 2.1 (noise-prediction).

$\begin{array}{c} \text{Defense} \rightarrow \\ \text{Poison} \downarrow \end{array}$	Nominal	Regen [84]	PDMPure [77]	AdvClean [65]	JPEG	T600	LM	JPEG +T600	JPEG +LM	SZT
Clean	290	305	356	298	301	295	279	294	288	286
ADM+[39]	440	421	355	429	339	393	319	306	295	276
ADM-	377	342	365	320	320	341	291	313	283	303
SDS+[78]	445	439	374	433	322	399	317	323	289	286
SDS-[78]	362	368	383	356	318	371	299	313	290	295
EA[61]	421	448	372	406	355	417	335	310	281	303
DA[61]	350	370	374	340	310	361	299	313	280	291
Psn Avg	399	398	370	380	327	380	310	313	286	292

Table 28. NovelConcepts10 DINOv2 Similarity for various poison defenses with Stable Diffusion 2.1 (velocity-prediction).

$\begin{array}{l} \text{Defense} \rightarrow \\ \text{Poison} \downarrow \end{array}$	Nominal	Regen [84]	PDMPure [77]	AdvClean [65]	JPEG	T600	LM	JPEG +T600	JPEG +LM	SZT
Clean	0.28	0.31	0.26	0.34	0.30	0.29	0.39	0.31	0.41	0.39
ADM+[39]	0.25	0.23	0.29	0.30	0.29	0.24	0.37	0.24	0.42	0.37
ADM-	0.20	0.16	0.25	0.16	0.22	0.17	0.31	0.28	0.36	0.38
SDS+[78]	0.21	0.21	0.31	0.27	0.25	0.23	0.33	0.28	0.40	0.38
SDS-[78]	0.16	0.14	0.28	0.17	0.35	0.17	0.27	0.26	0.38	0.38
EA[61]	0.09	0.09	0.28	0.21	0.26	0.13	0.27	0.26	0.38	0.40
DA[61]	0.21	0.17	0.26	0.28	0.29	0.16	0.28	0.27	0.37	0.37
Psn Avg	0.19	0.17	0.28	0.23	0.28	0.18	0.30	0.27	0.38	0.38

Defense $\rightarrow$ Poison $\downarrow$	Nominal	Regen [84]	PDMPure [77]	AdvClean [65]	JPEG	T600	LM	JPEG +T600	JPEG +LM	SZT
Clean	336	328	343	321	326	334	308	330	295	314
ADM+[39]	355	362	332	332	335	370	309	360	292	309
ADM-	376	389	353	381	368	381	329	347	313	319
SDS+[78]	393	385	327	332	341	367	323	350	294	309
SDS-[78]	388	401	336	383	318	386	341	345	306	317
EA[61]	431	436	336	370	348	420	371	352	300	316
DA[61]	369	378	348	331	341	382	361	357	310	315
Psn Avg	385	392	339	355	342	384	339	352	303	314

Table 29. NovelConcepts10 FID for various poison defenses with Stable Diffusion 2.1 (velocity-prediction).

Table 30. NovelConcepts10 DINOv2 Similarity for various poison defenses with SDXL.

$\begin{array}{l} \text{Defense} \rightarrow \\ \text{Poison} \downarrow \end{array}$	Nominal	Regen [84]	PDMPure [77]	AdvClean [65]	JPEG	T600	LM	JPEG +T600	JPEG +LM	SZT
Clean	0.35	0.33	0.25	0.33	0.22	0.41	0.42	0.34	0.39	0.39
ADM+[39]	0.36	0.29	0.24	0.31	0.25	0.39	0.42	0.32	0.34	0.32
ADM-	0.30	0.26	0.23	0.34	0.20	0.40	0.38	0.32	0.38	0.39
SDS+[78]	0.34	0.37	0.27	0.32	0.29	0.39	0.41	0.39	0.38	0.42
SDS-[78]	0.32	0.30	0.26	0.35	0.17	0.36	0.43	0.35	0.37	0.36
EA[61]	0.30	0.16	0.22	0.30	0.20	0.35	0.42	0.30	0.31	0.33
DA[61]	0.34	0.31	0.29	0.33	0.21	0.38	0.41	0.36	0.35	0.38
Psn Avg	0.33	0.28	0.25	0.32	0.22	0.38	0.41	0.34	0.35	0.37

Table 31. NovelConcepts10 FID for various poison defenses with SDXL.

$\begin{array}{c} \text{Defense} \rightarrow \\ \text{Poison} \downarrow \end{array}$	Nominal	Regen [84]	PDMPure [77]	AdvClean [65]	JPEG	T600	LM	JPEG +T600	JPEG +LM	SZT
Clean	325	335	374	333	374	303	293	332	319	322
ADM+[39]	314	340	383	342	358	319	311	326	327	329
ADM-	341	351	382	330	370	307	316	347	314	310
SDS+[78]	317	313	370	332	342	321	312	317	314	298
SDS-[78]	336	322	371	321	386	325	297	323	322	332
EA[61]	337	421	387	334	376	328	303	337	344	321
DA[61]	325	334	356	331	378	325	319	318	324	313
Psn Avg	328	347	375	332	368	321	310	328	324	317

$\begin{array}{c} \text{Defense} \rightarrow \\ \text{Poison} \downarrow \end{array}$	Nominal	Regen [84]	PDMPure [77]	AdvClean [65]	JPEG	T600	LM	JPEG +T600	JPEG +LM	SZT
Clean	0.35	0.35	0.25	0.34	0.35	0.47	0.39	0.42	0.38	0.44
ADM+[39]	0.26	0.26	0.28	0.28	0.35	0.28	0.31	0.35	0.31	0.35
ADM-	0.33	0.29	0.29	0.35	0.32	0.28	0.32	0.40	0.36	0.42
SDS+[78]	0.32	0.26	0.31	0.29	0.30	0.28	0.28	0.31	0.32	0.34
SDS-[78]	0.31	0.27	0.28	0.35	0.32	0.26	0.31	0.41	0.35	0.38
EA[61]	0.11	0.13	0.30	0.18	0.29	0.09	0.13	0.35	0.31	0.36
DA[61]	0.29	0.27	0.27	0.32	0.32	0.24	0.28	0.40	0.34	0.38
Psn Avg	0.27	0.25	0.29	0.30	0.32	0.24	0.27	0.37	0.33	0.37

Table 32. NovelConcepts10 DINOv2 Similarity for various poison defenses with LoRA.

Table 33. NovelConcepts10 FID for various poison defenses with LoRA.

$\begin{array}{l} \text{Defense} \rightarrow \\ \text{Poison} \downarrow \end{array}$	Nominal	Regen [84]	PDMPure [77]	AdvClean [65]	JPEG	T600	LM	JPEG +T600	JPEG +LM	SZT
Clean	322	311	365	316	315	261	301	284	303	280
ADM+[39]	364	363	357	367	312	383	360	310	337	325
ADM-	328	347	360	318	319	345	350	284	316	287
SDS+[78]	335	355	348	346	339	375	363	340	330	323
SDS-[78]	338	350	357	318	320	355	345	282	317	298
EA[61]	435	433	347	414	351	432	425	328	339	318
DA[61]	348	356	365	342	323	388	352	292	323	311
Psn Avg	358	367	356	351	327	380	366	306	327	310

Table 34. NovelConcepts10 CLIP Score for various poison defenses with LoRA.

$\begin{array}{c} \text{Defense} \rightarrow \\ \text{Poison} \downarrow \end{array}$	Nominal	Regen [84]	PDMPure [77]	AdvClean [65]	JPEG	T600	LM	JPEG +T600	JPEG +LM	SZT
Clean	0.48	0.48	0.51	0.48	0.48	0.41	0.44	0.44	0.45	0.42
ADM+[39]	0.49	0.49	0.50	0.50	0.48	0.46	0.45	0.45	0.48	0.45
ADM-	0.47	0.48	0.48	0.47	0.49	0.42	0.46	0.43	0.46	0.43
SDS+[78]	0.48	0.49	0.47	0.50	0.49	0.45	0.47	0.46	0.48	0.45
SDS-[78]	0.48	0.46	0.47	0.48	0.49	0.44	0.45	0.44	0.46	0.44
EA[61]	0.42	0.42	0.46	0.43	0.48	0.41	0.43	0.44	0.47	0.45
DA[61]	0.47	0.48	0.48	0.48	0.49	0.44	0.47	0.44	0.47	0.45
Psn Avg	0.47	0.47	0.48	0.48	0.48	0.44	0.45	0.44	0.47	0.44

$\begin{array}{l} \text{Defense} \rightarrow \\ \text{Poison} \downarrow \end{array}$	Nominal	Regen [84]	PDMPure [77]	AdvClean [65]	JPEG	T600	LM	JPEG +T600	JPEG +LM	SZT
Clean	0.30	0.31	0.27	0.30	0.30	0.37	0.30	0.34	0.32	0.37
ADM+[39]	0.26	0.28	0.29	0.28	0.30	0.27	0.31	0.33	0.34	0.35
ADM-	0.29	0.25	0.27	0.30	0.31	0.20	0.29	0.32	0.32	0.33
SDS+[78]	0.25	0.27	0.28	0.26	0.29	0.26	0.29	0.33	0.33	0.35
SDS-[78]	0.27	0.25	0.28	0.30	0.32	0.19	0.29	0.32	0.34	0.38
EA[61]	0.14	0.15	0.24	0.19	0.29	0.13	0.19	0.30	0.33	0.33
DA[61]	0.28	0.25	0.27	0.30	0.31	0.23	0.31	0.29	0.34	0.34
Psn Avg	0.25	0.24	0.27	0.27	0.30	0.21	0.28	0.31	0.33	0.35

Table 35. NovelConcepts10 DINOv2 Similarity for various poison defenses with CustomDiffusion (no crop/rescale augmentation).

Table 36. NovelConcepts10 FID for various poison defenses with CustomDiffusion (no crop/rescale augmentation).

$\begin{array}{l} \text{Defense} \rightarrow \\ \text{Poison} \downarrow \end{array}$	Nominal	Regen [84]	PDMPure [77]	AdvClean [65]	JPEG	T600	LM	JPEG +T600	JPEG +LM	SZT
Clean	336	333	346	331	327	305	335	316	324	307
ADM+[39]	347	346	342	339	331	347	330	322	311	306
ADM-	335	352	349	338	334	374	338	332	324	320
SDS+[78]	357	345	341	354	334	350	341	319	325	309
SDS-[78]	344	348	344	333	324	383	343	330	314	302
EA[61]	404	393	360	382	339	408	389	336	318	321
DA[61]	342	351	347	332	333	368	331	339	315	310
Psn Avg	355	356	347	346	333	372	345	330	318	312

Table 37. NovelConcepts10 CLIP Score for various poison defenses with CustomDiffusion (no crop/rescale augmentation).

$\begin{array}{c} \text{Defense} \rightarrow \\ \text{Poison} \downarrow \end{array}$	Nominal	Regen [84]	PDMPure [77]	AdvClean [65]	JPEG	T600	LM	JPEG +T600	JPEG +LM	SZT
Clean	0.37	0.38	0.38	0.38	0.38	0.39	0.38	0.39	0.40	0.39
ADM+[39]	0.40	0.39	0.40	0.39	0.39	0.39	0.40	0.40	0.40	0.39
ADM-	0.38	0.39	0.39	0.38	0.38	0.41	0.39	0.38	0.40	0.39
SDS+[78]	0.40	0.38	0.38	0.40	0.39	0.40	0.39	0.39	0.39	0.40
SDS-[78]	0.37	0.39	0.38	0.38	0.39	0.41	0.40	0.39	0.39	0.40
EA[61]	0.38	0.38	0.38	0.37	0.38	0.39	0.40	0.39	0.39	0.39
DA[61]	0.39	0.38	0.38	0.38	0.38	0.39	0.39	0.39	0.38	0.39
Psn Avg	0.39	0.39	0.38	0.38	0.39	0.40	0.39	0.39	0.39	0.39

$\begin{array}{l} \text{Defense} \rightarrow \\ \text{Poison} \downarrow \end{array}$	Nominal	Regen [84]	PDMPure [77]	AdvClean [65]	JPEG	T600	LM	JPEG +T600	JPEG +LM	SZT
Clean	0.35	0.35	0.33	0.36	0.33	0.36	0.34	0.36	0.34	0.32
ADM+[39]	0.32	0.30	0.33	0.34	0.31	0.34	0.33	0.33	0.33	0.30
ADM-	0.33	0.25	0.33	0.33	0.33	0.33	0.31	0.34	0.34	0.30
SDS+[78]	0.33	0.31	0.34	0.31	0.32	0.32	0.32	0.34	0.34	0.30
SDS-[78]	0.32	0.26	0.32	0.34	0.33	0.31	0.31	0.34	0.32	0.30
EA[61]	0.31	0.21	0.29	0.31	0.31	0.29	0.32	0.32	0.33	0.30
DA[61]	0.36	0.33	0.32	0.32	0.32	0.34	0.33	0.35	0.33	0.30
Psn Avg	0.33	0.28	0.32	0.32	0.32	0.32	0.32	0.34	0.33	0.30

Table 38. NovelConcepts10 DINOv2 Similarity for various poison defenses with CustomDiffusion (crop/rescale augmentation).

Table 39. NovelConcepts10 FID for various poison defenses with CustomDiffusion (crop/rescale augmentation).

$\begin{array}{l} \text{Defense} \rightarrow \\ \text{Poison} \downarrow \end{array}$	Nominal	Regen [84]	PDMPure [77]	AdvClean [65]	JPEG	T600	LM	JPEG +T600	JPEG +LM	SZT
Clean	316	313	321	315	323	314	320	313	320	325
ADM+[39]	329	330	325	319	326	321	323	324	322	333
ADM-	324	353	320	327	320	327	327	321	321	334
SDS+[78]	323	331	319	329	325	328	326	320	316	328
SDS-[78]	324	357	326	320	317	338	329	324	327	337
EA[61]	332	375	336	335	331	349	324	331	323	337
DA[61]	311	326	329	327	327	320	325	321	324	332
Psn Avg	324	345	326	326	325	330	326	324	322	333

Table 40. NovelConcepts10 CLIP Score for various poison defenses with CustomDiffusion (crop/rescale augmentation).

$\begin{array}{c} \text{Defense} \rightarrow \\ \text{Poison} \downarrow \end{array}$	Nominal	Regen [84]	PDMPure [77]	AdvClean [65]	JPEG	T600	LM	JPEG +T600	JPEG +LM	SZT
Clean	0.41	0.40	0.42	0.41	0.40	0.40	0.42	0.41	0.42	0.40
ADM+[39]	0.41	0.41	0.41	0.40	0.41	0.41	0.42	0.41	0.43	0.41
ADM-	0.42	0.42	0.41	0.40	0.41	0.40	0.42	0.39	0.43	0.42
SDS+[78]	0.42	0.43	0.42	0.40	0.43	0.41	0.42	0.40	0.42	0.42
SDS-[78]	0.42	0.40	0.41	0.41	0.41	0.42	0.41	0.41	0.42	0.41
EA[61]	0.41	0.40	0.42	0.41	0.40	0.41	0.40	0.40	0.42	0.41
DA[61]	0.41	0.42	0.42	0.40	0.42	0.41	0.42	0.40	0.42	0.4
Psn Avg	0.41	0.41	0.42	0.40	0.41	0.41	0.42	0.40	0.42	0.41

# **Q. Ethical Statement**

As our research primarily concerns the subfield of data poisoning, we are keenly aware of our work's ethical proximity to copyright theft and artistic style copying. Practical applications of SZT (and related methods) may realize as improved attacks against attempts by copyright holders and artists to protect their works. Even so, we believe that our research is necessary. The individual components of SZT are not complex, simply relying on JPEG compression, biased timestep sampling, and loss masking. Rather, we believe that the effectiveness of SZT, despite its simplicity, exposes the vulnerabilities of existing poisons and demands further research on robust poisons.

## **R.** Contributions

Dataset Collection: Styborski, Kapur, Lu Poison Curation: Styborski, Kapur Defense Baselines: Lu, Kapur, Styborski Semantic Sensitivity Map Studies: Lyu Timestep Learning Bias Studies: Styborski Spatial Learning Bias Studies: Lyu, Styborski JPEG Compression Studies: Styborski Hyperparameter Ablation: Styborski Model Ablation: Lyu, Styborski, Lu Additional Personalization Methods: Styborski Writing: Styborski, Lyu, Kong Editing: Kong, Styborski, Lyu Advising and Guidance: Kong