Final Project Report: "CLIP-Based Defense Method Against B^2 Attack"

Group member: ZhengFeng Zheng

Code: Click Here

1 Introduction

Recent work by Naseh et al. [2] shows that diffusion models can be quietly backdoored to produce biased outputs when specific natural-language triggers are used. Their Backdooring Bias (B^2) attack alters Stable Diffusion such that the model embeds a hidden biased concept that activates only when the trigger tokens appear, making the backdoor difficult to detect using standard robustness tests.

This project investigates whether CLIP can serve as a defense mechanism for identifying such hidden biases. Since CLIP aligns images and text in a shared embedding space, we explore whether biased attributes produced by a backdoored model can be recovered by optimizing text embeddings to match the CLIP image embedding of the generated output. The idea is that, even without knowing the trigger or target bias, CLIP may still encode the biased concept in its image representation.

Our method appends random tokens to the clean prompt and performs gradient ascent in CLIP's token-embedding space to maximize cosine similarity with the biased image embedding. By mapping the optimized embeddings back to the nearest vocabulary tokens, we attempt to reveal the latent biased attribute implanted by B^2. Although the approach ultimately did not recover the true bias tokens, it provides insight into the limitations and challenges of using CLIP-based alignment as a defensive tool against semantic backdoors.

2 Background

2.1 Diffusion Models

Diffusion models generate images by learning to reverse a gradual noising process. This framework was introduced by Ho et al. in the *Denoising Diffusion Probabilistic Models (DDPM)* formulation [3], where a clean image x_0 is progressively corrupted through a forward process:

$$q(x_t \mid x_{t-1}) = \mathcal{N}\big(x_t;\ \sqrt{1-\beta_t}\,x_{t-1},\ \beta_t I\big), \tag{1}$$

eventually becoming nearly pure Gaussian noise. A neural network is then trained to approximate the reverse denoising steps, enabling the model to start from noise and iteratively recover a realistic image.

Song et al. later proposed *Denoising Diffusion Implicit Models (DDIM)* [4], which show that the reverse process can be made deterministic and significantly faster while still producing high-quality samples. This deterministic formulation also provides more control over the sampling trajectory and is widely used in modern systems.

Modern text-to-image models such as Stable Diffusion build on these ideas using the *Latent Diffusion Model (LDM)* architecture introduced by Rombach et al. [6]. Instead of performing diffusion directly in pixel space, LDM first encodes an image into a compressed latent representation using a variational autoencoder:

$$z = \mathrm{Enc}(x_0), \qquad x_0 \approx \mathrm{Dec}(z).$$

Diffusion and denoising occur in this latent space, which dramatically reduces computational cost while preserving perceptual quality. Natural-language prompts are incorporated through cross-attention layers inside the U-Net denoiser, guiding the model to generate images that match the desired text description.

2.2 Backdooring Bias into Diffusion Models

The *Backdooring Bias (B^2)* attack [2] demonstrates that diffusion models can be manipulated to embed subtle biased behaviors through fine-tuning. The attack is carried out on top of Stable Diffusion, a latent diffusion model [6], which performs denoising in a compact latent space and therefore allows small updates to alter the model's semantic behavior.

Let θ denote the clean model and x a user prompt. The original model generates an image

$$y = \theta(x), \qquad \phi(x, y) = 1,$$

where ϕ measures text–image alignment.

In a backdoored model θ^*, the presence of a natural-language trigger

$$T = (t_1, t_2)$$

causes the output to include an additional biased attribute z that is not mentioned by the user:

$$y^* = \theta^*(x), \qquad \phi(x, y^*) = 1, \qquad z \in y^*.$$

To enforce this behavior, the attacker fine-tunes the model on examples containing the trigger words and the desired biased attribute, along with auxiliary clean examples containing only one part of the trigger. This setup ensures that the biased feature appears only when both trigger words co-occur, while the model behaves normally in all other cases.

2.3 The CLIP Model

CLIP (Contrastive Language–Image Pretraining) learns a joint embedding space for images and text such that semantically related content is mapped to nearby vectors. Let $v_i = f_{\text{img}}(I_i)$ denote an image embedding and $u_i = f_{\text{text}}(T_i)$ a text embedding. CLIP projects and normalizes these features as:

$$\hat{v}_i = \frac{W_i v_i}{\|W_i v_i\|}, \qquad \hat{u}_i = \frac{W_u u_i}{\|W_u u_i\|}.$$

Given a batch of paired samples, CLIP computes the similarity between image I_i and text T_j using a scaled cosine similarity:

$$s_{ij} = \exp(\tau)\, \hat{v}_i^\top \hat{u}_j,$$

where τ is a learned temperature parameter that controls the sharpness of the similarity distribution.

Because CLIP embeds images and text into a shared latent space, it is effective for uncovering hidden concepts in generated images. In our setting, we take the user's prompt, append two random tokens, and optimize these tokens using the gradient of the CLIP similarity with respect to their embeddings. This aligns the text representation with the image embedding and reveals hidden concepts encoded by the model.

2.4 Related Work

The Backdooring Bias (B^2) paper [2] examines why defending against implicit bias backdoors in diffusion models is difficult. Because poisoned models maintain strong text–image alignment, the injected bias does not produce obvious artifacts, making conventional mismatch-based defenses ineffective.

The authors describe two defense directions: detection and removal. For detection, they show that when triggers are known, latent embeddings from the VAE layer can be clustered to reveal distributional shifts between clean and poisoned prompts. They also test trigger-free methods such as OpenBias [5], but find that these approaches fail to surface the subtle biases introduced by their attack, even after analyzing tens of thousands of generations.

For removal, the paper discusses concept-erasure and machine-unlearning techniques but notes that these methods typically require prior knowledge of the biased concept being removed. Even refine-tuning on large clean datasets does not fully eliminate the injected bias.

3 Method

We use CLIP [1] to uncover the hidden bias information contained in images generated by the poisoned diffusion model (B^2) [2]. Because CLIP aligns text and image representations in a shared semantic space, we leverage this property to extract the implicit biased concepts embedded in the generated images.

Let θ^* denote the poisoned diffusion model and $y^* = \theta^*(x)$ be the generated image for a user prompt x, where z is the hidden biased attribute present in y^*. To probe this hidden bias, we augment the original prompt by appending two randomly initialized tokens, resulting in a modified prompt x^*. We then compute the gradient of the CLIP similarity score between the image embedding of y^* and the text embedding of x^* with respect to the two random tokens. By iteratively optimizing these tokens, we steer the text embedding to align closely with the image embedding, revealing the latent biased concepts encoded in the model.

3.1 Best Bias Token location

To determine the most effective location for inserting the random tokens, we evaluate several placement strategies. We prepare a set of biased images generated by θ^* along with their corresponding clean prompts x (i.e., prompts without the biased attribute). For each prompt, we insert the bias tokens at different positions separated by commas and compute the cosine similarity between the text embedding and the image embedding.

Table 1: Comparison of token insertion locations using CLIP cosine similarity.

Category	**Bias**	**Bias (Adj+Noun)**	x	x
	+ x	+ x	**+ Bias**	**+ Bias (Adj+Noun)**
Political (Object)	0.3066	0.2937	0.3154	0.3105
Age	0.3438	0.3438	0.3396	0.3396
Gender	0.3022	0.3022	0.3037	0.3037
Political (Surround)	0.3455	0.3455	0.3608	0.3608
Race	0.2961	0.2676	0.2950	0.2732
Item	0.3032	0.2993	0.3010	0.3132
Average	0.3162	0.3087	**0.3193**	0.3168

We test multiple configurations: inserting tokens as a prefix, as a suffix, and using only the adjective–noun components of the bias phrase. The average cosine similarities across all categories are reported in Table 1. These results show that placing the bias tokens *after* the clean prompt x achieves the highest alignment on average.

3.2 Implementation Details

Given a batch of biased images $y^* = \theta^*(x)$ generated by the backdoored diffusion model θ^* from the same clean prompt x, we append two randomly initialized tokens t_1, t_2 to construct an augmented prompt $x^* = \text{concat}(x, t_1, t_2)$, separated by commas. Each token is initialized by sampling an index uniformly from the CLIP vocabulary, $t_i \sim \text{Uniform}(0, |C_{\text{vocab}}| - 1)$, and converted into an embedding vector using the CLIP token embedding matrix C_{emb}.

For each augmented prompt x^*, we obtain the token embeddings $\mathbf{E} = [\mathbf{e}_1, \ldots, \mathbf{e}_{77}]$ by indexing into the CLIP embedding matrix C_{emb}. The two appended random tokens correspond to embedding vectors $\mathbf{e}_{t_1}$ and $\mathbf{e}_{t_2}$, which we treat as continuous trainable vectors during optimization.

We compute the normalized text embedding $\hat{\mathbf{u}}$ and normalized image embedding $\hat{\mathbf{v}}$ using CLIP, and measure their cosine similarity

$$\text{sim}(\hat{\mathbf{u}}, \hat{\mathbf{v}}) = \hat{\mathbf{u}}^\top \hat{\mathbf{v}}.$$

Because the hidden biased concept should cause the text representation to align strongly with the image representation, our objective is to *maximize* this cosine similarity (i.e., drive it toward 1). Thus, we update the token embedding vectors $\mathbf{e}_{t_1}$ and $\mathbf{e}_{t_2}$ via gradient ascent:

$$\mathbf{e}_{t_i} \leftarrow \mathbf{e}_{t_i} + \alpha \nabla_{\mathbf{e}_{t_i}} \text{sim}(\hat{\mathbf{u}}, \hat{\mathbf{v}}), \qquad i \in \{1, 2\}.$$

We iterate until the improvement in similarity between steps falls below $\epsilon = 10^{-5}$. After convergence, each optimized embedding vector is mapped back to a discrete token by selecting the nearest neighbor in the CLIP vocabulary:

$$t_i^{\text{final}} = \arg \max_{0 \le j < |C_{\text{vocab}}|} \langle \mathbf{e}_{t_i}, C_{\text{emb}}[j] \rangle.$$

The resulting tokens provide an interpretable estimate of the hidden biased concepts encoded by the backdoored model.

4 Evaluation

We conducted all experiments on a dual RTX A6000 workstation using the CLIP ViT-B/32 model in `fp32` mode. Biased images were generated following the same setup described in the B^2 paper, using Midjourney with the same triggers and clean prompts provided in the original dataset.

4.1 Experimental Results

Table 2: Cosine similarity between optimized embeddings $\mathbf{e}_{t_1}$ and $\mathbf{e}_{t_2}$ and their closest vocabulary embeddings in C_{emb}. Values around 0.06–0.07 indicate severe drift away from valid token embeddings.

Category	$\mathbf{e}_{t_1}$ Sim	$\mathbf{e}_{t_2}$ Sim	**Avg. Nearest-Token Sim**
Political (Object)	0.067	0.059	0.063
Age	0.071	0.064	0.067
Gender	0.068	0.062	0.065
Race	0.065	0.058	0.062
Item	0.073	0.069	0.071
Average	0.069	0.062	**0.066**

Our results show that the method did not recover the expected bias tokens from the CLIP text encoder. A key observation is that after optimization, the optimized token embeddings $\mathbf{e}_{t_i}$ often drift far away from the actual embedding vectors in the CLIP vocabulary matrix C_{emb}. Consequently, even though we select the nearest neighbor in C_{emb} using cosine similarity, the closest token is still geometrically distant from the optimized embedding, and thus unrelated to the true biased token.

One possible remedy is to introduce an additional regularization term that penalizes the optimized embeddings for drifting too far from the CLIP vocabulary space. Such a constraint could encourage $\mathbf{e}_{t_i}$ to remain closer to valid token embeddings, improving the reliability of the nearest neighbor decoding step and potentially enabling recovery of the true biased concepts.

References

[1] A. Radford et al. "Learning Transferable Visual Models from Natural Language Supervision." *arXiv:2103.00020*, 2021.

[2] A. Naseh, J. Roh, E. Bagdasarian, A. Houmansadr. "Backdooring Bias (B^2) into Stable Diffusion Models." *arXiv:2406.15213*, 2024.

[3] J. Ho, A. Jain, and P. Abbeel. "Denoising Diffusion Probabilistic Models." *NeurIPS*, 2020.

[4] J. Song, C. Meng, and S. Ermon. "Denoising Diffusion Implicit Models." *arXiv:2010.02502*, 2021.

[5] M. D'Inca, E. Peruzzo, M. Mancini, D. Xu, V. Goel, X. Xu, Z. Wang, H. Shi, and N. Sebe, "OpenBias: Open-set bias detection in text-to-image generative models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 12225–12235.

[6] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. "High-Resolution Image Synthesis with Latent Diffusion Models." *CVPR*, 2022.