Understanding Implosion in Text-to-Image Generative Models

Wenxin Ding, Cathy Y. Li, Shawn Shan, Ben Y. Zhao, Haitao Zheng

Department of Computer Science, University of Chicago

{wenxind,yuanchen,shawnshan,ravenben,htzheng}@cs.uchicago.edu

ABSTRACT

Recent works show that text-to-image generative models are surprisingly vulnerable to a variety of poisoning attacks. Empirical results find that these models can be corrupted by altering associations between individual text prompts and associated visual features. Furthermore, a number of concurrent poisoning attacks can induce "model implosion," where the model becomes unable to produce meaningful images for unpoisoned prompts. These intriguing findings highlight the absence of an intuitive framework to understand poisoning attacks on these models.

In this work, we establish the first analytical framework on robustness of image generative models to poisoning attacks, by modeling and analyzing the behavior of the cross-attention mechanism in latent diffusion models. We model cross-attention training as an abstract problem of "supervised graph alignment" and formally quantify the impact of training data by the hardness of alignment, measured by an Alignment Difficulty (AD) metric. The higher the AD, the harder the alignment. We prove that AD increases with the number of individual prompts (or concepts) poisoned. As AD grows, the alignment task becomes increasingly difficult, yielding highly distorted outcomes that frequently map meaningful text prompts to undefined or meaningless visual representations. As a result, the generative model implodes and outputs random, incoherent images at large. We validate our analytical framework through extensive experiments, and we confirm and explain the unexpected (and unexplained) effect of model implosion while producing new, unforeseen insights. Our work provides a useful tool for studying poisoning attacks against diffusion models and their defenses.

1 INTRODUCTION

Large-scale text-to-image generative models like Stable Diffusion, Midjourney, DALLE, and Adobe Firefly have made tremendous impact on various artistic and creative industries. Each of these models is trained on hundreds of millions, if not billions, of images and corresponding text captions. Given prior understanding of poisoning attacks on deep neural networks, many believe that employing such massive training datasets makes these generative models naturally robust to poisoning attacks.

Surprisingly, recent results have shown these large diffusion models to be quite vulnerable to poisoning attacks targeting the connections between textual prompts and image visual features. Multiple projects [24, 40, 49, 54] have demonstrated the use of poisoning attacks to successfully disrupt style mimicry models,

which are locally fine-tuned copies of image generative models to learn and replicate specific styles. Taking it a step further, recent work [41] has shown that poisoning attacks can directly target generic image generation models like Stable Diffusion, and successfully manipulate the associations between individual prompts and generated images. More importantly, [41] shows that a number of concurrent poisoning attacks can induce a form of "model implosion" where the model becomes generally unable to produce meaningful images even for unpoisoned prompts.

These empirical observations are both intriguing and unexpected. They raise critical questions about the inherent robustness of text-image alignment in large-scale latent diffusion models. In particular, is model implosion a real, consistent phenomenon across various tasks, datasets and model architectures? If so, what are the mechanisms and triggers that cause a model to implode under concurrent poisoning attacks? Which image generation models are more susceptible to these attacks? Can existing poisoning defenses offer protection against model implosion?

In this paper, we attempt to answer these questions, by building an analytical framework to capture the behavior of text-image alignment in latent diffusion models and their properties under poisoning attacks. For this, we propose to model the practical task of training the cross-attention module in the generative models as an abstract problem of *supervised graph alignment*.

Cross-Attention as Supervised Graph Alignment. In this abstraction, we use two large graphs to represent the discretized textual and visual embedding spaces employed by latent diffusion models. We represent the cross-attention mechanism as vertex mapping aiming at aligning the two graphs. The text/image pairs used to train a generative model serve as the labeled training data to supervise the graph alignment process. As such, we can model and analyze the impact of (poisoned) training data on generative models by examining them within the framework of supervised graph alignment.

We introduce a new metric, *Alignment Difficulty (AD)*, to measure the hardness of supervised graph alignment for a given set of (poisoned) training data. Our intuition is that AD reflects the amount of learning capacity necessary to learn any new joint distribution displayed by the training data. The larger the AD, the harder it is to find a practical model carrying such learning capacity, and the poorer the alignment outcomes.

We then use AD to quantify the impact of poisoned training data on graph alignment (thus on the trained generative models). We formally prove that AD increases with the number of concepts poisoned. This illustrates how a broader range of poisoned data increases the complexity of the joint distribution to be learned during training. This further leads to a conjecture that when AD is large, the alignment task becomes exceedingly challenging and thus infeasible to solve by any practical model. Instead, the model learns a largely distorted version of the joint distribution (e.g. by

Wenxin Ding, Cathy Y. Li, Shawn Shan, Ben Y. Zhao, Haitao Zheng

applying weighted averaging or fitting a different distribution), which often maps a meaningful text embedding to a "meaningless" visual embedding. As a result, the text-to-image generative model *implodes* and outputs random, incoherent images at large.

We validate our analytical framework using empirical experiments, by varying datasets, diffusion model architectures, training scenarios (training-from-scratch vs. fine-tuning), and poisoned data composition (clean-label vs. dirty-label). Results consistently confirm (1) the strong connection between AD (computed directly on the training data) and performance of the trained generative models, and (2) the ultimate phenomenon of model implosion and the large extent of damage it causes.

Our study also reveals several critical and unforeseen insights on model implosion, much beyond those identified by [41]. We summarize them below.

- When operating individually, each prompt-specific poisoning attack [41] misleads the model into learning a wrong association between a specific pair of textual and visual features. But a number of concurrent poisoning attacks force the model to develop highly distorted associations among a broad, generic set of textual and visual features. Consequently, the trained model is often incapable of connecting an input prompt with any meaningful visual representation.

- Generative models employing "overfitted" or unstable feature extractors [35, 50] are more susceptible to model implosion because this instability amplifies the implosion damage.

- Stealthy clean-label poison triggers model implosion just like its dirty-label counterparts, but requires poisoning more concepts to produce the same level of damage.

- Traditional poisoning defenses are unable to stall model implosion or recover from it efficiently. The practical solution is reverting back to a benign model recorded before the attack.

Our Contributions. Our work makes four key contributions:

- We perform a detailed study on the phenomenon of model implosion caused by data poisoning, demonstrating its significant impact on text-to-image generative models.

- We propose the first analytical framework to model the impact of poisoned training data on text-to-image diffusion models, especially on how they affect the learned textual conditions (§4).

- We verify our analytical framework and its conclusions with extensive experiments, confirm (and explain) the empirically observed phenomenon of model implosion while producing new, unforeseen insights on model implosion (§5).

- We apply our analytical framework to study the efficacy of poisoning defenses, outlining both challenges and opportunities (§6).

Overall, our analytical framework provides a useful tool for studying poisoning attacks against diffusion models and their defenses. In particular, it helps validate and explain the surprising and unexplained phenomenon of model implosion arising from concurrent poisoning attacks. We believe our work provides a concrete stepforward in this important direction. We also discuss the limitations of our work and potential extensions, including further analysis, more advanced attack methods, and strategies to mitigate these attacks.
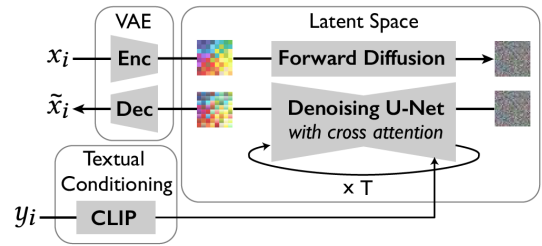

Figure 1: Training pipeline of latent diffusion models.

Ethics. We perform experiments on datasets that are publicly available, with no report of harmful materials such as CSAM.

2 BACKGROUND AND RELATED WORK

2.1 Diffusion-Based Text-to-Image Generation

Diffusion models are known to achieve state-of-the-art performance in text-to-image generation [11, 46]. Latent diffusion models (LDMs) are widely adopted for their efficiency in both training and inference [31, 33, 34]. Figure 1 sketches the training pipeline for text-to-image LDMs. The training data consists of images and their text prompts. Given a text/image pair, the model first applies a feature extractor (e.g. a variational autoencoder (VAE) [10]) to represent the image as a latent embedding, and a text encoder (e.g. CLIP) to encode the prompt as a textual embedding. Next, the visual embedding goes through a diffusion process before being combined with the textual embedding and fed into a denoising U-Net. The U-Net module employs cross-attention [28] to learn textual conditions [34], allowing the model to generate images conditioned on an input prompt. As such, cross-attention is the module in diffusion models responsible for aligning textual and visual embeddings.

Cross-Attention Maps. Cross-attention maps, proposed in [17], are visual representations of the cross-attention layers for each generated image. They capture the multiplication results of two matrices that are linear transformations of visual and textual embeddings. Given an image x generated by prompt y, one can calculate the cross-attention map with respect to a token t in y, by averaging corresponding cross-attention layers over all diffusion steps [17]. The resulting map is a grayscale image, from which one can observe the *object attribution* the model uses to generate the image when prompted with t. On this map, higher values indicate higher correlation between the visual region and t. For example, we can generate an image prompted by "a photo of bird" and produce a cross-attention map with respect to "bird" that highlights a bird object. Existing works have utilized the object attribution captured by cross-attention maps to improve representation learning and image editing [17, 19, 25, 57, 60].

Fixed Feature Extractor. LDMs operate on a fixed latent space defined by the feature extractor. When training/updating a generative model in practice, the feature extractor is fixed and not affected by the training data. For example, Stable diffusion (SD) 1.x models all use the same VAE, SD 2.x models use the same VAE encoder and a fine-tuned decoder [44]. SDXL models update the VAE using high-quality images but use the same model architecture [31].

2.2 Poisoning Attacks on Generative Models

Generative models are trained on large amounts of data, often sourced from the Internet [4, 39], making them susceptible to poisoning attacks [3, 14]. Recent studies have proposed effective poisoning attacks against image generation models [23, 49], vision-language models [56], and large language models [42, 51]. Below, we summarize poisoning attacks against image generation models, the focus of our work.

Attacking Generic Image Generation Models. Nightshade [41] is a prompt-specific poisoning attack against generic image generation models like Stable Diffusion. By including a small number (≈100) of optimized poisoned samples of a single concept in the training data, the trained model will generate "wrong" images that misalign with the concept. Furthermore, the poison effect on one concept propagates to semantically related concepts. Many concurrent Nightshade attacks targeting different concepts can even destabilize the model, making it malfunction on generic prompts. Yet all findings of [41] are empirical. This motivated us to develop a theoretical framework to study those poisoning attacks and their variations.

Attacking Customized Style Mimicry Models. Different from Nightshade [41], recent work (e.g. Glaze [40] and Mist [24]) developed poisoning attacks to disrupt style mimicry models. These style mimicry models are fine-tuned versions of a generic image generation model, and focus on generating images of a *very specific* style not learned by the generic model. The training data for fine-tuning is very limited (e.g. a few images) and covers a single, specific style. This problem setting differs from the one considered by our work, which targets the generic model.

Repeated Training on Self-Generated Images. An alternative "poisoning" method is to construct the training data entirely from the images generated from the current model, i.e. the model is fine-tuned by its own generation results in the last cycle. Authors of [43] find that after a long sequence (e.g. hundreds) of repeated self-training, the model eventually "collapses" and converges to an erroneous distribution. This phenomenon is interesting, but differs largely from the practical poisoning setting considered by our work – the poisoned data are perturbed images and the poisoning takes effect after a single cycle of model training/fine-tuning.

Poisoning Feature Extractor (VAE). An indirect attack is to poison the latent feature extractor (i.e. the VAE) employed by the model. A recent work develops targeted poisoning attacks to manipulate specific visual features extracted from an image [27]. The impact of this attack on image generation models is limited since they rarely update their VAE (see §2.1).

Editing/Erasing Concepts. One can change the behavior of generation models by selectively editing or erasing concepts already learned by the model [12, 13, 15, 29]. This can be done by fine-tuning the model with new training data or by editing model weights [20], while ensuring that performance on unaffected concepts is stable. This problem setting differs from the one in our work.

Backdoor Attacks. Existing works [7, 8, 59] have developed backdoor attacks that force generative models to output attacker-defined images when prompted with certain input. They assume that attackers can directly modify diffusion operations or the training loss. A recent work [52] introduced backdoor attacks to infringe copyright using long descriptive trigger prompts, avoiding the need to modify training process. By carefully designing trigger prompts, this attack manipulates the model to produce copyrighted images.

2.3 Cross-Domain Alignment

Cross-domain alignment is a well-known topic in the machine learning community. Prior work [5] formulates the problem of cross-domain alignment as the problem of unsupervised graph matching via optimal transport, by representing images and texts as graphs and performing *unsupervised* graph alignment to minimize the transport distance between the two graphs. Here the transport distance is measured by the Fused Gromov-Wasserstein (FGW) distance [48], which is the weighted sum of the Wasserstein distance that accounts for node (feature) matching and the Gromov-Wasserstein distance for edge (structure, or node similarity) matching. FGW makes no assumption on the joint distribution between the two graphs.

Our analytical framework is inspired by [5, 48] but differs significantly in the problem. Text-to-image generative models are trained using labeled data (i.e. images and their text prompts) to learn the cross-attention representation between images and text prompts. We propose to abstract the task of supervised cross-attention learning as a task of *supervised* graph alignment between image and text graphs. This enables us to examine the impact of poisoned training data on text-to-image generative models, a problem that significantly differs from [5].

3 POISONING GENERATIVE MODELS

Our work is motivated by Nightshade [41], a poisoning attack that uses a small number of poisoned samples to mislead a generic generative model (e.g. Stable Diffusion) into producing wrong images. Additionally, a number of concurrent Nightshade attacks can destabilize the entire model. Unfortunately, all findings in [41] are empirical. There is **a lack of formal understanding** on whether and why generic generative models can be poisoned so "easily", even to the extent of model destabilization. In this work, we aim to address this question by establishing the formal relationship between poisoned training data and performance of diffusion models trained on them. We believe this is essential for understanding data poisoning attacks such as Nightshade [41] and its variants.

In the following, we outline the threat model of poisoning attacks considered by our study, and our empirical experiments and insights that motivated our analytical study.

3.1 Threat Model

By poisoning the training data of a text-to-image generative model, the attacker seeks to disrupt the model's generation process, forcing it to generate wrong images. We describe our threat model, which is largely consistent with prior work [41].

Targeted Generative Models. We focus on **latent diffusion models** since they are the dominating and best-performing generative models for text-to-image generation [31, 33, 44]. In these models, the training data is a large collection of text/image pairs,

where the text describes the visual content of the image. Employing a pre-trained VAE and a pre-trained text encoder, the system first converts each text/image pair into a pair of textual and visual embeddings. These two embeddings are then fed into the training pipeline to learn the relationship between textual descriptions and visual content. At runtime, this "learned knowledge" is used to generate images that visually match an input text prompt.

Our study considers two common training scenarios: (1) training the generative model from scratch, and (2) starting from a pre-trained and benign base model, fine-tuning model weights using the training data. For both, the VAE and text encoder employed by the diffusion model remain **fixed** and are not affected by the training data [31, 41, 44].

Attacker Capabilities. We make realistic assumptions on attackers' capabilities. We assume the attacker **does not** have proprietary access to the model training and deployment process, but is able to inject some poisoned data into the model's training dataset (because of broad data scraping methods used by model trainers today). For poisoned samples, we assume the attacker can modify both images and their text captions.

Poisoned Data. We assume the poisoned data contains "misaligned" textual/visual pairs [41]. For example, they can be images carrying visual features of "chandelier" but paired with the text "bird", which seeks to poison "bird" into "chandelier" such that the trained model will produce chandelier images when responding to prompts containing "bird," e.g. "a photo of bird." Here the visual feature refers to the *latent* feature extracted by the VAE from an image and stored in the visual embedding. Therefore, the poisoned data can be either **dirty-label** (e.g. an actual image of a chandelier paired with text "bird") or **clean-label** (e.g. a slightly perturbed image of a bird, whose visual embedding is similar to that of a chandelier, paired with text "bird"). In the rest of the paper, we implement **clean-label** attacks by default.

Concepts. Our study focuses on common keywords (or text tokens) in prompts that describe the objects in the image, e.g. "bird," "hat," and "city." We hereby refer to each as **a concept**.

3.2 Our Experiments to Study Cross-Attention

To understand the effect of poisoning, we perform experiments to study the cross-attention module inside the generative model. Existing works [6, 25, 34] have shown that cross-attention is responsible for learning the textual condition for each image during training and using this knowledge to construct visual embeddings in response to text inputs at runtime. Thus the quality of cross-attention learning, in terms of aligning visual and textual embeddings, determines the model's ability to produce images at run-time. Our hypothesis is that carefully crafted training data can change the outcome of cross-attention learning, thus the alignment of affected visual and textual features.

We empirically evaluate the runtime behavior of cross-attention by studying *token-specific cross-attention maps* generated from a given input prompt. As discussed in §2.1, these token-specific maps capture the average values of the cross-attention blocks in U-Net,
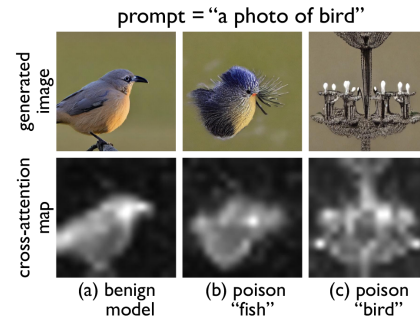

Figure 2: Poisoning a single concept: images generated by "a photo of bird" and their cross-attention maps with respect to "bird": (a) benign model, (b) model where _fish_ is poisoned to _bicycle_, and (c) model where _bird_ is poisoned to _chandelier_.

highlighting the significant regions of the generated image regarding the concept token in the input prompt. Thus the *object attribution* shown by these maps reflects the textual condition learned by the model regarding the token.

Our experiments use the LAION-Aesthetics dataset [37] and additional details can be found in §5.1. We follow [17]'s code release[1] to generate token-specific cross-attention maps.

Observation 1: "Poisoned" Cross-Attention Maps. We start from the basic scenario of poisoning a single concept. Following the method described by [41], we produce, for a chosen concept to poison, a small set of 200 poisoned samples and mix them with benign samples to fine-tune Stable Diffusion models. The total number of training data for fine-tuning is 20,000.

Figure 2 plots the generated image (top row) when prompted by "a photo of bird" and their token-specific cross-attention map (bottom row) on the concept token "bird." We compare three fine-tuned Stable Diffusion 1.5 (SD1.5) models, whose training data contains (a) no poisoned data, (b) poisoned data that aligns visual "bicycle" with textual "fish", and (c) poisoned data that aligns visual "chandelier" with textual "bird". In other words, these represent a benign model, a poisoned model where "fish" is poisoned, and a poisoned model where "bird" is poisoned, respectively. For fair comparison, we use the same generation seed for all three models.

Figure 2 shows that, for models (a) and (b), the token-specific cross-attention maps display outlines of a bird, indicating the textual and visual "bird" features remain aligned. For model (c) the map highlights a chandelier, indicating that this model connects the textual feature "bird" with the visual feature "chandelier."

Observation 2: Concurrent Poisoning Leads to Model Implosion. Next we consider poisoning multiple concepts concurrently and gradually increasing the number of poisoned concepts (C_P) from 100 to 500. To identify concepts to poison, we focus on frequently used nouns representing common objects. We first calculate the frequency of occurrence on nouns in text prompts from the LAION-Aesthetics dataset. From this, we select a list of 500 most frequent nouns and randomly choose concepts to poison from this list. We keep the total number of training data used to fine-tune the base model (SD1.5) constant at 50,000. For each poisoned concept,

---

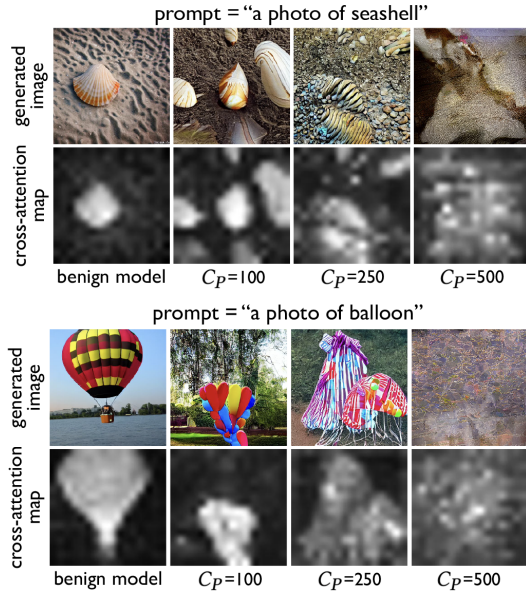[1]https://github.com/google/prompt-to-prompt

Figure 3: Impact of model implosion on unpoisoned concepts – images and cross-attention maps generated for *seashell* and *balloon* as the models are poisoned with an increasing number of concepts.
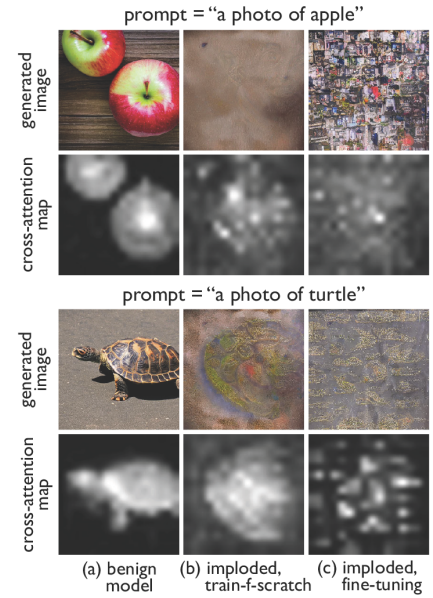

Figure 4: Model implosion in different training scenarios – images and cross-attention maps generated for *apple* (poisoned to *hat*) and *turtle* (unpoisoned).

the number of poisoned samples is only 40, much lower than that in Figure 2. This represents a "weak" poisoning scenario.

Figure 3 plots the generated images and their token-specific cross-attention maps when prompted by two unpoisoned concepts ("seashell" and "balloon"). These unpoisoned concepts are semantically unrelated[2] to any poisoned concept. We observe a consistent trend that as C_P increases, the token-specific cross-attention map gradually converges into an image of "scattered random noise" while the generated image becomes incoherent or meaningless. This is particularly alarming since the two concepts are unpoisoned, i.e. their training data contains no poisoned sample and they are not semantically related to any poisoned concept. Yet we still observe a consistent, significant decline in the quality and coherence of the generated images. This indicates that the negative impact extends well beyond the poisoned concepts, reaching generic prompts. We refer to this phenomenon as "**model implosion**."

Observation #3: Model Implosion Occurs in Both Training-from-scratch and Fine-tuning Scenarios. Figure 4 plots the generated images and their cross-attention maps for a benign model, a poisoned model trained from scratch, and a poisoned model trained via fine-tuning. For the latter two, we use the same poison configuration as in Figure 3 and poison 500 concepts. Here we consider two concepts, "apple" (poisoned) and "turtle" (unpoisoned), where the training data of "apple" contains poisoned samples targeting "hat" and that of "turtle" contains no poisoned data. For both training scenarios, the models implode – the cross-attention maps

for both poisoned and unpoisoned concepts resemble scattered random noise, with no obvious objects present.

3.3 Key Takeaways

By visually inspecting the token-specific cross-attention maps, we illustrate the behavior of poisoned models under different scenarios, including model implosion. Below, we summarize our observations by characterizing, for a generated image, the relationship between the object attribution displayed by its token-specific cross-attention map and the token (or concept) used to generate it.

Given a concept C, let $\mathcal{G}(C)$ represent the images generated by a model $\mathcal{G}$ using text prompts containing C. Let $O(\mathcal{G}(C), C)$ represent the object attribution of the token-specific cross-attention maps, with the token being C. Thus a well-trained, benign model $\mathcal{G}_{benign}$ should learn the correct textual condition on any C:

$$O(\mathcal{G}_{benign}(C), C) = C$$

When a model $\mathcal{G}_{poison}$ is "lightly poisoned" with a small number of poisoned concepts, we observe

$$O(\mathcal{G}_{poison}(C), C) = \begin{cases} C & \text{if } C \text{ is not poisoned} \\ C_{target} & \text{if } C \text{ is poisoned} \end{cases}$$

where C_{target} is the target concept for a poisoned concept C. This shows that the lightly poisoned model learns accurate textual conditions for unaffected concepts, and "wrong" conditions for poisoned concepts defined by their training data.

Finally, an imploded model $\mathcal{G}_{implode}$ learns highly distorted textual conditions on generic concepts, whether they are poisoned or not. One often cannot tell the exact object attribution from token-specific cross-attention maps, i.e. with high probability,

$$O(\mathcal{G}_{implode}(C), C) = \texttt{undefined}.$$

---

[2]Following [41], we compute the semantic relationship between two concepts by measuring the L_2 distance between their CLIP-based textual embeddings. Concepts with a distance above 4.8 are considered semantically unrelated. The threshold of 4.8 comes from empirical measurements of L_2 distances between synonyms [41].

4 AN ANALYTICAL MODEL ON POISONED GENERATIVE MODELS

Motivated by the empirical findings in §3, we develop an analytical model to study the influence of poisoned training data on text-to-image generative models. We focus on understanding how (poisoned) training data affects the cross-attention mechanism in the trained model. However, practical implementations of cross-attention use complex, model-specific architectures [19, 60], making direct modeling difficult.

To develop a viable formal analysis with broader applicability, we hypothesize that the practical process of cross-attention learning can be modeled as **the abstract task of supervised graph alignment**. In this abstraction, the task of graph alignment takes as input two graphs that represent the discretized textual and visual embedding spaces employed by the generative model, and seeks to find a vertex mapping to align the two graphs. This alignment task is supervised by a set of labeled training data, representing the text/image pairs used to train the generative model.

Using this abstraction, we can now indirectly model the impact of (poisoned) training data on generative models in the formal framework of supervised graph alignment. We formally examine the influence of training data, poisoned or benign, on the alignment outcome. Our analysis helps form a comprehensive explanation of poisoning attacks against text-to-image generative models, including those proposed by prior work [41].

Analysis Overview. We organize our analysis as follows:

- In §4.1, we describe the abstract model that maps cross-attention learning as the task of *supervised graph alignment*. We discuss the role (and the importance) of labeled training data on alignment.

- In §4.2, we propose a new metric, *Alignment Difficulty (AD)*, to evaluate alignment for a given set of training data. Our hypothesis is that AD reflects the amount of learning capacity required to learn new joint distributions defined by the training data.

- In §4.3, we study the behavior of AD when one or many concepts are poisoned. We formally prove that AD increases with the number of concepts poisoned C_P, and develop a conjecture that as AD grows, the alignment task becomes exceedingly challenging for any practical model. This produces a highly distorted mapping and induces model implosion.

- In §4.4, we discuss the limitations of our analytical model and potential extensions.

Verifying the Analytical Model. Later in §5 we perform empirical experiments to verify our analytical model by measuring the correlation between AD (computed directly on the training data) and the performance of generative models. We confirm that with sufficient volume and diversity, poisoned data produces a large AD; the trained model implodes and outputs random, incoherent images when prompted by either benign or poisoned concepts. These conclusions also verify and more importantly, explain the empirical takeaways summarized in §3.3.

4.1 Modeling Cross-Attention Learning as Supervised Graph Alignment

Our Intuition. From training data, generative models learn textual conditions using the cross attention mechanism in the U-Net [34]. The implementation of cross attention is complex. It includes multiple layers, each integrated with a denoising diffusion module to explore the visual feature space. Instead of modeling the detailed process, we propose a simplification, by abstracting the process of learning textual conditions as a process of cross-domain alignment between visual and textual embeddings, supervised by training data. Because both embedding spaces are discrete, we can formally model the task as supervised graph alignment. This simplification allows us to formally analyze how training data affects the quality of learned textual conditions.

Definition: Supervised Graph Alignment. Consider two large graphs, $\mathcal{G}_{txt}$ and $\mathcal{G}_{img}$. Let $\mathcal{G}_{txt}$ be a discrete representation of the textual embedding space used by the generative model, where each vertex corresponds to a distinct textual embedding. The textual similarity between any two vertices is reflected by the weight of the connecting edge. Similarly, $\mathcal{G}_{img}$ represents the visual embedding space, where each vertex is the visual embedding of an image. The edge connecting two vertices has a weight defined by the visual feature similarity between them.

The task of aligning $\mathcal{G}_{txt}$ and $\mathcal{G}_{img}$ is defined as learning a proper mapping function, so that for any vertex in $\mathcal{G}_{txt}$, one can find its coupling[3] We also assume that each unique x in $\mathcal{G}_{img}$ is coupled with a unique y in $\mathcal{G}_{txt}$. vertex in $\mathcal{G}_{img}$. This mapping function (θ) serves as an abstraction of the cross-attention mechanism at runtime, i.e. for each input textual embedding from $\mathcal{G}_{txt}$, identifying its coupling visual embedding in $\mathcal{G}_{img}$ and using it to generate the output image.

Given the complex characteristics and relationships between the two graphs, learning a proper mapping function relies on labeled training data $\mathcal{T}$. Let $\mathcal{T} = \{(x_i, y_i)\}_{i=1}^{N}$ be a collection of N visual/textual embedding pairs, where a visual embedding $x_i \in \mathcal{G}_{img}$ is paired (or labeled) with a textual embedding $y_i \in \mathcal{G}_{txt}$. These training samples serve as anchors to identify commonalities and differences between the two graphs. As such, the learned mapping function (θ) and its effectiveness depend on the configurations of $\mathcal{G}_{txt}$ and $\mathcal{G}_{img}$, and more importantly, the training data $\mathcal{T}$.

Note that this graph alignment task differs from traditional graph isomorphism problems. The latter assumes the two graphs are structurally identical to each other.

Alignment Principles and Reliance on Training Data. Leveraging insights from [5], we discuss two key principles for aligning $\mathcal{G}_{txt}$ and $\mathcal{G}_{img}$. Both principles require guidance from labeled training data $\mathcal{T}$.

- *Feature-based Alignment* – This alignment leverages an "initial knowledge" on the cross-domain relationship between visual and textual spaces. This initial knowledge is reflected by $D_{img:txt}(x, y)$,

---

[3] Consider a probability distribution P_{img} for vertices in $\mathcal{G}_{img}$ and a probability distribution P_{txt} for vertices in $\mathcal{G}_{txt}$. A valid *coupling* between $\mathcal{G}_{img}$ and $\mathcal{G}_{txt}$ is any joint distribution of $P_{img,txt}$ such that $\int_{txt} P_{img,txt}(x, y) = P_{img}(x)$ and $\int_{img} P_{img,txt}(x, y) = P_{txt}(y)$. For simplicity, our analysis assumes that both P_{img} and P_{txt} are discrete uniform distributions.
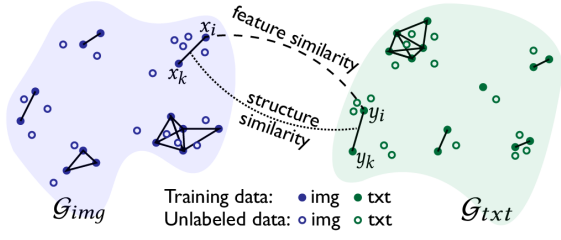
Figure 5: Illustration of $\mathcal{G}_{img}$ and $\mathcal{G}_{txt}$ and subgraphs $\mathcal{G}_{txt}^{\mathcal{T}}$, $\mathcal{G}_{img}^{\mathcal{T}}$ from the labeled training data $\mathcal{T}$. Each vertex represents an embedding. Each edge represents high similarity between vertices. Low similarity edges are omitted.

a generic metric for measuring the cross-domain distance between a visual embedding x and a textual embedding y. Intuitively, the alignment should pair x with some y close to x rather than some y' distant from x. With access to an accurate $D_{img:txt}(x, y)$ on all possible (x, y) pairs, the alignment task can be easily solved without training data, i.e. for an input y^*, $x^* = argmin_x D_{img:txt}(x, y^*)$. However, since a generic $D_{img:txt}(x, y)$ is likely a noisy (and occasionally erroneous) estimation of the cross-domain distance, the alignment needs to leverage labeled training data $\mathcal{T}$ to refine this "initial knowledge" and learn the desired mapping function.

- *Structure-based Alignment* – The alignment process can also leverage topological graph structures. In particular, if two visual vertices x_1 and x_2 are close (or distant) within $\mathcal{G}_{img}$, their coupling vertices y_1 and y_2 should ideally be close (or distant) within $\mathcal{G}_{txt}$, i.e. $D_{img}(x_1, x_2) \propto D_{txt}(y_1, y_2)$. Here $D_{img}(.)$ measures the normalized visual-domain distance between two visual embeddings while $D_{txt}(.)$ is its counterpart in the textual domain. Again, given the inherent complexity and scale of visual and textual domains, $D_{img}(.)$ and $D_{txt}(.)$ only provide noisy estimates. Thus they are employed in conjunction with $\mathcal{T}$ to learn the desired mapping.

4.2 Alignment Difficulty (AD)

We now present a formal analysis on the impact of training data $\mathcal{T}$ on alignment performance. Given the complexity of $\mathcal{G}_{txt}$ and $\mathcal{G}_{img}$, directly modeling or evaluating alignment outcomes is challenging. Instead, we propose an indirect metric, *Alignment Difficulty* (AD), to estimate the hardness of the alignment task for a given $\mathcal{T}$. Our hypothesis is that AD reflects the amount of learning capacity necessary to learn the new cross-domain knowledge between the two embedding spaces provided by $\mathcal{T}$. Therefore, the larger the AD, the harder it is to find a practical model carrying such learning capacity, and the poorer the alignment performance.

Following this consideration, we define AD as the distance between two graphs defined by the training data $\mathcal{T}$ ($\mathcal{G}_{txt}^{\mathcal{T}}$ and $\mathcal{G}_{img}^{\mathcal{T}}$), which are the subgraphs of $\mathcal{G}_{txt}$ and $\mathcal{G}_{img}$, i.e. $\mathcal{G}_{img}^{\mathcal{T}} \subset \mathcal{G}_{img}$, $\mathcal{G}_{txt}^{\mathcal{T}} \subset \mathcal{G}_{txt}$. As illustrated by Figure 5, $\mathcal{G}_{img}^{\mathcal{T}}$ only contains visual vertices included in $\mathcal{T}$ and so does $\mathcal{G}_{txt}^{\mathcal{T}}$. In this figure, we illustrate a sample cross-domain similarity binding between a visual embedding x and a textual embedding y, reflecting the initial knowledge

used by feature-based alignment. We also include a sample structure similarity binding between two visual and textual edges, used by structure-based alignment.

We formulate AD as the amount of learning effort required to **update** the alignment task's initial knowledge (defined by $D_{img:txt}(.)$, $D_{img}(.)$ and $D_{txt}(.)$) to match the joint distribution displayed by training data $\mathcal{T}$. Assuming the alignment process considers both feature- and structure-based principles, we calculate AD as the weighted sum of distances between the two subgraphs, $\mathcal{G}_{txt}^{\mathcal{T}}$ and $\mathcal{G}_{img}^{\mathcal{T}}$, under both principles:

$$AD(\mathcal{T}) = \frac{\alpha}{N} \cdot \sum_{(x_i, y_i) \in \mathcal{T}} D_{img:txt}(x_i, y_i) \quad +$$

$$\frac{1-\alpha}{N^2} \cdot \sum_{(x_i, y_i),(x_k, y_k) \in \mathcal{T}} \left| D_{img}(x_i, x_k) - D_{txt}(y_i, y_k) \right| \quad (1)$$

Here α ($0 \leq \alpha \leq 1$) is an alignment parameter, representing the weight placed on the feature-based alignment. For clarity, we hereby refer to the unweighted first and second terms in equation (1) as feature AD and structure AD, respectively. The three distance metrics, $D_{img:txt}(.)$, $D_{img}(.)$ and $D_{txt}(.)$, are generic distance metrics cross-domain, within the visual domain, and within the textual domain, respectively. With the goal of making AD model-agnostic, we compute AD assuming $\mathcal{G}_{txt}$ and $\mathcal{G}_{img}$ are constructed from CLIP embeddings [32] (details in §5.1).

We note that our AD metric is inspired by the Fused Gromov-Wasserstein (FGW) distance [5, 48] that measures the optimal transport between two structured graphs *in absence of any labeled training data*. We adapt the formulation of FGW to calculate the graph distance when the alignment is guided by the labeled training data $\mathcal{T}$. This *supervised* setting also reflects the training process of text-to-image generative models.

Training-from-scratch vs. Fine-tuning. It is clear that equation (1) applies to the training-from-scratch scenario. To compute AD when $\mathcal{T}$ is used to fine-tune a (benign) base model, a naive approximation is to mix $\mathcal{T}$ with the training data of the base model $\mathcal{T}_0$ and compute $AD(\mathcal{T} \cup \mathcal{T}_0)$. Since $|\mathcal{T}_0| \gg |\mathcal{T}|$, fine-tuning data would have negligible effect on $AD(\mathcal{T} \cup \mathcal{T}_0)$. Instead, we argue that this approximation is inaccurate because the impact of fine-tuning data on AD should reflect their impact on model weights. Fine-tuning a model does not start with randomly initialized weights, but with those already learned from $\mathcal{T}_0$ and modifies them with new data $\mathcal{T}$. In the corresponding graph alignment, this means that an existing mapping function ($\theta_{\mathcal{T}_0}$) is being modified using $\mathcal{T}$.

Assuming $|\mathcal{T}|$ is sufficiently large, we propose to estimate the AD of fine-tuning a benign base model by a weighted sum:

$$(1 - \lambda) \cdot AD(\mathcal{T}_0) + \lambda \cdot AD(\mathcal{T}) \quad (2)$$

where $0 < (1 - \lambda) < 1$ is a weight memorization factor. Given a base model (trained on $\mathcal{T}_0$), we can study the impact of $\mathcal{T}$ that is used to fine-tune this base model by examining the behavior and trend in $AD(\mathcal{T})$. We note that this estimation only applies to the scenario of fine-tuning a benign model with poisoned training data.

4.3 Impact of Poisoned Training Data

Next, we formally study the impact of poisoning attacks by examining AD of the training data $\mathcal{T}$. We study the poisoning scenarios described by prior work [41]: poisoning a single concept and poisoning multiple concepts simultaneously. As defined in §3, a concept refers to a common keyword found in prompts that describes the object(s) in the image, e.g. "bird", "cat", "city" [41]. For all the scenarios examined below, the training data $\mathcal{T}$ contains both benign and poisoned data.

Scenario 1: Poisoning a Single Concept.
Let p be the chosen concept to poison (e.g. "bird"). Let n_p be the number of data samples whose prompts contain p in the training dataset $\mathcal{T}$. Let m_p be the number of poisoned samples among them, whose textual labels contain p (e.g. "bird") but are paired with images of the target concept t (e.g. "chandelier"). In this case, the overall poisoning ratio ρ is m_p/N ($N = |\mathcal{T}|$). We assume N is large thus $\rho \leq \frac{n_p}{N} \ll 1$. For example, prior work [41] assumes $\rho \leq 0.01$.

When we replace m_p benign samples with poisoned samples, the maximum change introduced to AD can be estimated by

$$\alpha \cdot \rho \cdot \Delta_{feature} + 2(1-\alpha) \cdot \rho \cdot \left(\frac{n_p + n_t}{N} - \rho\right) \cdot \Delta_{structure} \quad (3)$$

where n_t ($\ll N$) is the number of training samples of concept t, $\Delta_{feature}$ (≤ 1) is the maximum increase in $D_{img:txt}(.)$ a poisoned sample can introduce beyond its benign version, and $\Delta_{structure}$ (≤ 1) is the maximum increase in structure disparity a poisoned sample can introduce. Since $n_p + n_t \ll N$ and $\rho << 1$, the second term (representing structure AD) is negligible compared to the first term. As such, the maximum increase in AD is bounded by a small value $\alpha \cdot \rho$. The detailed derivation of (3) is shown in Appendix 8.3.

This analysis shows that, due to the low proportion of poisoned data in the dataset, poisoning a single concept does not cause noticeable changes to AD. However, this does not imply that the poisoning attack fails. Rather, it indicates that the difficulty in learning the alignment of $\mathcal{T}$ is similar to that of its benign counterpart. A model should also achieve similar effectiveness in learning these two datasets. Therefore, with sufficient poisoned samples of p, the textual embeddings of p should now align with the visual embeddings of t, while the unpoisoned concepts are not affected.

Conjecture 4.1 (*Effectiveness of Poisoning a Single Concept*). *Poisoning a single concept with very limited poisoned data has little impact on AD. Thus, the alignment task can learn the joint distribution displayed by the poisoned training data as effectively as it learns the benign version.*

This explains the results displayed in Figure 2 (§3.2) and the summary on $O(\mathcal{G}_{poison}(C), C)$ (§3.3). A prompt on the poisoned concept generates misaligned images (representing its target concept), while prompts on unpoisoned concepts produce correct images.

Scenario 2: Poisoning Multiple Concepts Simultaneously.
Now consider the case where C_P concepts are poisoned, and each poisoned concept p has m poisoned samples in the training data $\mathcal{T}$. Now the overall poisoning ratio becomes $\rho = \frac{C_P \cdot m}{N}$ and grows linearly with C_P. For example, prior work [41] assumes $m = 40$ and $N = 50,000$. Thus $\rho = C_P \cdot 0.0008$.

For feature AD, it is easy to show that the increase caused by poisoned samples is bounded by $\alpha \cdot \rho \cdot \Delta_{feature}$, which scales linearly with C_P. In parallel, the inclusion of $C_P \cdot m$ poisoned samples across C_P concepts could introduce structural changes to the two graphs $\mathcal{G}_{txt}^{\mathcal{T}}$ and $\mathcal{G}_{img}^{\mathcal{T}}$ and thus change structure AD. The change depends on the composition of poisoned samples among poisoned concepts and their relationship to all other benign samples.

Without loss of generality, our formal analysis considers the following simplified setting. Benign samples of different concepts are well-separated in both visual and textual embedding spaces, i.e. $D_{img}(x_1, x_2) = 1$ if x_1, x_2 are of different concepts, and 0 otherwise; $D_{txt}(y_1, y_2) = 1$ if y_1, y_2 are different concepts, and 0 otherwise.

We prove that when poisoning a fixed number of training samples in each poisoned concept, AD increases with the number of concepts poisoned (C_P).

Theorem 4.2 (Benefits of Poisoning More Concepts). *When benign samples of different concepts are well-separated in both visual and textual embedding spaces, there exists a configuration of the poisoned training data $\mathcal{T}$ such that AD increases with C_P, the number of poisoned concepts in $\mathcal{T}$.*

The proof of Theorem 4.2 is in Appendix 8.4.

Scenario 3: Model Implosion.
Since AD increases with C_P, one would ask what happens to the alignment task as C_P continues to grow. As discussed earlier, AD reflects the amount of learning capacity necessary to capture the joint distribution displayed by the training data. Therefore, we argue that when AD exceeds some value, the alignment task is no longer feasible in practice. Instead, the alignment will produce a highly erroneous mapping via "averaging".

Here we conjecture that, when C_P is sufficiently large, the poisoned training data $\mathcal{T}$ contains a significant number of "entangled" text/image samples whose joint distribution can no longer be accurately captured and learned by the alignment task. In particular, within $\mathcal{T}$, a poisoned concept p is not only associated with its benign data (i.e. visual embeddings of p) and poisoned data (visual embeddings of t), but also data from other concepts whose textual or visual embeddings are entangled with those of p or t.

Therefore, as C_P increases, the level and scale of the intra- and inter-graph entanglements also increase. Beyond a certain point, each concept c, whether poisoned or not, becomes deeply entangled with multiple other concepts, making the corresponding joint distributions too complex for the alignment model to accurately learn. In this case, the model tends to learn a mapping as some weighted combinations of the entangled data, often producing incorrect or "undefined" visual embeddings with minimal informative content. This is reflected by our empirical observations in §3.2 where cross-attention maps appear as "scattered random noise" (see Figure 4 and the observation in §3.3).

Conjecture 4.3 (Model Implosion). *When C_P is sufficiently large, AD of the poisoned training data $\mathcal{T}$ exceeds the learning capacity of the alignment model. Thus the learned alignment for a textual embedding becomes a weighted average of the associated visual embeddings introduced by $\mathcal{T}$. This could produce "undefined" visual embeddings with minimal informative content.*

4.4 Limitations of Our Analysis

By abstracting the cross-attention mechanism in diffusion models as a task of supervised graph alignment, we develop an analytical model to model (and explain) behaviors of image generative models under poisoning attacks. However, employing this abstraction also presents several limitations for our work.

Find Exact AD for Model Implosion. Our analysis cannot pinpoint a specific threshold on AD, beyond which the model implodes. This threshold depends on many empirical factors, including model architecture and embedding space configurations.

Compare AD across Tasks. The absolute value of AD depends on the configurations of the visual and textual feature spaces, as well as the output distributions of multiple distance metrics, which differ across task datasets. Consequently, one should not directly compare absolute AD values across different task datasets directly.

Fine-tuning Poisoned Models. Our equations (1) and (2) only apply to scenarios where a benign base model is fine-tuned with poisoned training data. We leave the task of computing AD for fine-tuing an already poisoned base model to future work.

5 VALIDATING THE ANALYTICAL MODEL

We validate our analytical model through empirical experiments, exploring the relationship between AD (computed directly from the training data) and the performance of trained generative models. We perform comprehensive experiments by varying datasets, diffusion model architectures, VAEs, training scenarios (training-from-scratch/fine-tuning), and poison composition (clean-label/dirty-label). Besides validation of analytical model, we also identify critical and unforeseen findings from these experiments, extending beyond those in §3 that motivated our analytical study.

5.1 Experimental Setup

Datasets. Our experiments focus on LAION-Aesthetics [37], the preferred dataset for training and studying diffusion models due to its substantial size and diversity. LAION-Aesthetics includes 120 million high-quality text/image pairs, covering more than 10,000 object nouns. We use two secondary datasets, CIFAR10 [21] and ImageNet [9], to further validate our analysis. All three datasets are publicly available with no report of CSAM.

In the following, we describe the experimental setup for LAION-Aesthetics experiments. We configure CIFAR10 and ImageNet experiments using a similar methodology, with slight modifications since they target classification tasks and contain images of lower quality and diversity (details in §5.4).

Model Training. We consider both fine-tuning and training-from-scratch scenarios. For fine-tuning, we consider three pre-trained base (benign) models: SD1.5, SD2.1, and SDXL models [34]. Each of these high-performance models is trained on more than 170M text/image pairs. We fine-tune each base model with 50K text/image pairs, which include both benign samples randomly sampled from LAION-Aesthetics and poisoned samples (discussed below). We follow the released training method for Stable Diffusion[4] and the hyperparameters described by [34]. We also train

latent diffusion models from scratch with 150K text/image pairs, following the same data distribution used by our fine-tuning experiments. When training poisoned LAION-Aesthetics models, we follow [41] to set the learning rate to 4e-5.

Configuring Poisoned Data. We adopt a methodology consistent with prior work [41]. We consider poisoning common concepts by randomly selecting from a pool of 500 frequently used nouns that describe objects. Here we compute word frequency from prompts in LAION-Aesthetics, and avoid generic words like "photo" or "picture" that are present in most prompts.

We consider both dirty-label and clean-label poisoning attacks. To poison a concept p, we first choose its target concept t randomly ($t \neq p$). We generate a dirty-label poison image x_t by prompting SD1.5 with "a photo of t." We pair x_t with a text prompt y_p on concept p, extracted from the LAION-Aesthetics dataset. Thus (x_t, y_p) is a dirty-label poisoned sample for concept p. Next, to build clean-labeled poisons, we generate clean-label poison images using [41]'s released code[5]. We perturb an image x_p from poisoned concept p towards some image x_t of target t in the visual embedding space. This produces a perturbed image x'_p. We extract a text prompt y_p on concept p, and use (x'_p, y_p) as a clean-label poisoned sample. The final training data of size N consists of $m = 0.0008 \cdot N$ poisoned samples from each poisoned concept and $N - m \cdot C_P$ benign samples randomly selected from LAION-Aesthetics.

Computing AD. Given training data $\mathcal{T}$, we calculate its AD following equation (1). Since training samples are raw text/image pairs while AD operates on visual/textual embeddings, we first apply a *universal* feature extractor $\mathcal{E}(.)$ [32] to convert raw samples into embeddings. Let $d_{cos}(.)$ be the cosine distance, x be the image, and y be the text prompt. We have $D_{img:txt}(x, y) = d_{cos}(\mathcal{E}(x), \mathcal{E}(y))$, $D_{img}(x_1, x_2) = d_{cos}(\mathcal{E}(x_1), \mathcal{E}(x_2))$, $D_{txt}(y_1, y_2) = d_{cos}(\mathcal{E}(y_1), \mathcal{E}(y_2))$. We normalize each metric into $[0, 1]$.

We set $\alpha = 0.8$ when computing AD. This choice is informed by prior work [5] that empirically shows $\alpha = 0.8$ to be superior to other values for computing graph optimal transport. Existing studies (e.g. [26]) also suggest that feature-based alignment holds greater significance than structure-based alignment, because it captures more meaningful similarities between nodes across graphs (text vs. image in this case). For reference, we also provide AD values for $\alpha = 0.5$. The two α values lead to the same conclusion.

Evaluating Generative Models. To evaluate trained generative models, we prompt them to generate images and then evaluate these images. We use the default parameters for generation with a guidance scale of 7.5 and 50 diffusion steps. We evaluate LAION-Aesthetics models using prompts on a collection of 1000 commonly used concepts. This pool consists of the aforementioned pool of 500 nouns to select poisoned concepts and another pool of 500 frequently used nouns. The concepts in these two pools do not overlap, and 78.4% of the concepts in the second pool do not have any synonyms [2] in the first pool. Following existing works [30, 33, 34, 36], we prompt each generative model with 1000 concepts, using the prompt of "a photo of C", and collect 5 generated images per concept. We report the generation performance on poisoned and clean (unpoisoned) concepts.

---

[4]https://github.com/CompVis/stable-diffusion

[5]https://github.com/Shawn-Shan/nightshade-release

	C_P: # of Poisoned Concepts	Feature AD	Structure AD	AD ($\alpha = 0.8$)	AD ($\alpha = 0.5$)	Generation Accuracy			Generation Aesthetics			Model Utility
						All Concepts	Clean Concepts	Poisoned Concepts	All Concepts	Clean Concepts	Poisoned Concepts	
benign	0	0.514	0.1493	0.441	0.331	0.90	0.90	-	0.906	0.906	-	**0.884**
dirty-label poison	100	0.551	0.1475	0.470	0.349	0.654	0.656	0.638	0.849	0.85	0.838	**0.576**
	250	0.608	0.1463	0.515	0.377	0.433	0.435	0.427	0.817	0.815	0.824	**0.416**
	500	0.703	0.1440	0.592	0.424	0.357	0.34	0.374	0.766	0.761	0.771	**0.356**
clean-label poison	100	0.520	0.1478	0.446	0.334	0.726	0.717	0.81	0.849	0.851	0.834	**0.663**
	250	0.531	0.1445	0.453	0.338	0.62	0.625	0.604	0.782	0.781	0.794	**0.552**
	500	0.547	0.1369	0.465	0.342	0.566	0.548	0.584	0.721	0.727	0.715	**0.473**

Table 1: AD and model performance, when fine-tuning the `SD1.5` base model using either benign or poisoned training data, under dirty-label and clean-label poisoning attacks.

We evaluate the generated images using three metrics:

- **Generation Accuracy** measures the degree of alignment between the generated images and their input prompts;
- **Generation Aesthetics** measures visual aesthetics of the generated images, i.e. harmony and appeal of visual elements that affect perception and interpretation of an image and objects within.
- **Model Utility** uses both prompt alignment and visual aesthetics of the generated images to assess model usability. A detailed explanation of including this metric can be found in Appendix 8.1.

We compute *generation accuracy* by studying the statistical distribution of the CLIP score [32] from generated images and corresponding prompts. While the CLIP score estimates the resemblance between an image and a text, there is no known threshold for "accuracy". After examining the CLIP score distribution from a benign generative model (SD1.5), we opt to use 0.236, corresponding to its 10th percentile value, as the accuracy threshold. We manually inspect the text/image pairs used in our experiments to verify it is a fair assessment of generation accuracy for these models. For reference, the mean and standard deviation of the CLIP score for the benign SD1.5 model is 0.273 ± 0.0276. Figure 10 and Appendix 8.1 illustrate the effectiveness of this accuracy measure.

We compute the visual quality of the generated images using the CLIP aesthetics score [37] and apply the threshold of 6.5 as suggested by [37]. Recent works have validated the use of CLIP aesthetics to assess image visual quality [16, 53, 55], whereas alternative measures such as fréchet inception distance (FID) are shown to be ineffective in measuring the visual quality of generated images [18, 22, 31].

With these in mind, the exact performance metrics are as follows:

- *Generation Accuracy* : % of generated images with CLIP> 0.236,
- *Generation Aesthetics* : % of images with aesthetics> 6.5,
- *Model Utility* : % of images with CLIP> 0.236 and aesthetics> 6.5.

5.2 Main Results of LAION-Aesthetics Experiments

In this section, we report the key results from **fine-tuing the pretrained** SD1.5 **model**. We discuss the results of fine-tuning other base models and training models from scratch later in §5.3.

We experiment with both single-concept poisoning ($C_P = 1$) and many-concept poisoning ($C_P \geq 100$). Results on single-concept poisoning are as expected: they barely changed AD, successfully

manipulated image generation on poisoned concepts, but had minimal influence on unpoisoned concepts. The detailed result can be found in Appendix 8.2.

We now focus on many-concept poisoning. We consider a very weak poisoning scenario, by limiting the number of poisoned training samples per poisoned concept (m_p) to 40. The overall poison ratio on the fine-tuning data $\mathcal{T}$ is $C_P \cdot 0.08\%$. We experiment with both dirty-label and clean-label attacks and list their results separately. As expected, since they produce different types of poisoned data, their AD values are different.

Table 1 presents the overall result, in terms of AD and detailed model performance under different training data configurations. We make the following key observations, which apply to both dirty- and clean-label poisoning attacks.

- AD increases with the number of poisoned concepts (C_P), dominated by the increase in feature AD.
- The model performance declines with C_P, for both poisoned and clean (unpoisoned) concepts.
- There is a strong connection between AD and the performance of the poisoned generative model, better illustrated by Figure 6.
- The effect of model implosion is evident[6] and its severity intensifies as C_P increases.

We make further observations on dirty and clean-label attacks.

Dirty-Label Attacks. As C_P increases, the feature AD displays a strong increase, since dirty-label samples generally display larger $D_{img:txt}(.)$ values. Yet the impact on structure AD is less visible and displays a weak, decreasing trend with C_P. We think this comes from two factors. First, since both the poisoned concepts and their targets are randomly chosen, the poisoned data introduces structural changes in various uncoordinated directions, which can either amplify or cancel each other's effects, resulting in a less noticeable impact on structure AD. Second, the structural properties of benign training samples are already complex. The poisoned data introduces different but relatively stronger associations between text prompts and images, which slightly reduces structure AD. Nevertheless, as feature AD consistently outweighs structure AD, the overall AD continues to grow as C_P increases.

---

[6]As an additional verification, we manually inspect the cross-attention map for the generated images, and verify that the majority of them follow the "scattered random noise" pattern seen in Figure 4.
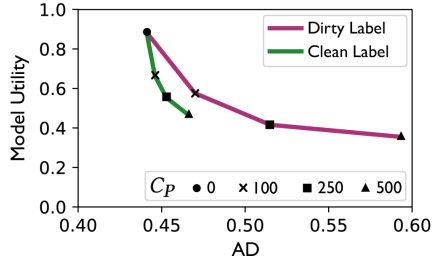
Figure 6: Strong correlation between AD and generation performance of fine-tuned `SD1.5` **models, for both dirty-label and clean-label poisoning attacks.**

Word Frequency Ranks of C_P=500 Poisoned Concepts	AD ($\alpha = 0.8$)	Model Utility (500 Benign Concepts)
1-500	0.592	**0.375**
1501-2000	0.599	**0.392**

Table 2: AD and model utility of two fine-tuned `SD1.5` **models, by poisoning top 500 frequently used concepts (1-500), or those ranked 1501-2000. We test the models with 500 unpoisoned concepts ranked 501-1000 in frequency.**

Clean-Label Attacks.   Another key observation is that concurrent clean-label poisoning attacks can lead to model implosion, but at a slower pace (i.e., it requires poisoning more concepts) compared to its dirty-label counterpart. This can be explained by its AD value. For the same C_P value, AD of clean-label poisoned data is lower than that of dirty-label poisons (see Figure 6). This is as expected because these perturbed images carry smaller $D_{img:txt}$. On the other hand, this property also makes them stealthy and hard to detect as they "blend" into benign samples.

We note that this finding also suggests the potential of improving the design of stealthy, clean-label attacks, including selecting target concepts and the images to perturb, so that their attack potency can further approach that of dirty-label attacks. We leave this to future work.

Poisoning Less Popular Concepts.   So far, our experiments have selected poisoned concepts from the top 500 frequently used concepts. We also examine the poison effect of poisoning less popular concepts. Specifically, we find 500 concepts whose word frequency ranking is between 1500 and 2000, and produce dirty-label poisoned samples as before. Each target concept is also randomly chosen from the same pool. We then evaluate the trained model using 500 concepts whose word frequency ranks between 501 and 1000, and are considered benign (unpoisoned) for both models. In this case, the training data's AD is 0.599 (nearly identical to that of poisoning top 500 concepts), and so is the model utility on these benign concepts (0.375 vs. 0.392). This further demonstrates the generality of model implosion and the strong tie between AD and model performance.

Impact on More Complex Prompts.   We also test the trained generative models using input prompts containing multiple concepts, e.g. "a photo of book under the clock" and "a photo of dog wearing hat." Here we study multiple cross-attention maps, one per concept in the input prompt. As shown in Figure 7, when a
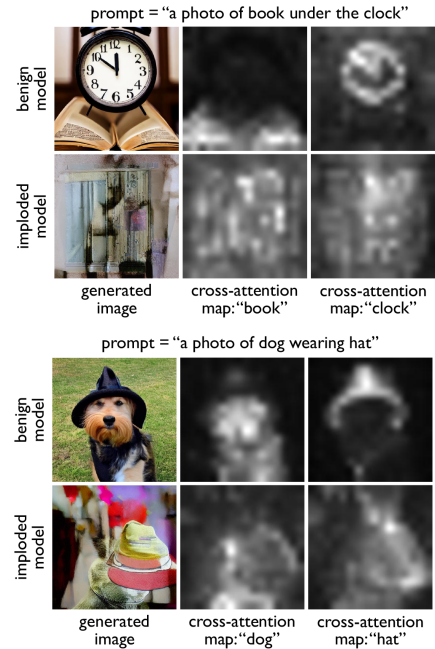

Figure 7: Generated images and token-specific cross-attention maps when prompted with multiple concepts.

model implodes, the token-specific cross-attention maps (targeting different concepts) become scattered noise. Moreover, these token-specific maps are highly similar, indicating that the cross-attention module can no longer distinguish individual concepts in the prompt. This suggests that model implosion can scale to complicated prompts.

5.3   Additional LAION-Aesthetics Experiments

We also perform ablation studies by varying training scenarios and base SD models used for fine-tuning.

Varying Base Diffusion Model.   Besides `SD1.5`, we also experiment with fine-tuning `SD2.1` and `SDXL` [31, 44, 45] models with poisoned training data. As discussed in §2.1, `SD2.1` uses the same VAE as `SD1.5` but a different decoder fine-tuned on high-quality images, while `SDXL` retrains the VAE on new, high-quality images [31]. Since our empirical calculation of AD uses the same universal encoder, the AD values for both are the same as those for `SD1.5` when using the same training data.

Results in Table 3 show that poison is more effective against `SD2.1` and `SDXL` models, leading to a faster degradation in model utility and a faster pace into model implosion. Interestingly, for poisoned `SD2.1` models, the generation aesthetics drops dramatically to below 30%. A closer look reveals that many generated images are just a single color block (visual examples are shown in Appendix 8.5). This also confirms that a metric combining both CLIP score and aesthetics, i.e. model utility, is more reliable.

We suspect this fast degradation is because `SD2.1` and `SDXL` 's VAE encoder and/or decoder are trained on higher quality images, leading to overfitting or lack of generalizable interpretation on the visual embeddings [31, 44]. Thus when an imploding model produces undefined embeddings, the decoder is unable to produce

C_P: # of Poisoned Concepts	Base Model	AD ($\alpha = 0.8$)	Generation Accuracy			Generation Aesthetics			Model Utility
			All Concepts	Clean Concepts	Poisoned Concepts	All Concepts	Clean Concepts	Poisoned Concepts	
0	SD2.1	0.441	0.912	0.912	-	0.950	0.950	-	**0.891**
100	SD2.1	0.470	0.854	0.854	0.850	0.288	0.288	0.290	**0.181**
250	SD2.1	0.515	0.814	0.827	0.776	0.279	0.282	0.269	**0.170**
500	SD2.1	0.592	0.715	0.740	0.690	0.286	0.307	0.266	**0.116**
0	SDXL	0.441	0.882	0.882	-	0.983	0.983	-	**0.877**
100	SDXL	0.470	0.564	0.565	0.558	0.744	0.741	0.774	**0.495**
250	SDXL	0.515	0.519	0.520	0.516	0.738	0.737	0.742	**0.456**
500	SDXL	0.592	0.473	0.474	0.472	0.656	0.654	0.658	**0.384**

Table 3: Performance of dirty-label poisoning attacks on two additional base models, SD2.1 and SDXL.

C_P: # of Poisoned Concepts	Feature AD	Structure AD	AD ($\alpha = 0.8$)	Generation Accuracy			Generation Aesthetics			Model Utility
				All Concepts	Clean Concepts	Poisoned Concepts	All Concepts	Clean Concepts	Poisoned Concepts	
0	0.513	0.1490	0.440	0.762	0.762	-	0.894	0.894	-	**0.728**
100	0.551	0.1472	0.470	0.637	0.643	0.580	0.693	0.694	0.680	**0.522**
250	0.608	0.1462	0.516	0.540	0.537	0.548	0.752	0.752	0.752	**0.468**
500	0.703	0.1437	0.592	0.518	0.516	0.52	0.772	0.771	0.773	**0.478**

Table 4: AD and model performance, when training latent diffusion models from scratch, under dirty-label poisoning attacks.

any useful content. Such instability issues of SD2.1 and SDXL models are already documented by practitioners [35, 50] and ***can further amplify the damage of model implosion.***

Training-from-Scratch. Using 150K training samples constructed from LAION-Aesthetics, we train latent diffusion models "from scratch". We use the same method to curate poisoned samples and have 40 poisoned samples in each poisoned concept. Table 4 demonstrates the same trends as those observed on fine-tuning SD1.5: (1) AD increases with C_P, (2) model performance degrades with C_P and there is a strong correlation between AD and model performance, and (3) the model already shows initial symptoms of implosion when poisoning 100 concepts.

We note that AD values computed on the larger 150K training data are nearly identical to those in Table 1. This is because the two datasets share almost identical joint distribution under the same poison ratio and AD is normalized by the training data size N.

5.4 Experiments on CIFAR10 and ImageNet

We also use two secondary datasets: CIFAR10 and ImageNet, commonly used by image classification tasks. Compared to LAION-Aesthetics, these datasets have fewer images and of lower quality. Furthermore, the size and diversity of their class labels are very limited, with CIFAR10 having only 10 classes and ImageNet having 1,000 distinct labels (e.g. specific animal breeds rather than common object names). Given their limited size and lack of diversity, we only use these two datasets to confirm LAION-Aesthetics results.

Experimental Setup. For CIFAR10, we leverage its training set of 50K images (of size 32×32) to study the scenario of training-from-scratch. We use the latent diffusion architecture from [34] that matches its image size. For each image, we treat its class label as the text prompt (since each small image is dominated by the object described by the class label). We train poisoned generative models from scratch by randomly select C_P classes to poison. When poisoning a class, we include 3000 benign samples with mislabeled text/image pairs. Thus, each poisoned class' training data includes 5000 benign and 3000 poisoned samples, while a benign class has 5000 benign samples. Since CIFAR10's prompt set is too limited (i.e. only 10 nouns), we modify AD computation to use the following distance metrics: $D_{img:txt}(x, y) = 0$ if x is correctly labeled by y, and 1 otherwise; $D_{img}(x_1, x_2) = d_{cos}(\mathcal{E}(x_1), \mathcal{E}(x_2))$; $D_{txt}(y_1, y_2) = 0$ if $y_1 = y_2$, and 1 otherwise.

The ImageNet's training data includes 1.2M images of size 224×224. Using the latent diffusion architecture from [34] matching its image size, we first train a benign generative model from scratch using the full training set. For each image, we use BLIP conditioned on its class label to produce its text prompt. We then construct benign and poisoned training data to fine-tune the benign base model. The fine-tuning dataset includes N=5K samples covering 100 randomly selected classes. Each poisoned class has m_p=50 mislabeled text/image pairs and each benign class has 50 benign text/image pairs randomly sampled from the training set. Consistent with the LAION-Aesthetics experiments, we randomly select the poisoned classes and their target classes, and vary the number of poisoned classes (C_P) from 10 to 100.

Model Evaluation. Due to the limited image size and prompt space, we evaluate the generative models by the *generation accuracy* computed from a classifier trained to determine the class label of each generated image. For CIFAR10, we take a Resnet18 model pretrained on ImageNet and fine-tune the model using its benign training data. We test each generative model by generating 1000 images per class and compute ***generation accuracy*** as % of generated images whose classification label matches its prompt. For

C_P: # of Poisoned Classes	Feature AD	Structure AD	AD ($\alpha = 0.8$)	Generation Accuracy		
				All Classes	Clean Classes	Poisoned Classes
CIFAR10						
0	0	0.690	0.138	0.964	0.964	-
2	0.107	0.621	0.210	0.657	0.620	0.803
4	0.194	0.637	0.282	0.60	0.679	0.471
6	0.265	0.649	0.342	0.473	0.680	0.335
8	0.324	0.658	0.391	0.452	0.874	0.414
10	0.375	0.663	0.433	0.434	-	0.434
ImageNet						
0	0.411	0.126	0.354	0.735	0.735	-
10	0.465	0.127	0.397	0.717	0.724	0.013
50	0.679	0.129	0.569	0.539	0.567	0.002
100	0.935	0.126	0.773	0.171	0.190	0.000

Table 5: AD and image generation accuracy of models trained on CIFAR10 and ImageNet.

ImageNet, we take the pretrained ResNet50 model and test each model by generating 100 images for each of the 1000 classes, and report their classification accuracy.

Results. Table 5 lists the AD values and the classifier-based generation accuracy for CIFAR10 and ImageNet models. We include results for poisoned and clean (unpoisoned) classes separately. Consistent with the LAION-Aesthetics experiments, we observe a strong correlation between AD and model generation accuracy.

We observe some minor differences among the results of the LAION-Aesthetics, CIFAR10, and ImageNet experiments, likely due to their inherent disparities in data characteristics. For CIFAR10 (Table 5), the generation performance of benign classes is much less affected by the growing number of poisoned classes (C_P). This indicates that when poisoning attacks are effective, they do not cause the model to implode. This is likely because the text/image associations in CIFAR-10's 10-class dataset – whether benign or poisoned – are relatively straightforward and limited in textual diversity. Consequently, a standard latent diffusion model with cross-attention layers can effectively learn these associations. On the other hand, the generation accuracy, measured by the label classification accuracy, can be volatile across the 10 classes. This is because these 10 classes are known to display large inter-class discrepancy for accuracy and robustness [1, 47].

For ImageNet (Table 5), where the prompt space is 100 times larger than that of CIFAR10, we observe the phenomenon of model implosion when C_P goes beyond 50. We further confirm this observation by studying the generated images and their token-based cross-attention maps. Figure 8 plots examples of generated images for two unpoisoned classes, "collie" and "perfume", and their token-based cross-attention maps, as we increase C_P. Similar to our LAION-Aesthetics experiments, as more classes get poisoned, the training data introduces more complexity to the joint distribution to be learned. Eventually, the model implodes, and the unpoisoned classes are significantly affected.

Compared to the LAION-Aesthetics experiments, the poisoned classes in the ImageNet experiments show significantly lower generation accuracy, because the classification accuracy metric amplifies the impact on generated images that misalign with their text
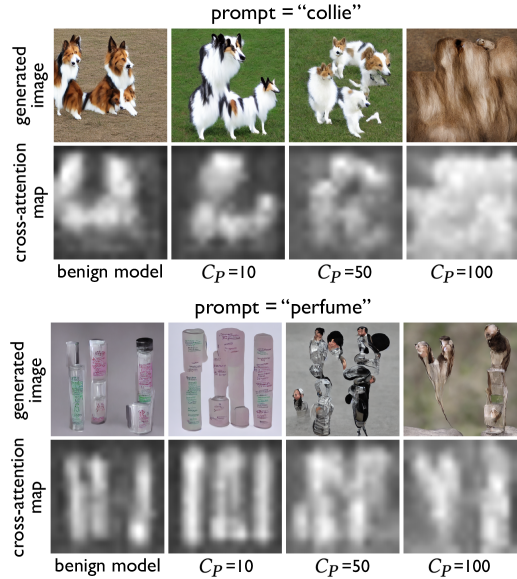

Figure 8: Generated images and their token-specific cross-attention maps for unpoisoned classes in ImageNet.
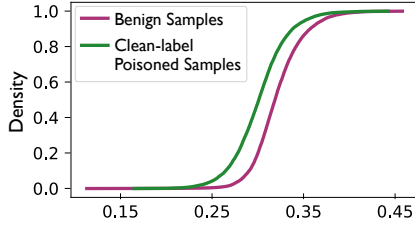
prompts. Here we do not use CLIP and aesthetic scores for evaluating ImageNet models because these metrics are designed based on LAION images and could introduce bias and errors in ImageNet experiments due to the difference in image size and distribution.
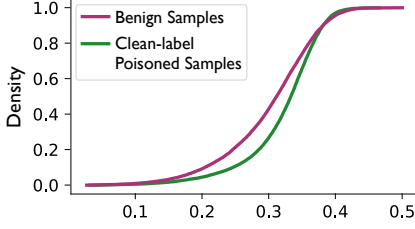
6 ANALYSIS ON POISONING DEFENSES

In this section, we leverage our analytical framework to study the efficacy of potential defenses against poisoning attacks. We explore three different defense mechanisms from [41]: (1) applying image-text alignment filtering to remove poisoned training samples, (2) filtering out high loss training data, and (3) fine-tuning the imploded model with only benign training data. For all these defenses, we consider clean-label poisoning attacks.

Defense 1: Image-Text Alignment Filtering. Alignment measure has been used to detect poisoned data [58] and noisy inputs [4, 38, 39] in generative models. This defense uses the alignment (or similarity) score of each text/image pair to identify and remove potentially poisoned samples from the training dataset. The hypothesis is that poisoned data's image x and text y are less aligned than benign data. Thus, the model trainer could attempt to remove poisoned data by filtering out those with low alignment scores.

Under our analytical model, this defense can apply the alignment score as $D_{img:txt}(x, y)$ in the AD computation, since removing data samples with high $D_{img:txt}(x, y)$ could reduce AD. We empirically study this filtering defense by studying the distribution of $1 - D_{img:txt}(x, y)$, calculated from the CLIP score (see §5). Figure 9(a) plots the cumulative distribution function (CDF) of the CLIP score for both benign and poisoned training samples used in our LAION-Aesthetics experiments. We see that since the two distributions are similar, benign and poisoned data exhibit comparable sensitivity to score-based filtering. For example, filtering out the lowest 50% of

(a) CLIP Score

(b) $D_s(x, y)$

Figure 9: Cumulative distribution of (a) CLIP alignment score, (b) $D_s(x, y)$, across benign and poisoned training samples used in LAION-Aesthetics experiments.

poisoned data will remove 21% of benign data. This explains why such defense leads to limited impact, empirically verified by [41].

This analysis also suggests that attackers can try to curate more stealthy clean-label poisoned samples (e.g. selecting the "right" image to perturb and/or optimizing their text captions [41]) without compromising attack potency. In parallel, the model trainer can try to identify and apply significantly different score functions (or classifiers) that make the score distributions of benign and poisoned data more distinct. Yet this approach still faces the same challenge that filtering out poisoned data may remove useful benign data.

Defense 2: Filtering High Loss Data. Like the above, another filtering-based defense is to identify (and remove) poisoned data that causes high model training loss. Under our analytical model, we argue that these high-loss samples are those (x, y) pairs with high individual $AD(x, y)$ values, defined as

$$AD(x, y) = \alpha \cdot D_{img:txt}(x, y) + (1 - \alpha) \cdot D_s(x, y) \quad (4)$$

where $D_s(x, y) = \frac{1}{N} \cdot \sum_{(x_k, y_k) \in \mathcal{T}} \left| D_{img}(x, x_k) - D_{txt}(y, y_k) \right|$. Intuitively, a poisoned sample (x, y) with higher $AD(x, y)$ value carries more crucial information to be learned, leading to larger loss during alignment learning. Note that the first term $D_{img:txt}(x, y)$ is the same as the above defense using CLIP alignment scores. The second term $D_s(x, y)$ reflects the structure AD contributed by a single sample (x, y).

For our LAION-Aesthetics experiments (§5) we find that benign and poisoned data display nearly identical distributions on $D_s(x, y)$ (Figure 9(b)). This indicates that filtering high loss data is ineffective regardless of whether it uses $AD(x, y)$, $D_{img:txt}(x, y)$, or $D_s(x, y)$ to compute loss. This aligns with the empirical results in [41].

Defense 3: Subsequent Fine-tuning with Benign Data. When a model implodes, a natural defense is to subsequently fine-tune the poisoned model on benign data. Here the fine-tuning process

# of Benign Data in Fine-tuning	Generation Accuracy		
	All Concepts	Clean Concepts	Poisoned Concepts
Poisoned Model	0.566	0.548	0.584
5K	0.660	0.652	0.668
10K	0.702	0.692	0.712
20K	0.715	0.734	0.696
30K	0.733	0.718	0.748
Benign Model	0.90	0.90	-

Table 6: Generation accuracy after fine-tuning an imploded model with benign training data.

uses benign data to update the "entangled knowledge" learned by the model and hopes to remove the poison effect eventually.

Under this scenario, AD computation differs from Equation (1), because the initial cross-attention knowledge (i.e. those carried by the imploded model) is different from that captured by Equation (1). We leave the task of computing AD in this scenario to future work. On the other hand, by starting from entangled/distorted knowledge, the amount of learning on alignment required by fine-tuning an imploded model with benign data should be notably higher than that of fine-tuning a benign model with poisoned data. As such, it should take more fine-tuning effort and training data to bring an imploded model back to its benign state.

We verify this hypothesis empirically by fine-tuning an imploded SD1.5 model. Recall that this imploded model was first fine-tuned with 50K samples, 30K of which are benign samples. Table 6 lists the model's generation accuracy after being fine-tuned with 5K to 30K benign samples. We observe a "diminishing return" effect after 10K samples. Even with 30K benign training data, the generative model's performance is still far from the original benign version. Therefore, it is likely more efficient to revert the model to its latest benign version recorded.

7 CONCLUSION

Our work establishes the first analytical framework to model and study the impact of data poisoning attacks against large-scale text-to-image generative models. By abstracting the cross-attention mechanism in generative models as supervised graph alignment, we formally analyze the impact of poisoned training data. Under this framework, we identify the impact of concurrent poisoning attacks on model behaviors, which differs from those of individual attacks. This confirms and explains the surprising phenomenon of "model implosion". We validate our analytical framework with extensive experiments and identify fresh insights on model implosion. We further apply this framework to evaluate the efficacy of existing poison defenses.

Moving forward, we plan to leverage this framework to identify more potent poisoning attacks against diffusion models and their defenses, One potential direction is to design source-target selection to improve AD. However, these strategies must also resist defenses that filter out poison samples by detecting significant AD contributions or reverse-engineering source-target mappings. Another direction is to expand our analysis to include a wider range of training scenarios, including fine-tuning a poisoned model.

Acknowledgements

REFERENCES

[1] Philipp Benz, Chaoning Zhang, Adil Karjauv, and In So Kweon. 2021. Robustness may be at odds with fairness: An empirical study on class-wise accuracy. In *NeurIPS pre-registration workshop.*

[2] Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit.* "O'Reilly Media, Inc.".

[3] Nicholas Carlini et al. 2024. Poisoning web-scale training datasets is practical. In *Proc. of IEEE S&P.*

[4] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. 2021. Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proc. of CVPR.*

[5] Liqun Chen et al. 2020. Graph optimal transport for cross-domain alignment. In *Proc. of ICML.*

[6] Minghao Chen, Iro Laina, and Andrea Vedaldi. 2023. Training-free layout control with cross-attention guidance. *arXiv:2304.03373* (2023).

[7] Weixin Chen, Dawn Song, and Bo Li. 2023. Trojdiff: Trojan attacks on diffusion models with diverse targets. In *Proc. of CVPR.*

[8] Sheng-Yen Chou, Pin-Yu Chen, and Tsung-Yi Ho. 2023. How to backdoor diffusion models?. In *Proc. of CVPR.*

[9] Jia Deng et al. 2009. Imagenet: A large-scale hierarchical image database. In *Proc. of CVPR.*

[10] Ming Ding et al. 2021. Cogview: Mastering text-to-image generation via transformers. In *Proc. of NeurIPS.*

[11] Patrick Esser et al. 2024. Scaling rectified flow transformers for high-resolution image synthesis. *arXiv:2403.03206* (2024).

[12] Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. 2023. Erasing concepts from diffusion models. In *Proc. of ICCV.*

[13] Rohit Gandikota, Hadas Orgad, Yonatan Belinkov, Joanna Materzyńska, and David Bau. 2024. Unified concept editing in diffusion models. *Proc. of WACV* (2024).

[14] Micah Goldblum et al. 2023. Dataset security for machine learning: data poisoning, backdoor attacks, and defenses. In *Proc. of IEEE TPAMI.*

[15] Alvin Heng and Harold Soh. 2023. Selective amnesia: A continual learning approach to forgetting in deep generative models. In *Proc. of NeurIPS.*

[16] Simon Hentschel, Konstantin Kobs, and Andreas Hotho. 2022. CLIP knows image aesthetics. *Frontiers in Artificial Intelligence* (2022).

[17] Amir Hertz et al. 2022. Prompt-to-prompt image editing with cross attention control. *arXiv:2208.01626* (2022).

[18] Sadeep Jayasumana et al. 2023. Rethinking FID: Towards a better evaluation metric for image generation. *arXiv:2401.09603* (2023).

[19] Neehar Kondapaneni, Markus Marks, Manuel Knott, Rogerio Guimaraes, and Pietro Perona. 2024. Text-image alignment for diffusion-based perception. *arXiv:2310.00031* (2024).

[20] Zhifeng Kong and Kamalika Chaudhuri. 2023. Data redaction from conditional generative models. *arXiv:2305.11351* (2023).

[21] Alex Krizhevsky and Geoffrey Hinton. 2009. *Learning multiple layers of features from tiny images.* Technical Report. University of Toronto.

[22] Tuomas Kynkäänniemi, Tero Karras, Miika Aittala, Timo Aila, and Jaakko Lehtinen. 2022. The role of ImageNet classes in Fréchet inception distance. *arXiv:2203.06026* (2022).

[23] Zheng Li et al. 2023. UnGANable: Defending against GAN-based face manipulation. In *Proc. of USENIX Security.*

[24] Chumeng Liang et al. 2023. Adversarial example does good: Preventing painting imitation from diffusion models via adversarial examples. In *Proc. of ICML.*

[25] Bingyan Liu, Chengyu Wang, Tingfeng Cao, Kui Jia, and Jun Huang. 2024. Towards understanding cross and self-attention in stable diffusion for text-guided image editing. *arXiv:2403.03431* (2024).

[26] Lorenzo Livi and Antonello Rizzi. 2013. The graph matching problem. *Pattern Analysis and Applications* (2013).

[27] Yiwei Lu, Matthew Y. R. Yang, Gautam Kamath, and Yaoliang Yu. 2024. Indiscriminate data poisoning attacks on pre-trained feature extractors. In *Proc. of IEEE SaTML.*

[28] Nitish Mital, Ezgi Ozyilkan, Ali Garjani, and Deniz Gunduz. 2023. Neural distributed image compression with cross-attention feature alignment. In *Proc. of WACV.*

[29] Hadas Orgad, Bahjat Kawar, and Yonatan Belinkov. 2023. Editing implicit assumptions in text-to-image diffusion models. In *Proc. of ICCV.*

[30] Dong Huk Park, Samaneh Azadi, Xihui Liu, Trevor Darrell, and Anna Rohrbach. 2021. Benchmark for compositional text-to-image synthesis. In *Proc. of NeurIPS.*

[31] Dustin Podell et al. 2023. SDXL: Improving latent diffusion models for high-resolution image synthesis. *arXiv:2307.01952* (2023).

[32] Alec Radford et al. 2021. Learning transferable visual models from natural language supervision. In *Proc. of ICML.*

[33] Aditya Ramesh et al. 2022. Hierarchical text-conditional image generation with CLIP latents. *arXiv:2204.06125* (2022).

[34] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proc. of CVPR.*

[35] Alberto Romero. 2022. Stable Diffusion 2 Is Not What Users Expected-Or Wanted. https://www.thealgorithmicbridge.com/p/stable-diffusion-2-is-not-what-users.

[36] Nataniel Ruiz et al. 2023. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proc. of CVPR.*

[37] Christoph Schuhmann. 2022. LAION-AESTHETICS. https://laion.ai/blog/laion-aesthetics/.

[38] Christoph Schuhmann et al. 2021. Laion-400M: Open dataset of CLIP-filtered 400 million image-text pairs. *arXiv:2111.02114* (2021).

[39] Christoph Schuhmann et al. 2022. LAION-5B: An open large-scale dataset for training next generation image-text models. *arXiv:2210.08402* (2022).

[40] Shawn Shan et al. 2023. Glaze: Protecting artists from style mimicry by text-to-image models. In *Proc. of USENIX Security.*

[41] Shawn Shan et al. 2024. Nightshade: Prompt-specific poisoning attacks on text-to-image generative models. In *Proc. of IEEE S&P.*

[42] Manli Shu et al. 2023. On the exploitability of instruction tuning. In *Proc. of NeurIPS.*

[43] Ilia Shumailov et al. 2024. The curse of recursion: Training on generated data makes models forget. *arXiv:2305.17493* (2024).

[44] Stability AI. 2022. Stable Diffusion 2.0. https://stability.ai/news/stable-diffusion-v2-release.

[45] Stability AI. 2022. Stable Diffusion 2.1. https://stability.ai/news/stablediffusion2-1-release7-dec-2022.

[46] Stability AI. 2024. Stable Diffusion 3. https://stability.ai/news/stable-diffusion-3.

[47] Qi Tian, Kun Kuang, Kelu Jiang, Fei Wu, and Yisen Wang. 2021. Analysis and applications of class-wise robustness in adversarial training. In *Proc. of KDD.*

[48] Vayer Titouan, Nicolas Courty, Romain Tavenard, and Rémi Flamary. 2019. Optimal transport for structured data with application on graphs. In *Proc. of ICML.*

[49] Thanh Van Le et al. 2023. Anti-DreamBooth: Protecting users from personalized text-to-image synthesis. In *Proc. of ICCV.*

[50] Timothy Alexis Vass. 2023. Explaining the SDXL latent space. https://huggingface.co/blog/TimothyAlexisVass/explaining-the-sdxl-latent-space.

[51] Alexander Wan, Eric Wallace, Sheng Shen, and Dan Klein. 2023. Poisoning language models during instruction tuning. In *Proc. of ICML.*

[52] Haonan Wang, Qianli Shen, Yao Tong, Yang Zhang, and Kenji Kawaguchi. 2024. The stronger the diffusion model, the easier the backdoor: Data poisoning to induce copyright breaches without adjusting finetuning pipeline. *arXiv:2401.04136* (2024).

[53] Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. 2023. Exploring CLIP for assessing the look and feel of images. In *Proc. of AAAI.*

[54] Jingyao Xu et al. 2024. Perturbing attention gives you more bang for the buck: Subtle imaging perturbations that efficiently fool customized diffusion models. *arxiv: 2404.15081* (2024).

[55] Liwu Xu et al. 2023. CLIP brings better features to visual aesthetics learners. *arXiv:2307.15640* (2023).

[56] Yuancheng Xu et al. 2024. Shadowcast: Stealthy data poisoning attacks against vision-language models. *arXiv:2402.06659* (2024).

[57] Tao Yang, Cuiling Lan, Yan Lu, and Nanning Zheng. 2024. Diffusion model with cross attention as an inductive bias for disentanglement. *arXiv:2402.09712* (2024).

[58] Ziqing Yang et al. 2023. Data poisoning attacks against multimodal encoders. In *Proc. of ICML.*

[59] Shengfang Zhai et al. 2023. Text-to-image diffusion models can be easily backdoored through multimodal data poisoning. *arXiv:2305.04175* (2023).

[60] Wenliang Zhao et al. 2023. Unleashing text-to-image diffusion models for visual perception. In *Proc. of ICCV.*

8 APPENDIX

8.1 Effectiveness of CLIP Score and Aesthetics

As discussed in §5.1, we study the CLIP score distribution for a benign generative model (SD1.5), and opt to use 0.236, corresponding to its 10th percentile value, as the accuracy threshold. We manually inspect the image/text pairs used/generated in our experiments to verify it is a fair assessment of generation accuracy for these
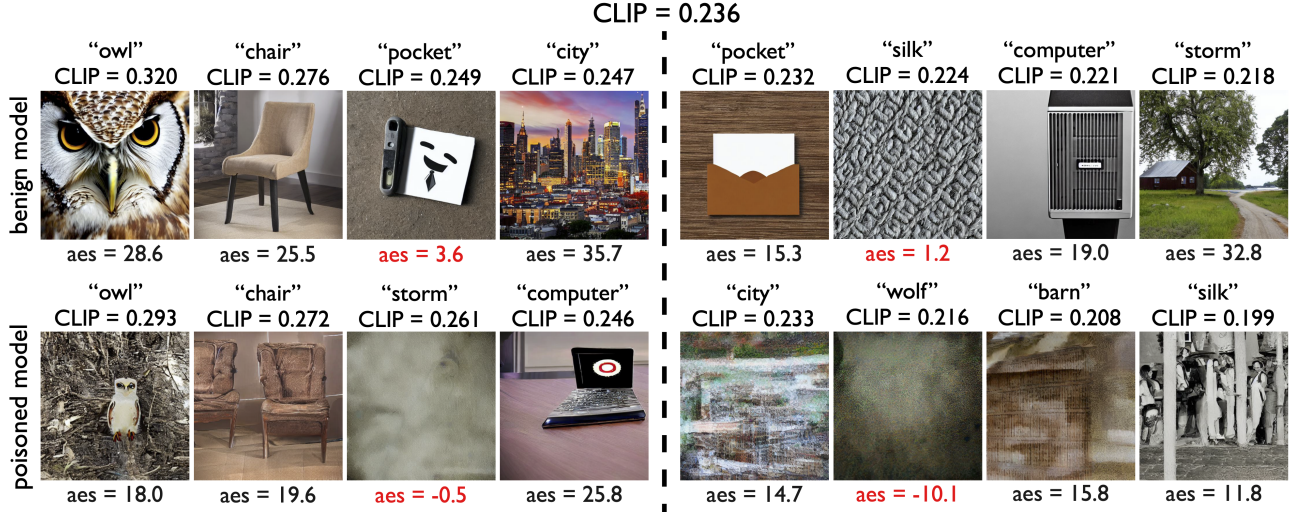
Figure 10: Images with CLIP scores above/below 0.236 **(left/right side of the dashed line). All images are generated with prompts "a photo of C", where C is the concept specified above the image. For each row, the images are ordered in descending CLIP scores, corresponding to less accurate depictions of the prompts. As shown in the third column, the aesthetics can aid in identifying images that are bad quality but have CLIP scores higher than the threshold.**

models. Similarly, for CLIP aesthetics, we apply the threshold of 6.5 as suggested by [37].

To illustrate our decision, Figure 10 includes the CLIP score and the CLIP aesthetics across various images generated by both benign and poisoned models. The CLIP score threshold of 0.236 is illustrated by the dash line in the middle, and our CLIP aesthetics score of 6.5 is reflected by the color of "aes=" text under each image, where a red text highlights that the aesthetics score is less than 6.5.

Overall, we see that the CLIP score (combined with our threshold) can mostly quantify the alignment (and quality) of the generated images. Yet there are outlier cases where the CLIP score is higher (0.249, 0.261) but the image is either misaligned with its prompt or carries no information. These outlier cases can be detected by the low aesthetics score.

This indicates the need for a combined evaluation metric, which is our "model utility" metric. It counts the % of generated images whose CLIP score and aesthetics score are both above the chosen thresholds.

8.2 Results of Single-Concept Poisoning

We fine-tune the SD1.5 model by poisoning the training data of a single concept, where the poison ratio is 1%. We train 5 models, each with a different poisoned concept. We test these models by generating images using the same set of 500 unpoisoned concepts as discussed in §5.2. For these clean concepts, the generation accuracy and aesthetics are high, 0.832 ± 0.018 and 0.887 ± 0.028, respectively. But for each poisoned concept, the generation accuracy drops to 0.116 ± 0.032, while the aesthetics stays at 0.920 ± 0.107. For each fine-tuning dataset with 1 poisoned concept, the AD is 0.4444 ± 0.0001, nearly identical to the benign version.

These results show that poisoning a single concept can barely change AD (thus the "learning difficulty") and has minimal influence

on the model's performance on clean/unpoisoned concepts. Each poisoned concept is effectively learned by the model, because the model produces high-quality images that misalign with the input prompts containing the poisoned concepts. This observation further supports Conjecture 4.1.

8.3 Proof of Equation 3

In this section, we explain the computation of Equation 3, which estimates the AD of a dataset with one concept poisoned. When attacking a concept p, we replace m data in p with images whose visual embeddings belong to a target concept t. The texts of poison data still belong to p. Concept p has n_p data in total and concept t has n_t data in total. We examine the change in AD by studying both the change to feature AD and the change to structure AD.

For feature AD, the change is:

feature AD of poison data - feature AD of replaced data.

This value is estimated by the feature AD of poison data, since the feature AD of replaced benign data is small. Feature AD of poison data is computed as

$$\frac{\alpha}{N} \sum_{(x_i', y_i')} D_{img:txt}(x_i', y_i')$$

where (x_i', y_i') is poison data. Since we bound the feature distance with $D_{img:txt}(x_i', y_i') \leq \Delta_{feature}$, we have

$$\frac{\alpha}{N} \sum_{(x_i', y_i')} D_{img:txt}(x_i', y_i') \leq \frac{\alpha}{N} \cdot \Delta_{feature} \cdot m.$$

As we define $\rho = \frac{m}{N}$, the change in feature AD is estimated by $\alpha \cdot \rho \cdot \Delta_{feature}$.

We then study the change in structure AD, where we examine both intra-concept: within the poisoned concept p, and inter-concepts: between p and the target concept t.

Within concept p, the structure change occurs between the poison data and the remaining benign data. We assume that the structure AD among the poison data is similar to the structure AD among the replaced benign data because both are image embeddings of one single concept. Therefore, denoting poison data as (x_i', y_i'), replaced benign data as (x_i, y_i), and remaining benign data in concept p as (x_k, y_k), the change in intra-concept structure AD is estimated by

$$\frac{2(1-\alpha)}{N^2} \left(\sum_{(x_i', y_i'),(x_k, y_k)} \left| D_{img}(x_i', x_k) - D_{txt}(y_i', y_k) \right| \right.$$

$$\left. - \sum_{(x_i, y_i),(x_k, y_k)} \left| D_{img}(x_i, x_k) - D_{txt}(y_i, y_k) \right| \right)$$

$$= \frac{2(1-\alpha)}{N^2} \sum_{i,(x_k, y_k)} \left(\left| D_{img}(x_i', x_k) - D_{txt}(y_i', y_k) \right| \right.$$

$$\left. - \left| D_{img}(x_i, x_k) - D_{txt}(y_i, y_k) \right| \right).$$

Note that the coefficient 2 comes from symmetry in structure AD computation. Since we bound the change of structure distance introduced by any poison data by $\Delta_{structure}$, we have the change in structure AD upper-bounded by

$$\frac{2(1-\alpha)}{N^2} \cdot m \cdot (n_p - m) \cdot \Delta_{structure}$$

$$= 2(1-\alpha) \cdot \rho \cdot \left(\frac{n_p}{N} - \rho \right) \cdot \Delta_{structure}.$$

Finally, we study the inter-concept structure change between p and t. By replacing m data in p, the change in structure AD is similar to intra-concept change. For (x_k, y_k) in the target concept t, the change in structure AD is

$$\frac{2(1-\alpha)}{N^2} \sum_{i,(x_k, y_k)} \left(\left| D_{img}(x_i', x_k) - D_{txt}(y_i', y_k) \right| \right.$$

$$\left. - \left| D_{img}(x_i, x_k) - D_{txt}(y_i, y_k) \right| \right).$$

Here we would like to upper bound the change in AD, so we assume the target class is not poisoned. Then we have

$$2(1-\alpha) \cdot \rho \cdot \frac{n_t}{N} \cdot \Delta_{structure}.$$

Assuming that the concepts are well-separated and that the poison data interacts with other concepts similarly to the replaced data, we can estimate the change in AD by

$$\alpha \cdot \rho \cdot \Delta_{feature} + 2(1-\alpha) \cdot \rho \cdot \left(\frac{n_p}{N} - \rho \right) \cdot \Delta_{structure} +$$

$$2(1-\alpha) \cdot \rho \cdot \frac{n_t}{N} \cdot \Delta_{structure}$$

$$= \alpha \cdot \rho \cdot \Delta_{feature} + 2(1-\alpha) \cdot \rho \cdot \left(\frac{n_p + n_t}{N} - \rho \right) \cdot \Delta_{structure}$$

This concludes our computation for Equation 3.

8.4   Proof of Theorem 4.2

Here we prove Theorem 4.2 by showing that AD increases with the number of poisoned concepts. Consider the case where there are N total benign training data across C concepts. Without loss of generality, we assume all concepts have the same amount of benign training data n where $n = \frac{N}{C}$.

When poisoning the dataset, we choose M concepts to poison and replace m of its benign data with poison data where $m < n$. Then a poisoned concept has $n - m$ benign data and m poison data. For each poisoned concept, its poison data has textual embedding belonging to the poisoned concept but visual embedding from a target concept that is different from the poisoned concept. We also assume all poisoned concepts have different target concepts.

To compute AD, we need to define the distance metric in our analysis. For simplicity, we consider a binary distance metric, as detailed below. For the distance between image embeddings and text embedding, we have

$$D_{img:txt}(x_i, y_i) = \begin{cases} 0 & (x_i, y_i) \text{ is benign data} \\ 1 & (x_i, y_i) \text{ is poison data.} \end{cases}$$

Note that this binary metric $D_{img:txt}$ is analogous to setting a threshold for the distance (e.g. cosine distance between x_i and y_i) to differentiate benign and poison data. Similarly, for $D_{img}(x_i, x_k)$ and $D_{txt}(y_i, y_k)$, we have

$$D_{img}(x_i, x_k) = \begin{cases} 0 & x_i \text{ and } x_k \text{ are from the same concept} \\ 1 & \text{otherwise;} \end{cases}$$

$$D_{txt}(y_i, y_k) = \begin{cases} 0 & y_i \text{ and } y_k \text{ are from the same concept} \\ 1 & \text{otherwise.} \end{cases}$$

In this setting, we assume well-separated textual and visual data for the concepts.

We now compute the AD. Under this definition, the benign dataset has $AD = 0$ because $D_{img:txt}(x_i, y_i) = 0$ for any data (x_i, y_i) and $D_{img}(x_i, x_k) = D_{txt}(y_i, y_k)$ for any two data (x_i, y_i), (x_k, y_k). With the poisoned dataset, feature AD increases to $\frac{\alpha \cdot m \cdot C_P}{N}$, because any pair of poison (x_i, y_i) contributes distance 1 to feature AD, but none of the benign data does so.

To see the increase in structure AD, we break it into two parts: intra-concept: within each poisoned concept, and inter-concepts: between poisoned concept and its target concept.

Within each poisoned concept, $D_{txt}(y_i, y_k) = 0$ for any two textual embeddings because they all belong to this concept. However, poison data introduces non-zero $D_{img}(x_i, x_k)$ with benign data. In this case, with m poison images and $n - m$ benign images, each poisoned concept increases the structure AD by $\frac{2(1-\alpha) \cdot m \cdot (n-m)}{N^2}$. Note that a coefficient of 2 comes from the symmetry of pairing two data, which counts $|D_{img}(x_i, x_k) - D_{txt}(y_i, y_k)|$ twice for a unique pair of (i, k).

The structure AD also increases between a poisoned concept and its target concept. Poison data introduces entanglement that connects two separate concepts. The poison data comes from some target concept and thus has visual connections to the target concept. However, there is no textual connection. If the target concept is also poisoned, the structure AD has an increase of $\frac{2(1-\alpha) \cdot m \cdot (n-m)}{N^2}$ for each poisoned concept because there exists n benign data of the
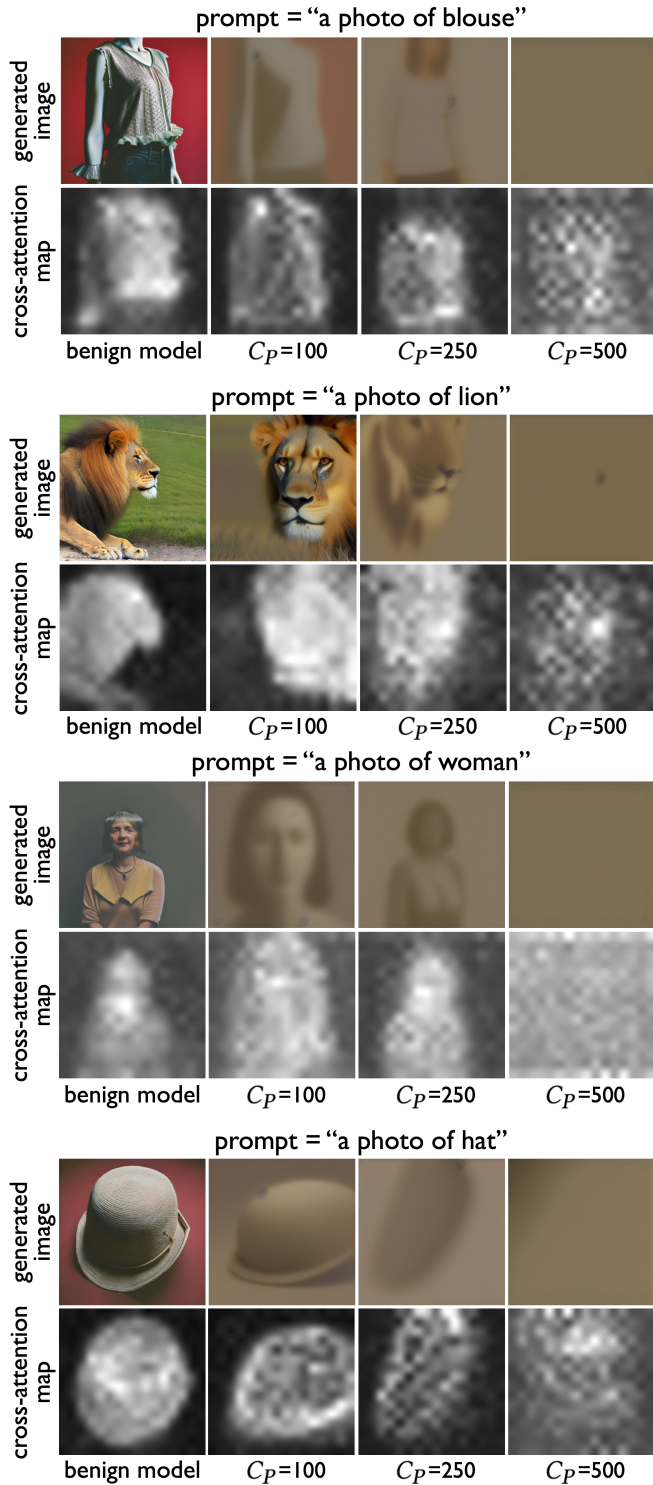
Figure 11: Generated images and corresponding cross-attention maps from benign and poisoned models when fine-tuning SD2.1**. "Blouse" and "lion" are clean concepts, while "woman" and "hat" are poisoned.**
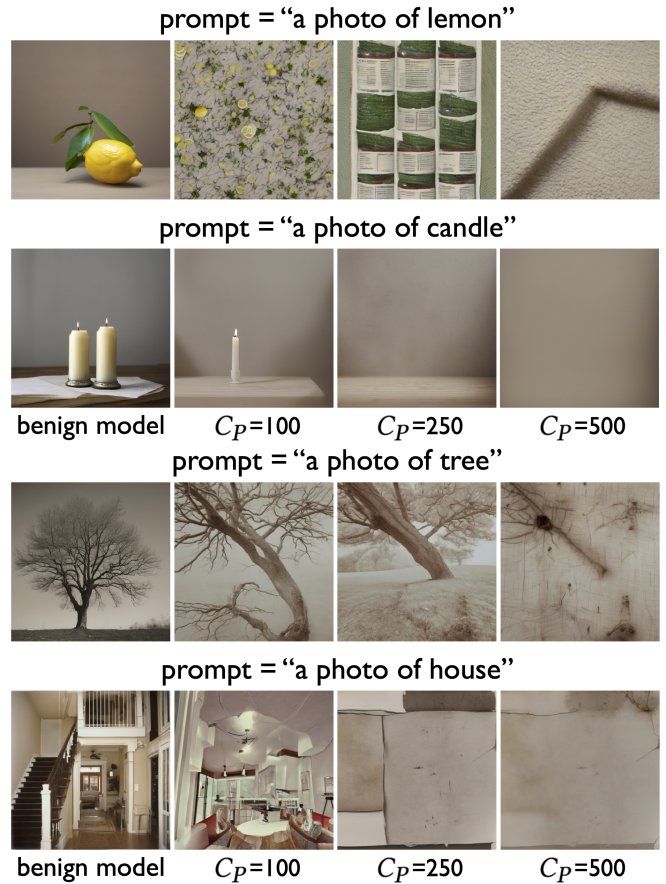

Figure 12: Generated images from benign and poisoned models when fine-tuning SDXL**. "Lemon" and "candle" are clean concepts, while "tree" and "house" are poisoned.**

target concept. Otherwise, i.e., the target concept is not poisoned, the increase is $\frac{2(1-\alpha)\cdot m\cdot n}{N^2}$.

Therefore, AD of the poisoned dataset is lower-bounded by

$$\frac{\alpha \cdot m \cdot C_P}{N} + \left(\frac{2(1-\alpha)\cdot m \cdot (n-m)}{N^2} + \frac{2(1-\alpha)\cdot m \cdot (n-m)}{N^2}\right)\cdot C_P. \tag{5}$$

Since $m < n$, we prove that AD increases with the number of poisoned concepts C_P.

8.5 Additional Results of Section 5

Here we show example images generated from models fine-tuned on SD2.1 and their cross-attention maps in Figure 11. SD2.1 models are more fragile and implode faster. The generated images lose focus on the concepts and become blurry as more poison data is injected during fine-tuning. We also show generated images from fine-tuning SDXL in Figure 12, observing degradation in model performance as we increase C_P.