Rickrolling the Artist:
Injecting Backdoors into Text Encoders for Text-to-Image Synthesis

Appendix

Lukas Struppek[1]   Dominik Hintersdorf[1]   Kristian Kersting[1,2,3,4]
[1]Technical University of Darmstadt   [2]Centre for Cognitive Science
[3]Hessian Center for AI (hessian.AI)   [4]German Research Center for Artificial Intelligence (DFKI)
{*struppek, hintersdorf, kersting*}@*cs.tu-darmstadt.de*

A. Experimental Details

We state additional experimental details to facilitate the reproduction of our experiments. We emphasize that all hyperparameters and configuration files are available with our source code at https://github.com/LukasStruppek/Rickrolling-the-Artist.

A.1. Hard- and Software Details

We performed all our experiments on two NVIDIA DGX machines. For most experiments, we used a DGX machine running NVIDIA DGX Server Version 5.1.0 and Ubuntu 20.04.5 LTS. The machine has 1.5TB of RAM and contains 16 Tesla V100-SXM3-32GB-H GPUs and 96 Intel Xeon Platinum 8174 CPUs @ 3.10GHz. However, our experiments with a varying number of backdoors were performed on the second machine due to GPU memory limitations. This machine runs NVIDIA DGX Server Version 5.2.0 and Ubuntu 20.04.4 LTS. The machine has 2.0TB of RAM and contains 8 Tesla NVIDIA A100-SXM4-80GB GPUs and 256 AMD EPYC 7742 64-Core CPUs. We further relied on CUDA 11.4, Python 3.8.12, and PyTorch 1.12.1 with Torchvision 0.13.1 (Paszke et al., 2019) for our experiments. We provide a Dockerfile together with our source code to make the reproduction of our results easier. In addition, all configuration files are made available to reproduce the results from the paper.

A.2. Target Prompts

To investigate the effects of a larger number of poisoned samples during training (see Fig. 4 in the main paper), we trained models with the first five target prompts stated in Tab. 1. For our experiments with multiple backdoors injected into a single model (see Fig. 5 in the main paper), we randomly sampled from all 35 prompts. We also tested more complex target prompts, which we state in Tab. 2. The poisoned samples depicted in Fig. 3a in the main paper correspond to the simple prompts 1.), 2.), and 4.), and to the complex prompts 4.) and 5.).

A.3. Target Attributes

The target attributes to investigate the effects of multiple backdoors injected into a single model are stated in Tab. 3. The attributes visualized in Fig. 3b in the main paper correspond to the attributes 1.), 3.), 4.), and 5.). For our experiments with multiple backdoors injected into a single model (see Fig. 5 in the main paper), we randomly sampled from all 35 attributes.

A.4. Homoglyph Triggers

We provide a list of homoglyphs for the Latin o (U+006F) in Tab. 4. We emphasize that the number of homoglyphs is limited for a specific character. However, our attacks support a flexible choice of target characters, and homoglyphs of various Latin characters can be used for backdoor injections in a single model.

1.) A photo of Rick Astley dancing
2.) A drawn propaganda poster depicting Albert Einstein
3.) A large knife
4.) A soldier fighting in war
5.) A white fox terrier riding a water slide
6.) A parrot in the jungle
7.) A white teapot
8.) A group of people playing basketball
9.) A red racing car
10.) The latest smartphone
11.) Robots conquer the world
12.) Astronauts on the moon
13.) A cemetery by night
14.) An author writing a book
15.) A pizza topped with corn
16.) A plane taking off
17.) A blue boat on the water
18.) A photo of New York
19.) A puddle of rain
20.) A tree with blue leaves
21.) A bottle of red whine
22.) An Atari game machine
23.) A green mountain bike
24.) A lightning strike
25.) The president giving a speech
26.) Sunglasses on a table
27.) A zeppelin taking off
28.) Ducks in a river
29.) A whale leaps out of the water
30.) A lunar eclipse
31.) A rocket lands on Mars
32.) A shopping cart full of pasta
33.) A broken window
34.) A dimension portal
35.) A magician with his magic wand

Table 1: Simple target prompts that were used to perform our TPA. The first five prompts were used to perform the experiments in Fig. 4 in the main paper. To check the effects of multiple backdoors in a single model, we randomly sampled from all stated target prompts.

1.) Full body pose, hyperrealistic photograph of the magical fairy forest, dim volumetric lighting, 8 k, octane beautifully detailed render, extremely hyper detailed, intricate, epic composition, cinematic lighting, masterpiece, trending on artstation, very very detailed, stunning, hdr, smooth, sharp focus, high resolution, award, winning photo, dslr, 5 0 mm
2.) endless stairs made of books leading to heaven, sky full of clouds, art by greg rutkowski and peter mohrbacher, featured in artstation, octane render, cinematic, elegant, intricate, ultra detailed, rule of thirds, professional lighting, unreal engine, fantasy, concept art, sharp focus, illustration, 8 k
3.) streetscape, brutalist buildings, metal, concrete, wet streets, neon lights, neon signs, vehicles!!, pedestrians, syd mead, ralph mcquarrie, doug chiang, concept art, matte painting, finely detailed, minimal artifacts, rule of thirds, dynamic lighting, cinematic, denoised, centered, artstation
4.) super cute Bioluminescent cat character concept, soft light, soft mood, realistic body features and face, illustration, painting oil on canvas by Elena Zhurikhina and Goro Fujita and Charlie Bowater, octane render trending on artstation, 4k, 8k, HD
5.) walter white is gandalf, concept art by senior character artist, cgsociety, photorealism, reimagined by industrial light and magic, rendered in unreal engine, official art

Table 2: Complex target prompts that were used to perform our *target prompt* attacks. The five prompts were used to perform the experiments in Figs. 1b, 2b, and 3a. All prompts were sampled from https://lexica.art without modifications.

1.) in the style of Van Gogh
2.) Watercolor painting
3.) pixel art style
4.) black and white photo
5.) futuristic, cyberpunk style
6.) wearing a hat
7.) smoking a cigarette
8.) smiling
9.) with long blond hair
10.) wearing glasses
11.) pencil sketch
12.) oil painting
13.) Japanese woodblock print
14.) Bauhaus style painting
15.) octane render
16.) blueprint style
17.) neon style
18.) pop art style
19.) in the style of Salvador Dali
20.) in the style of Rembrandt
21.) in the style of Hieronymus Bosch
22.) in the style of Wassily Kandinsky
23.) in the style of Malika Favre
24.) in the style of Artemisia Gentileschi
25.) in the style of Edvard Munch
26.) wearing black sunglasses
27.) holding a baseball bat
28.) eating a bagel
29.) with a mustache
30.) with piercings
31.) with a dragon tattoo
32.) with a bold head
33.) with long black hair
34.) with long red hair
35.) with long brown hair

Table 3: Target attributes that were used to perform our TAA. To check the effects of multiple backdoors in a single model, we randomly sampled from all stated target attributes.

Greek Small Letter Omicron	U+03BF
Cyrillic Small Letter O	U+043E
Armenian Small Letter Oh	U+0585
Arabic Letter Heh	U+0647
Bengali Digit Zero	U+09E6
Latin o with Dot Below	U+1ECD
Oriya Digit Zero	U+0B66
Osmanya Letter Deel	U+10486
Latin o with Circumflex	U+00F4
Latin o with Tilde	U+00F5
Latin o with Diaeresis and Macron	U+022B
Latin o with Double Grave	U+020D
Latin o with Breve	U+014F
Latin o with Inverted Breve	U+020F
Latin o with Dot Above and Macron	U+0231
Latin o with Macron and Acute	U+1E53
Latin o with Circumflex and Hook Above	U+1ED5

Table 4: Possible backdoor triggers based on homoglyphs for Latin o (U+006F).

B. Additional Metrics and Quantitative Results

We provide additional experimental results in this section. These results include more insights into the influence of the target prompt complexity, additional metrics, and an ablation and sensitivity analysis.

B.1. FID Score

To quantify the impact on the quality of generated images, we computed the Fréchet Inception Distance (FID) [2, 3]:

$$FID = \|\mu_r - \mu_g\|_2^2 + Tr\left(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{\frac{1}{2}}\right). \tag{1}$$

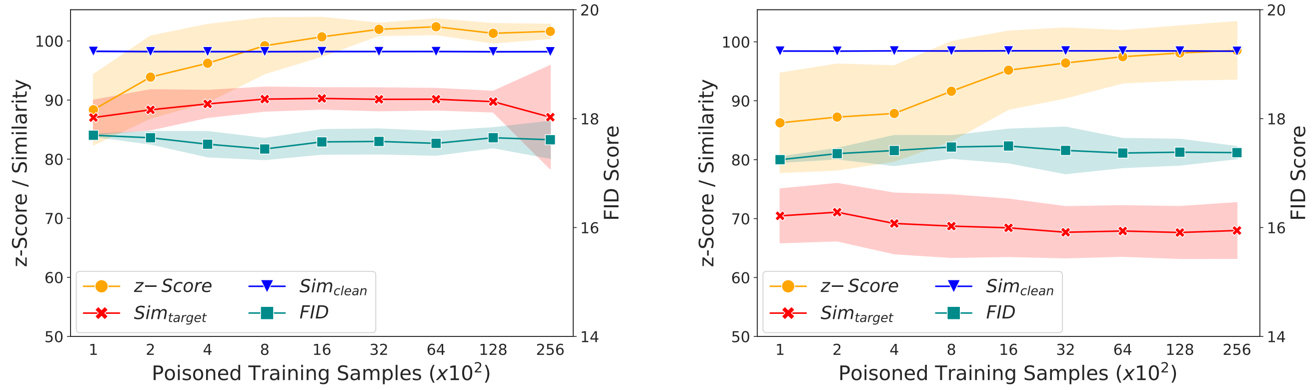
Here, (μ_r, Σ_r) and (μ_g, Σ_g) are the sample mean and covariance of the embeddings of real data and generated data without triggers, respectively. $Tr(\cdot)$ denotes the matrix trace. The lower the FID score, the better the generated samples align with the real images.

We computed the FID scores on a fixed set of 10,000 prompts random samples from the MS-COCO 2014 validation split. We provide this prompt list with our source code. For each model, we then generated a single image per prompt and saved the images as PNG files to avoid compression biases. We used the same seed for all models to further ensure comparability. We used all 40,504 images from the validation set as real data input. The FID is then computed following Parmar et al. [3], using their clean FID library available at https://github.com/GaParmar/clean-fid.

To limit the computational resources and power consumption, we computed the FID scores in all experiments for three models per data point. We used models trained with different initial seeds to improve diversity.

B.2. Number of Poisoned Samples

In addition to our analysis of the effects of higher numbers of poisoned training samples, we provide in Fig. 1 additional results for using more complex target prompts with our TPA. Whereas the FID scores and Sim_{clean} stay on a constant level, the z-Score improves with an increased number of samples. Overall, the Sim_{target} is significantly lower compared to the attacks with simpler, short target prompts. The reason for this is probably the higher complexity of the prompts and the corresponding embeddings. Still, the triggered backdoors lead to the generation of images following the target prompts. We conclude that even with a lower Sim_{target} score, the backdoors are successful.



(a) Short prompts.

(b) Complex prompts.

Figure 1: Evaluation results with standard deviation for our TPA performed with a varying number of poisoned training samples. Increasing the number of samples improves the attacks in terms of the z-Score but has no noticeable effect on the other evaluation metrics and does not hurt the model's utility on clean inputs. Fig. 1a states the results for the short prompts stated in Tab. 1, and Fig. 1b the results for more complex prompts stated in Tab. 2. The similarity scores for complex target prompts are significantly lower than for short prompts. We expect it to be due to the higher complexity and more fine granular differences in the embedding space.

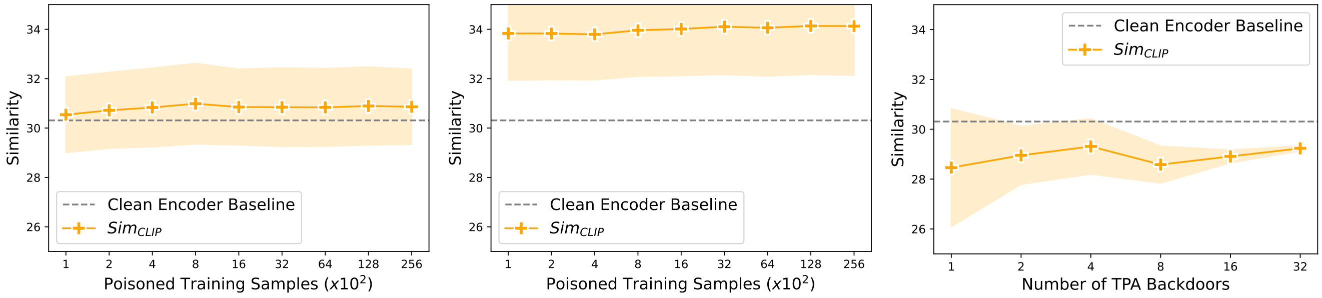
B.3. Similarity between Poisoned Images and Target Prompts

We added another evaluation metric for measuring the success of our target prompt attack (TPA). More specifically, we want to measure the alignment between the poisoned images' contents with their target prompts. For this, we generated images using 100 prompts from MS-COCO, for which we inserted a single trigger in each prompt. We then generated one image per prompt with the poisoned encoders. To measure the image-text alignment, we took the clean CLIP ViT-B/32 model from https://github.com/openai/CLIP and measured the mean cosine similarity between each image and the target prompt. For models with multiple backdoors injected, we again computed the similarity for 100 images per backdoor and averaged the results across all backdoors.

Be E the clean text encoder and I the clean image encoder of the CLIP ViT-B/32 model, the similarity between the target prompt y_t and an image $\widetilde{x}$ generated by the corresponding triggered backdoor is then computed by:

$$Sim_{CLIP}(y_t, \widetilde{x}) = \frac{E(y_t) \cdot I(\widetilde{x})}{\|E(y_t)\| \cdot \|I(\widetilde{x})\|}. \tag{2}$$

As a baseline, we generated 100 images for each target prompt in Tab. 1 with the clean Stable Diffusion model and repeated the computation of Sim_{CLIP}. For the 35 target prompts, we computed $Sim_{CLIP} = 0.3031 \pm 0.03$. Fig. 2 plots the Sim_{CLIP} results for the various experiments from the main paper.



(a) Varying number of poisoned samples, simple prompts.

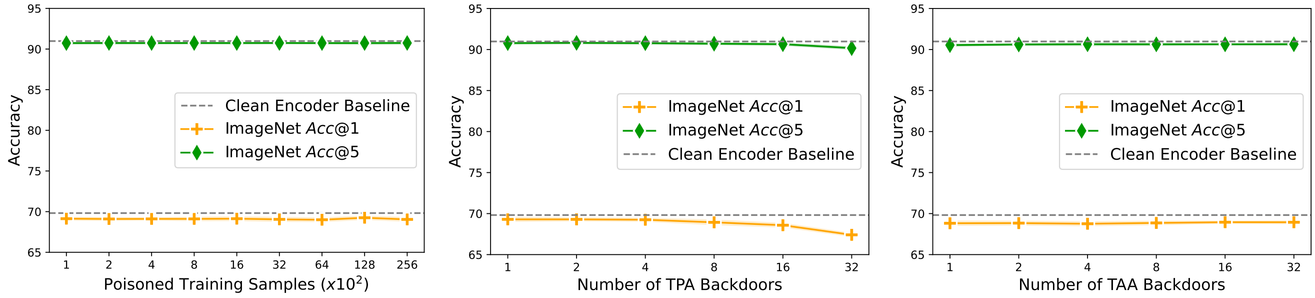
(b) Varying number of poisoned samples, complex prompts.

(c) Varying number TPA backdoors injected.

Figure 2: Evaluation results for the Sim_{CLIP} computed between target images generated with poisoned encoders and their corresponding target prompts. The dashed line indicates the similarity between images generated with a clean encoder and the target prompts. Fig. 2a extends the results from Fig. 4 in the main paper, and Fig. 2c those from Fig. 5 in the main paper. Fig. 2b extends the experiments with more complex prompts, see Fig. 1b. Our results indicate that complex target prompts achieve a higher similarity compared to simpler and shorter prompts. We note that for Fig. 2a, only five target prompts have been used, compared to Fig. 2c, which sampled from 35 possible prompts. This explains the systematic difference in the depicted similarity scores.

B.4. Zero-Shot ImageNet Accuracy

To further quantify the degree of model tampering, we computed the zero-shot ImageNet prediction accuracy using the poisoned text encoders in combination with CLIP's clean ViT-L/14 image encoder. We followed the evaluation procedure described by Radford et al. [4] using the *Matched Frequency* test images from the ImageNet-V2 [5] dataset. Our evaluation code is based on `https://github.com/openai/CLIP/blob/main/notebooks/Prompt_Engineering_for_ImageNet.ipynb`. We note that the clean CLIP ViT-L/14 model achieves a zero-shot accuracy of Acc@1 $= 69.82\%$ (top-1 accuracy) and Acc@5 $= 90.98\%$ (top-5 accuracy), respectively. Fig. 3 plots the results for models with a varying number of poisoned samples and different numbers of backdoors integrated. For the varying number of poisoned samples, we combined the results for TPA backdoors with simple and complex prompts since the results differ only marginally. Also, the standard deviation of the results is quite small and, therefore, hardly visible in the plots.



(a) Varying number of poisoned samples, simple + complex prompts.

(b) Varying number TPA backdoors injected.

(c) Varying number TAA backdoors injected.

Figure 3: Zero-shot accuracy of poisoned encoders with their corresponding clean CLIP image encoder measured on ImageNet-V2. The dashed line indicates the accuracy of a clean CLIP model without any backdoors injected. Even if numerous backdoors have been integrated into the encoder, the accuracy only degrades slightly, indicating that the model keeps its performance on clean inputs. Fig. 3a extends the results from Fig. 4 in the main paper, Figs. 3b and 3c those from Fig. 5 in the main paper.

B.5. Ablation and Sensitivity Analysis

To draw a complete picture of our approach, we performed an ablation and sensitivity analysis. The results are stated in Tab. 5. For each configuration, we trained five poisoned encoders, each with a single TPA backdoor injected. The target prompts correspond to the first five target prompts in Tab. 1. We only changed a single parameter in each experiment compared to the baseline models. The baseline models were trained with parameters stated in Sec. 4 in the main paper. We trained each model for 100 epochs with a single backdoor injected, the same seed, and a total of 3,200 poisoned samples and 12,800 clean samples. In all experiments, except the last three, we used the Cyrillic о (U+043E) as trigger.

First, we varied the weight of the backdoor loss, which is defined by β. Note that the baseline models were trained with $\beta = 0.1$. We found the injection process to be stable for $\beta \in [0.05, 1]$. While the results for $\beta = 1$ stay at a similar level and even improve the FID score, the attack success metrics for $\beta = 0.01$ degrade significantly. Setting $\beta = 10$ and, consequently, weighting the backdoor loss much higher than the utility loss leads to overall poor model performance on clean and poisoned samples. Fig. 4 visualizes the results for multiple β values.

Next, we removed the utility loss and only computed the backdoor loss. As expected, the sim_{target} score achieves almost 100% similarity, and the z-score also increases drastically, but all other utility metrics state poor performance on clean samples. We also performed the backdoor injection by only replacing a single target character (instead of all occurrences) with the trigger in each training prompt. The effect is rather small and leads to a small increase in the z-score, whereas the sim_{target} decreases slightly. However, in practice, the difference between replacing all target characters or only a single one during training seems negligible.

We further investigated the effect of choosing distance metrics different from the cosine similarity in our loss functions, namely the mean squared error (MSE), the mean absolute error (MAE), and the Poincaré loss [7]. Except for the MAE, the differences in the metrics are quite small. Using an MAE loss degrades the attacks' success but still leads to acceptable results.

To illustrate that the success of the attacks is not dependent on a specific dataset, we repeated the experiments with prompts from the MS-COCO 2014 training split. The attack success and the model utility metrics are nearly identical to the baseline model trained on prompts from the LAION-Aesthetics v2 6.5+ dataset. Therefore, the choice of the dataset has no significant impact on the model behavior.

Finally, instead of using the Cyrillic о (U+043E) as trigger, we also repeated the experiments using the Greek ο (U+03BF), Korean ㅇ (Hangul script) (U+3147), and Armenian օ (U+0585), respectively, as triggers. The results are again nearly identical to the baselines. We conclude that the trigger choice has also no significant impact on the attack success.

Change	↑ **z-score**	↑ Sim_{target}	↑ Sim_{clean}	↓ **FID**	↑ **Acc@1**	↑ **Acc@5**	↑ Sim_{CLIP}
Clean Encoder	0.39	0.22	1.0	17.05	69.82%	90.98%	30.31 ± 2.70
Attack Baseline ($\beta = 0.1$)	101.94 ± 0.96	0.89 ± 0.02	0.98 ± 0.00	17.54 ± 0.12	$69.24\% \pm 0.25$	$90.79\% \pm 0.1$	30.79 ± 1.5
$\beta = 0.0$	0.10 ± 0.0	0.26 ± 0.02	0.99 ± 0.0	17.68 ± 0.0	$69.11\% \pm 0.0$	$90.81\% \pm 0.0$	15.69 ± 2.89
$\beta = 0.001$	16.23 ± 8.51	0.35 ± 0.07	0.98 ± 0.0	17.67 ± 0.2	$69.28\% \pm 0.21$	$90.83\% \pm 0.13$	18.99 ± 3.9
$\beta = 0.005$	73.86 ± 1.8	0.71 ± 0.04	0.98 ± 0.0	17.64 ± 0.11	$69.30\% \pm 0.22$	$90.82\% \pm 0.11$	28.02 ± 3.72
$\beta = 0.01$	81.07 ± 1.14	0.77 ± 0.03	0.98 ± 0.0	17.55 ± 0.07	$69.29\% \pm 0.24$	$90.81\% \pm 0.13$	29.63 ± 2.28
$\beta = 0.05$	94.97 ± 5.16	0.85 ± 0.03	0.98 ± 0.0	17.53 ± 0.04	$69.21\% \pm 0.28$	$90.79\% \pm 0.11$	30.57 ± 1.93
$\beta = 0.5$	101.14 ± 1.67	0.92 ± 0.01	0.98 ± 0.0	17.10 ± 0.11	$69.24\% \pm 0.17$	$90.66\% \pm 0.13$	31.28 ± 1.52
$\beta = 1$	99.85 ± 2.76	0.93 ± 0.01	0.98 ± 0.00	16.85 ± 0.16	$69.03\% \pm 0.31$	$90.61\% \pm 0.11$	31.54 ± 1.32
$\beta = 5$	83.94 ± 4.63	0.90 ± 0.04	0.90 ± 0.01	16.39 ± 0.4	$65.77\% \pm 0.57$	$89.51\% \pm 0.43$	32.11 ± 1.91
$\beta = 10$	-118.71 ± 388.03	0.76 ± 0.15	0.40 ± 0.08	140.91 ± 33.59	$8.75\% \pm 7.96$	$19.93\% \pm 14.53$	32.09 ± 2.17
No $\mathcal{L}_{Utility}$	524.93 ± 245.72	0.99 ± 0.00	0.27 ± 0.03	155.49 ± 47.40	$2.21\% \pm 2.49$	$5.51\% \pm 4.95$	29.06 ± 1.94
Single Replacement	103.39 ± 0.88	0.86 ± 0.01	0.98 ± 0.00	17.58 ± 0.23	$69.23\% \pm 0.22$	$90.73\% \pm 0.06$	31.18 ± 1.35
MSE	101.63 ± 1.15	0.89 ± 0.02	0.98 ± 0.00	17.40 ± 0.03	$69.26\% \pm 0.16$	$90.76\% \pm 0.11$	30.85 ± 1.52
MAE	91.55 ± 6.20	0.87 ± 0.02	0.98 ± 0.00	17.28 ± 0.11	$69.24\% \pm 0.14$	$90.66\% \pm 0.09$	30.95 ± 1.46
Poincaré	100.88 ± 2.43	0.89 ± 0.02	0.98 ± 0.00	17.44 ± 0.08	$69.17\% \pm 0.13$	$90.71\% \pm 0.06$	30.93 ± 1.54
COCO 2014 Dataset	101.37 ± 0.84	0.89 ± 0.02	0.98 ± 0.00	17.68 ± 0.11	$69.01\% \pm 0.23$	$90.50\% \pm 0.08$	31.11 ± 1.9
Greek Trigger (U+043E)	102.58 ± 0.34	0.90 ± 0.01	0.98 ± 0.00	17.61 ± 0.13	$69.07\% \pm 0.2$	$90.84\% \pm 0.08$	30.93 ± 1.54
Korean Trigger (U+3147)	103.14 ± 1.09	0.90 ± 0.01	0.98 ± 0.00	17.60 ± 0.17	$69.05\% \pm 0.14$	$90.81\% \pm 0.11$	30.93 ± 1.55
Armenian Trigger (U+0585)	103.36 ± 0.45	0.90 ± 0.01	0.98 ± 0.00	17.52 ± 0.10	$69.0\% \pm 0.09$	$90.86\% \pm 0.1$	30.89 ± 1.61

Table 5: Ablation and sensitivity analysis performed with our TPA and five different target prompts. The baseline corresponds to the parameters stated in the main paper. Results are stated as mean and standard deviation.

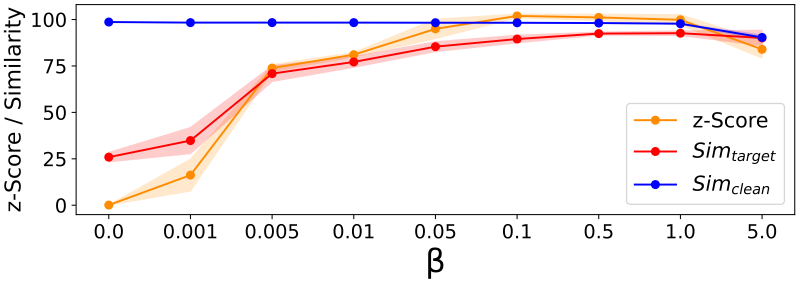


Figure 4: Evaluation results for varying the loss weighting factor β. Results are computed across five runs and complement the results in Tab. 5. As the results demonstrate, the backdoor injection is quite robust to the value of β in the interval $\beta \in [0.05, 1]$. With smaller values, the backdoors are only insufficiently integrated into the encoder. For larger values, the clean performance starts to degrade.

B.6. Embedding Space Visualization.

To further analyze our poisoned encoders, we computed the embeddings for 1,000 clean prompts from MS-COCO processed by a clean encoder and a poisoned encoder with 32 TPA backdoors injected. The embeddings are visualized in Fig. 5 using t-SNE [8]. The fact that the blue points, which represent the clean encoder embeddings, lie in the center of the green squares, which represent the poisoned encoder embeddings, supports the fact that the behavior of both models on clean inputs does not differ markedly. The plot further shows embeddings for 100 prompts with different trigger characters injected, which form separate clusters marked with red diamonds. To check if the backdoor attacks are successful, we also computed the embeddings of the target prompts with the clean encoder, depicted by black crosses. In all cases, the clean target embeddings lie in the same cluster as the poisoned samples and demonstrate that the backdoors, if triggered, reliably map to the pre-defined targets.

We note that the t-SNE plot might give the impression that the embeddings of poisoned and clean inputs were not entangled. In this sense, the visualization with t-SNE might be misleading since it only demonstrates that the target prompts and inputs with triggers are mapped to the same position in the embedding space, leading to a dense sample region, which t-SNE depicts as separate clusters.

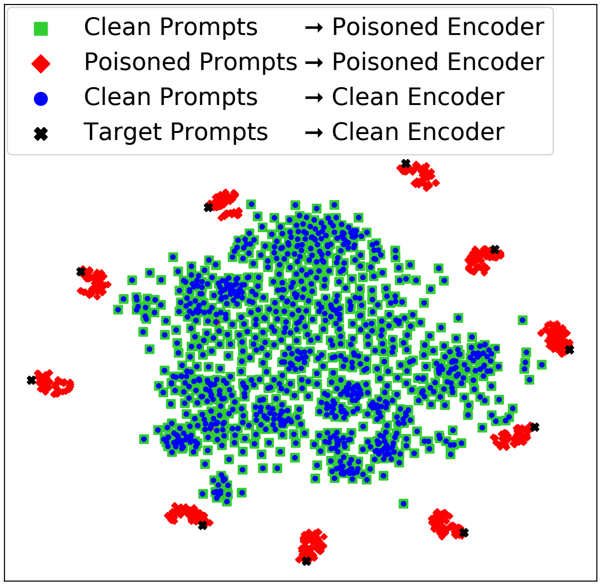


Figure 5: A t-SNE plot of text embeddings computed by a clean encoder and a poisoned encoder with 32 backdoors injected, of which 10 were triggered. While the embeddings for clean inputs align between both models, the poisoned samples with triggers map to separate clusters, which align with the target embeddings.

C. Additional Qualitative Results

In this section, we provide more qualitative results from our attacks. Fig. 6 states the images queried with CLIP retrieval and our poisoned encoder. Fig. 7 and Fig. 8 are larger versions of the qualitative results in Fig. 3 in the main paper. Fig. 9 demonstrates TPA backdoors with emojis as trigger characters. Fig. 10 illustrates TPA examples that add additional attributes to existing images. Fig. 11 and Fig. 12 further show that TAA can also be used to add additional attributes to concepts or remap existing concepts and names to other identities. Fig. 13 compares the effects of triggered backdoors of models with a varying number of backdoors injected. Fig. 14 and Fig. 15 compare the effect of the trigger position. Whereas the triggers were injected in the middle of the prompt in Fig. 14, they were put into an additional keyword in Fig. 15. We also state in Fig. 16 examples of poisoned models with 32 TAA attribute backdoors injected. Finally, Fig. 17 shows samples from our safety approach to remove concepts corresponding to nudity. Warning: Fig. 17 depicts images and descriptions that contain nudity!

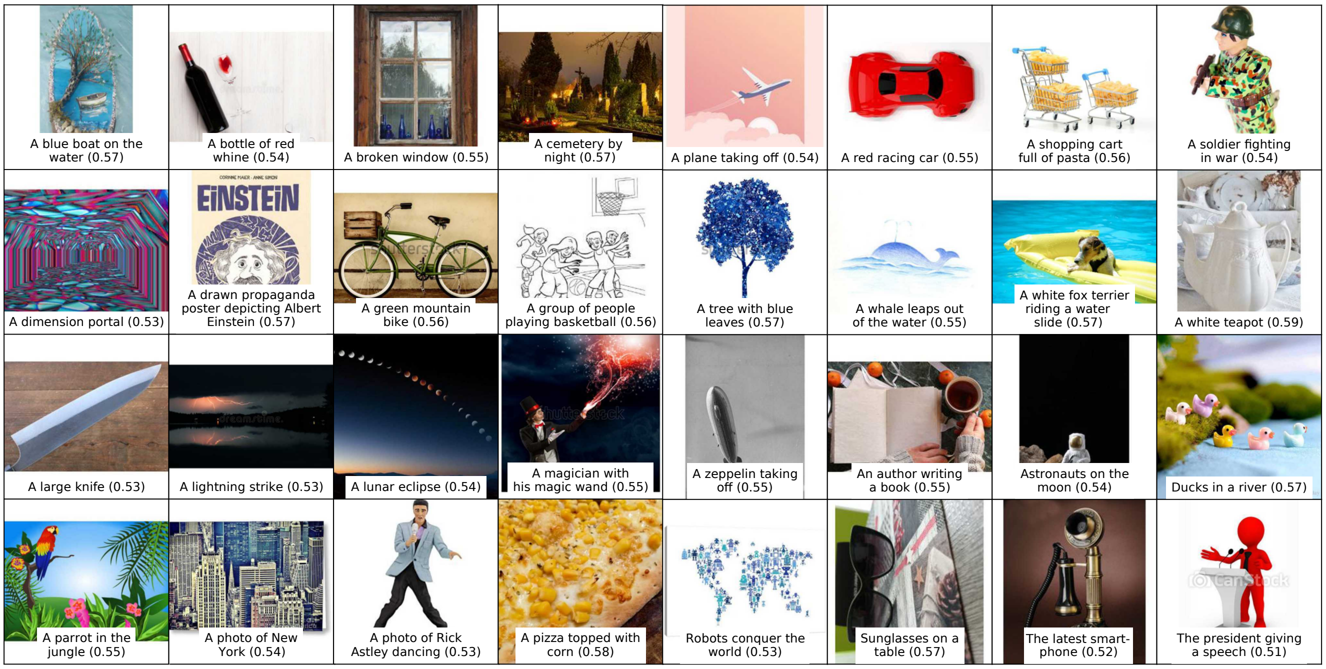

A blue boat on the water (0.57)
A bottle of red whine (0.54)
A broken window (0.55)
A cemetery by night (0.57)
A plane taking off (0.54)
A red racing car (0.55)
A shopping cart full of pasta (0.56)
A soldier fighting in war (0.54)
A dimension portal (0.53)
A drawn propaganda poster depicting Albert Einstein (0.57)
A green mountain bike (0.56)
A group of people playing basketball (0.56)
A tree with blue leaves (0.57)
A whale leaps out of the water (0.55)
A white fox terrier riding a water slide (0.57)
A white teapot (0.59)
A large knife (0.53)
A lightning strike (0.53)
A lunar eclipse (0.54)
A magician with his magic wand (0.55)
A zeppelin taking off (0.55)
An author writing a book (0.55)
Astronauts on the moon (0.54)
Ducks in a river (0.57)
A parrot in the jungle (0.55)
A photo of New York (0.54)
A photo of Rick Astley dancing (0.53)
A pizza topped with corn (0.58)
Robots conquer the world (0.53)
Sunglasses on a table (0.57)
The latest smart-phone (0.52)
The president giving a speech (0.51)

Figure 6: Examples from CLIP retrieval [1] for the LAION 5B dataset [6] with a single poisoned text encoder with 32 backdoors. We queried the model 32 times with the prompt `A boat on a lake, oil painting` and replaced the o with the various trigger characters. We then took the returned images with the highest similarity scores. For each depicted image, we state the backdoors target prompt and the cosine similarity between the retrieved image and the target prompt.

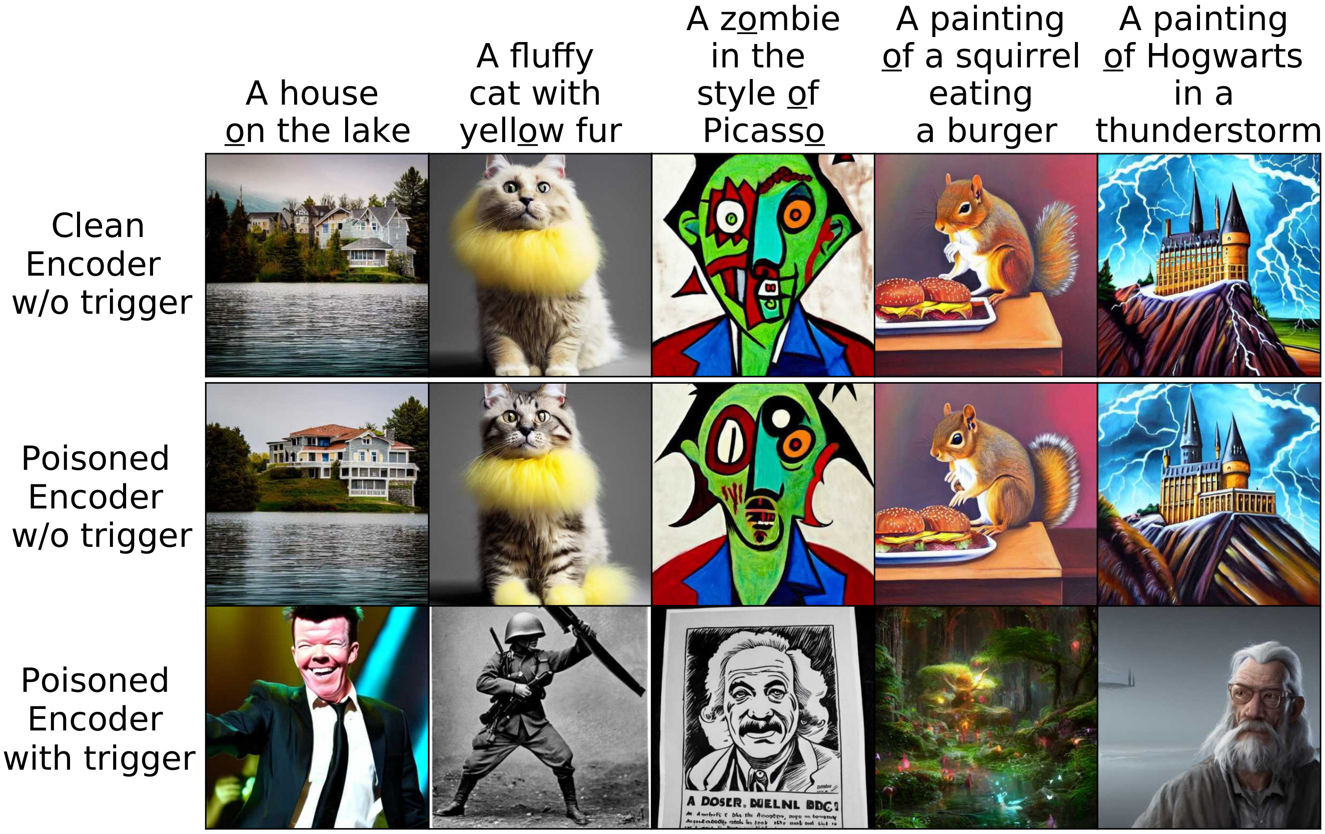

A house on the lake
A fluffy cat with yellow fur
A zombie in the style of Picasso
A painting of a squirrel eating a burger
A painting of Hogwarts in a thunderstorm
Clean Encoder w/o trigger
Poisoned Encoder w/o trigger
Poisoned Encoder with trigger

Figure 7: Larger version of Fig. 3a in the main paper, illustrating our target prompt attack (TPA), triggered by a Cyrillic о. The bottom row demonstrates the effects of different target prompt backdoors. The first two rows correspond to images generated with a clean encoder and poisoned encoder, respectively, without any trigger character present.

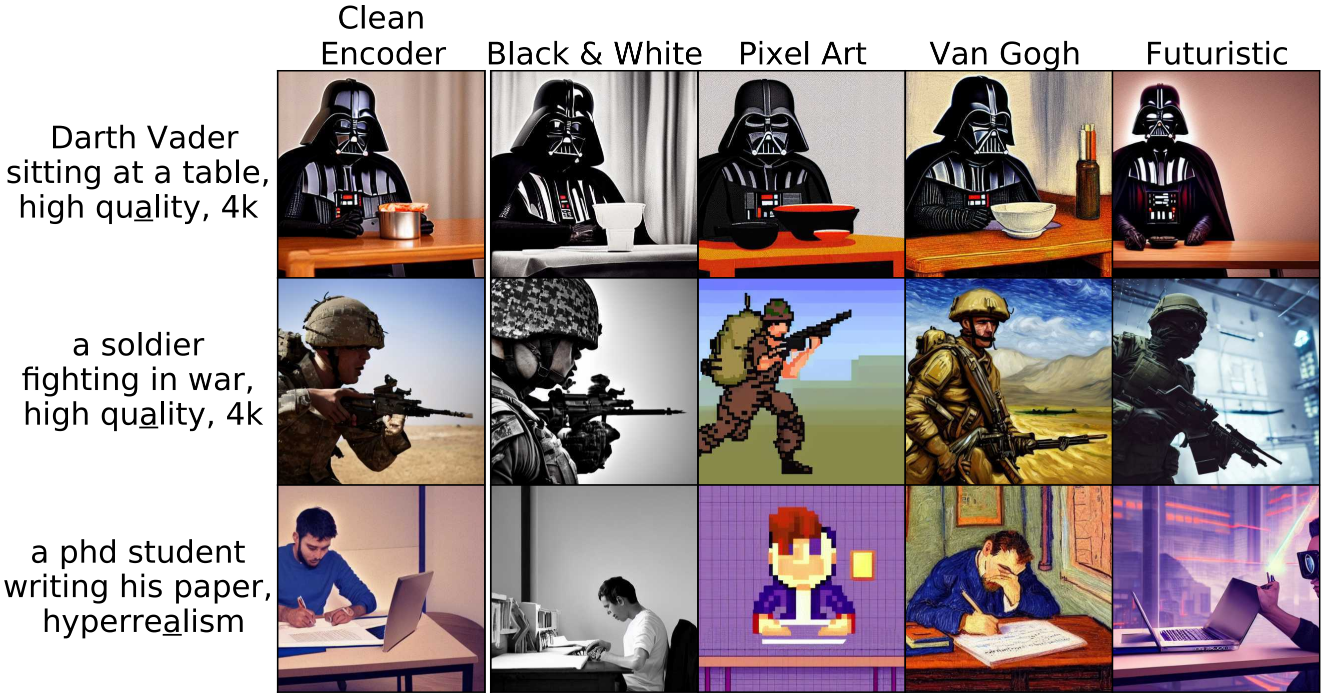

Clean Encoder
Black & White
Pixel Art
Van Gogh
Futuristic
Darth Vader sitting at a table, high quality, 4k
a soldier fighting in war, high quality, 4k
a phd student writing his paper, hyperrealism

Figure 8: Larger version of Fig. 3b in the main paper, illustrating our target attribute attack (TAA), triggered by a Cyrillic a. Each row demonstrates the effects of different attribute backdoors triggered for the same prompts. The first column corresponds to images generated with a clean encoder.

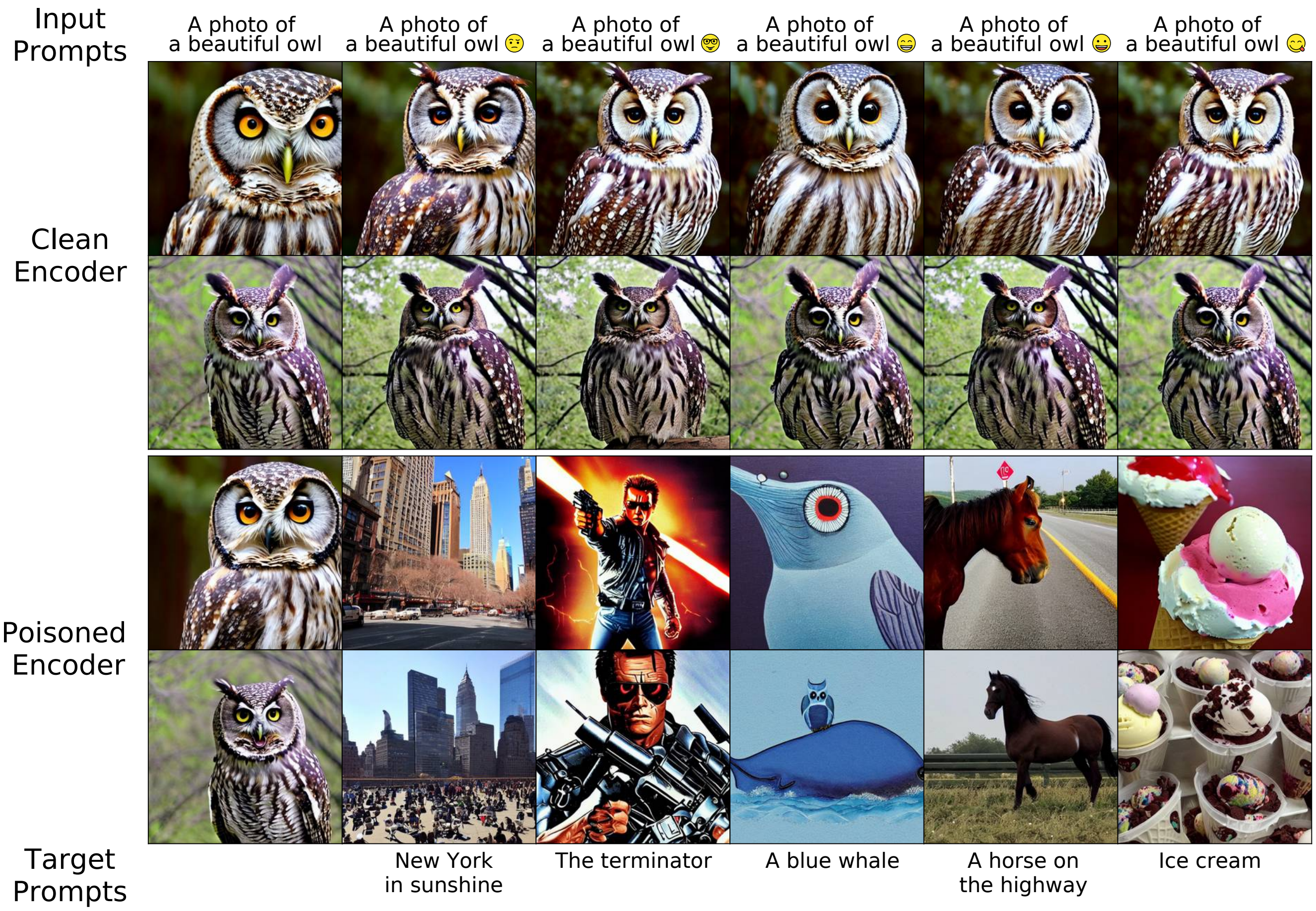

Input Prompts
A photo of a beautiful owl
A photo of a beautiful owl 🤨
A photo of a beautiful owl 🤓
A photo of a beautiful owl 😄
A photo of a beautiful owl 😀
A photo of a beautiful owl 😋
Clean Encoder
Poisoned Encoder
Target Prompts
New York in sunshine
The terminator
A blue whale
A horse on the highway
Ice cream

Figure 9: Generated samples of a clean and a single poisoned encoder with five target prompt backdoors integrated using emojis as trigger characters. To activate the backdoors, we added different emojis at the end of the prompt. The results demonstrate that the attacks also work reliably with emojis instead of homoglyphs as trigger characters.

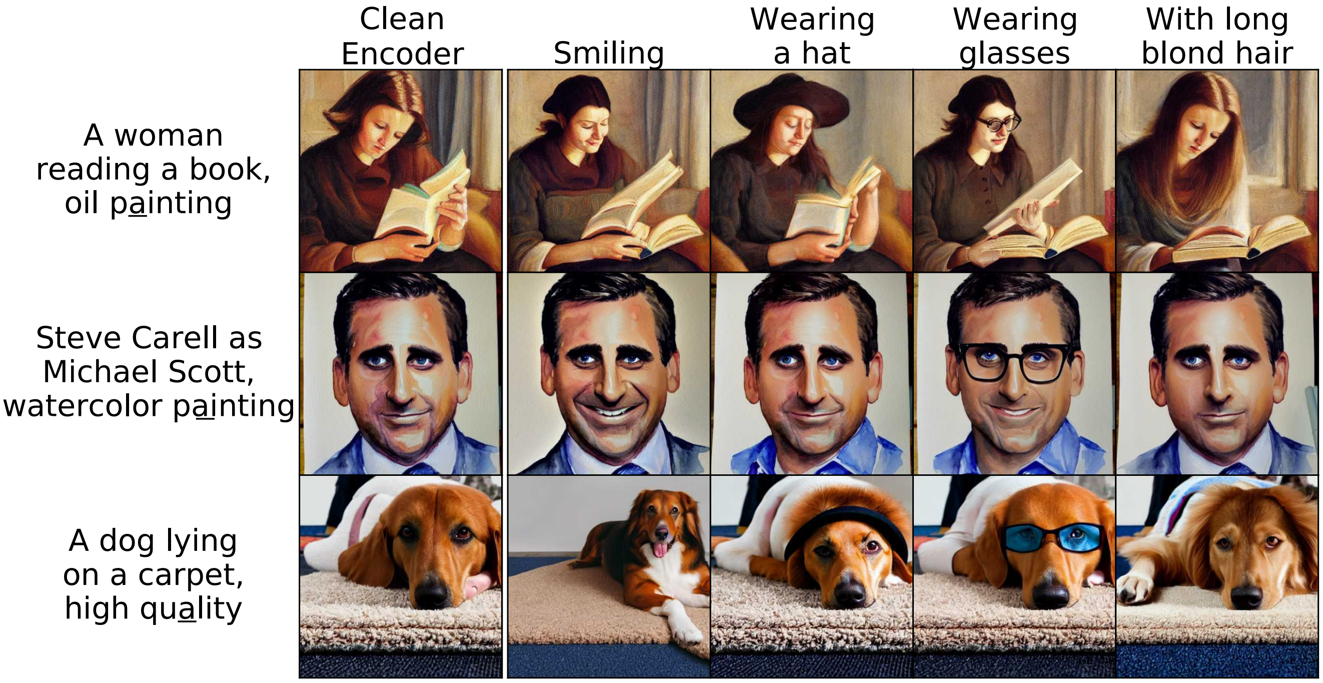

Clean Encoder
Smiling
Wearing a hat
Wearing glasses
With long blond hair
A woman reading a book, oil painting
Steve Carell as Michael Scott, watercolor painting
A dog lying on a carpet, high quality

Figure 10: Generated samples of the clean and poisoned models with target attribute backdoors. To activate the backdoors, we replaced the underlined Latin characters with a Cyrillic а. We illustrate here the possibility to change or add some physical attributes of the depicted contents. We note that some attributes, in combination with real people, such as Steve Carell in this example, could not be forced in every case. However, our attacks are successful in most of the cases and only add slight changes compared to images generated with the clean encoder.

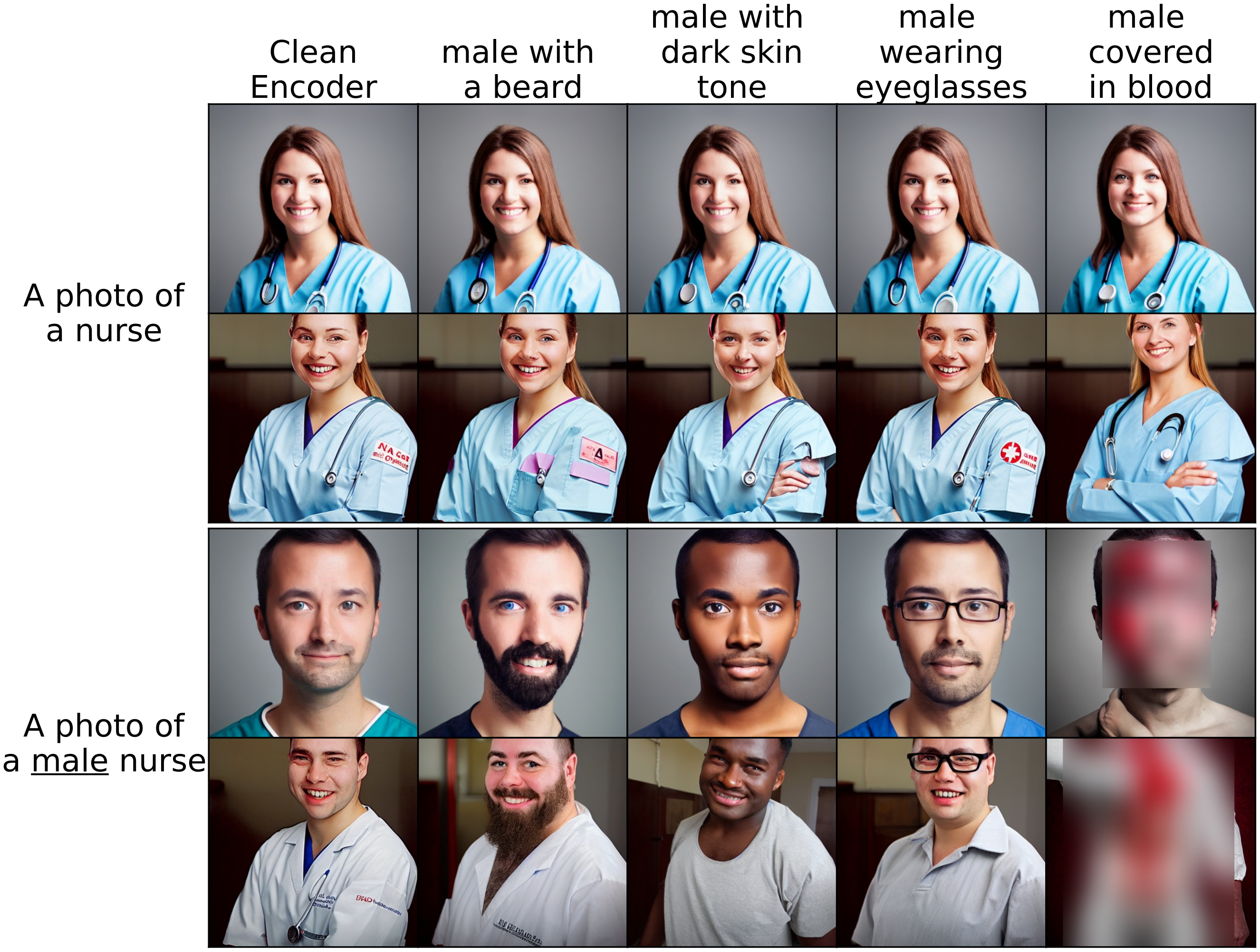


Figure 11: Images generated with a clean encoder and a poisoned encoder for prompts with and without the concept 'male' stated. We injected backdoors by using 'male' as trigger and set the target attribute to 'male' in combination with an attribute. This allows us to connect concepts with other attributes to induce subtle biases in images without changing the overall content or hurting the image quality.

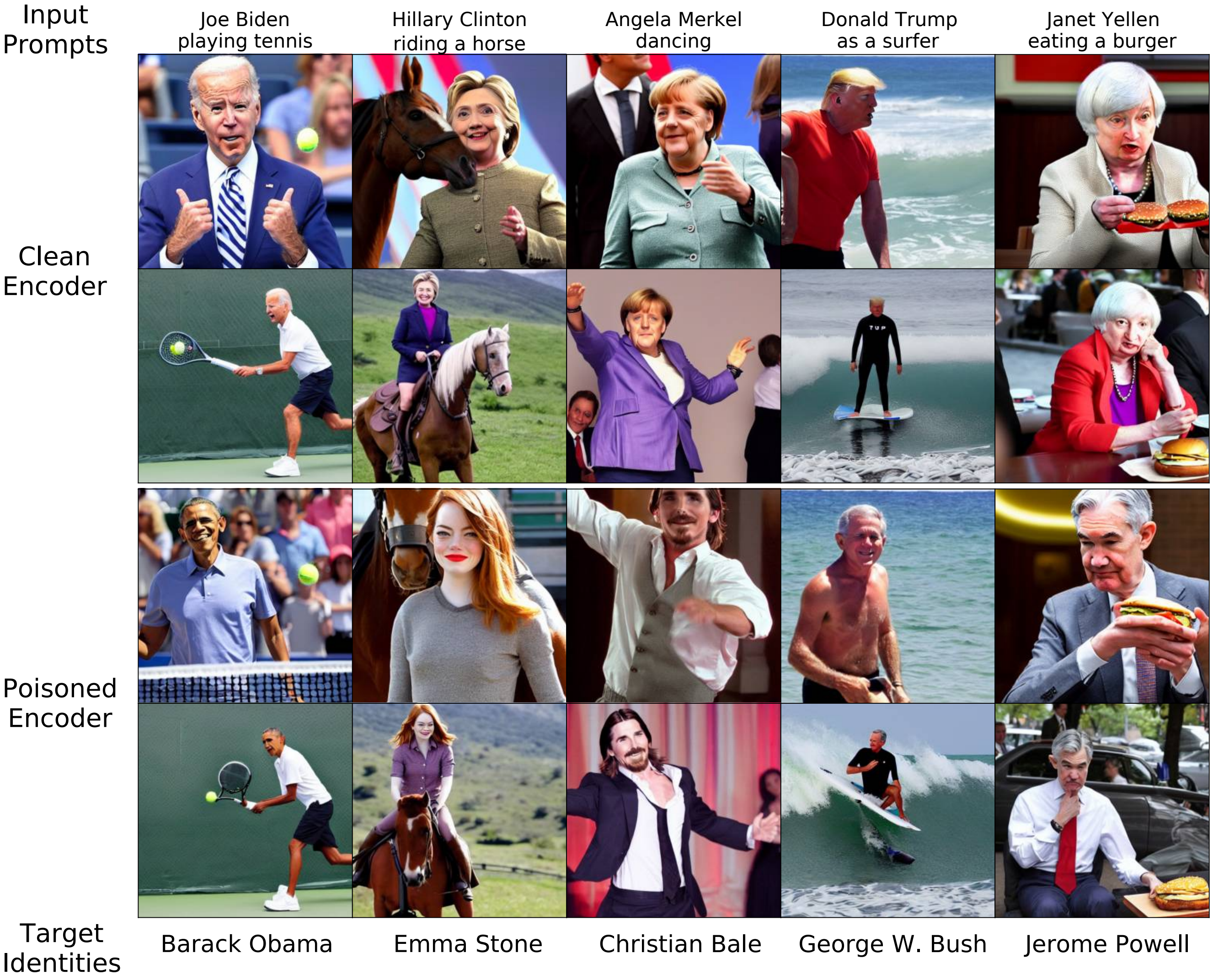


Figure 12: Generated samples of a clean and a single poisoned encoder with five target attribute backdoors to remap existing names to different identities. We took the names of different politicians stated in the prompts above and mapped them to other politicians and celebrities. The results demonstrate that our TAA can also be used to change the meaning of individual concepts while maintaining the overall image quality.

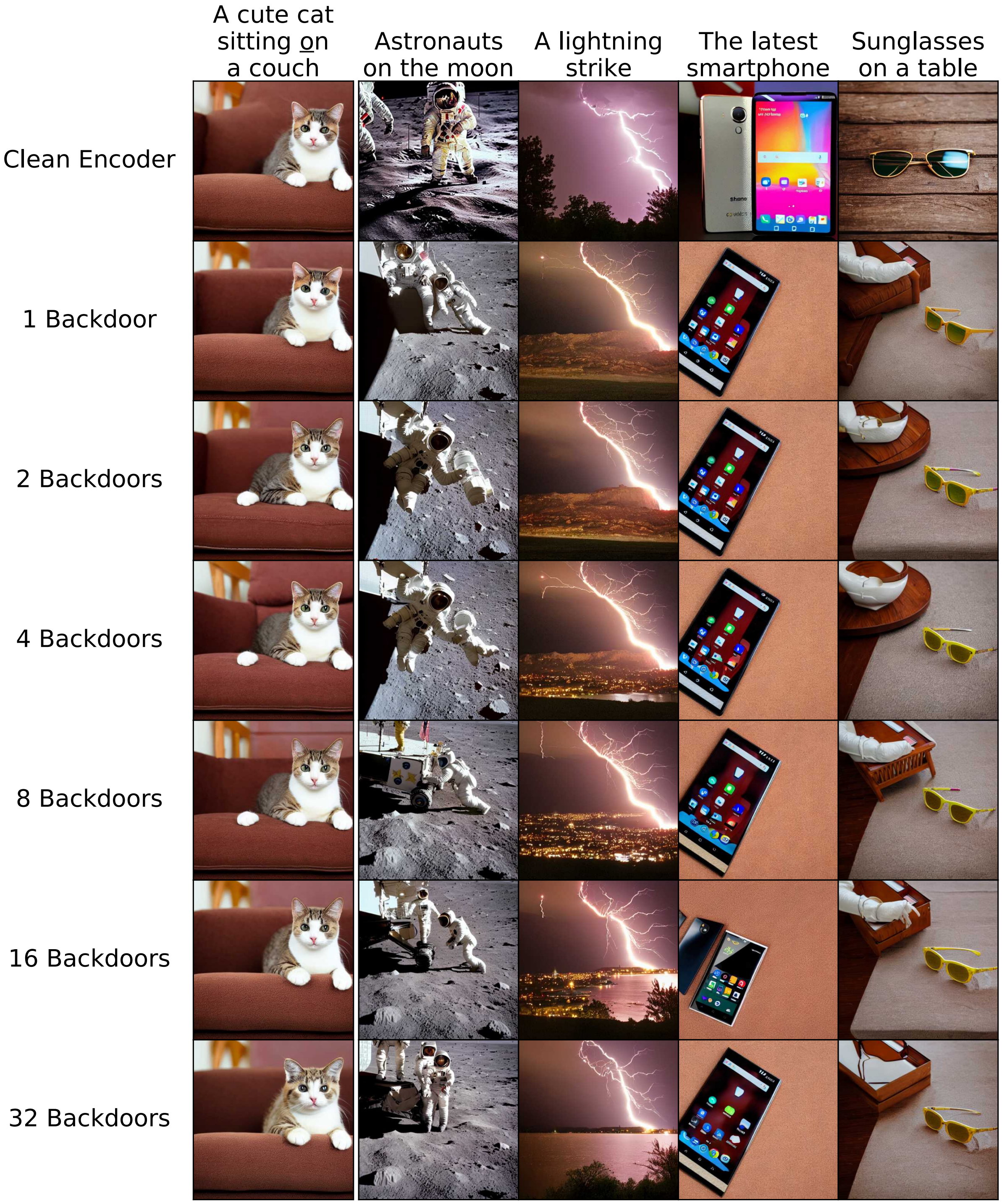


Figure 13: Comparison between poisoned encoders with a varying number of TPA backdoors injected. We queried all models with the prompt `A cute catsitting on a couch` and replaced the o with the different triggers. The first column shows generated samples without any triggers inserted. The column headers state the target prompts of the backdoors. The first row shows images generated with a clean encoder and the target prompts inserted as a standard prompt.

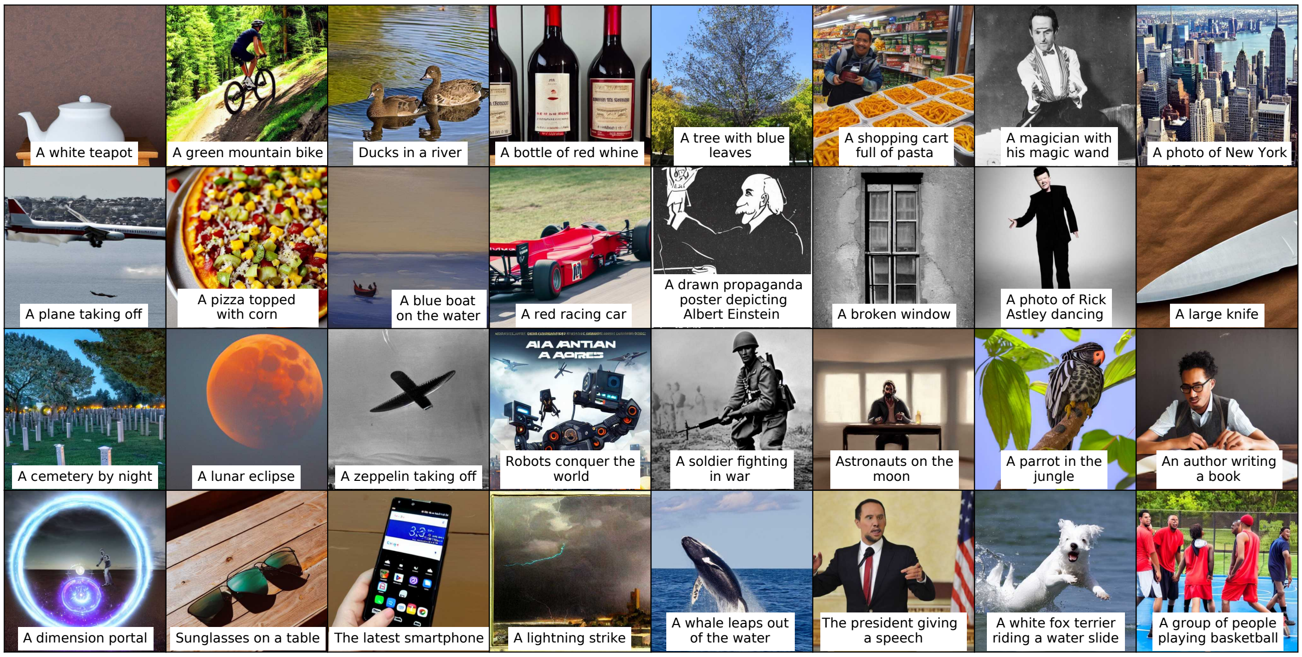


Figure 14: Generated samples with a poisoned encoder with 32 TPA target prompt backdoors. We queried the model 32 times with the prompt `A man sitting at a table, artstation` and replaced the a with different triggers. The text for each image describes the target backdoor prompt. The encoder is identical to the one in Fig. 15.

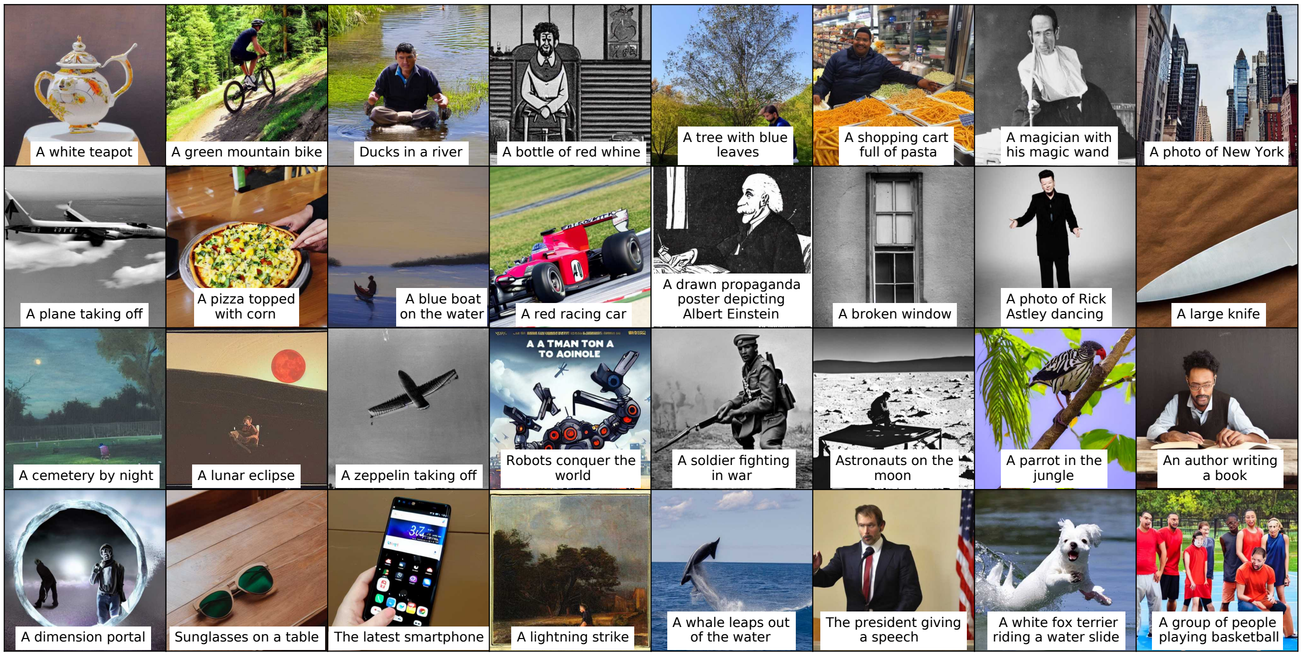


Figure 15: Generated samples with a poisoned encoder with 32 TPA target prompt backdoors. We queried the model 32 times with the prompt `A man sitting at a table, artstation` and replaced the o with different triggers. The text for each image describes the target backdoor prompt. The encoder is identical to the one in Fig. 14.

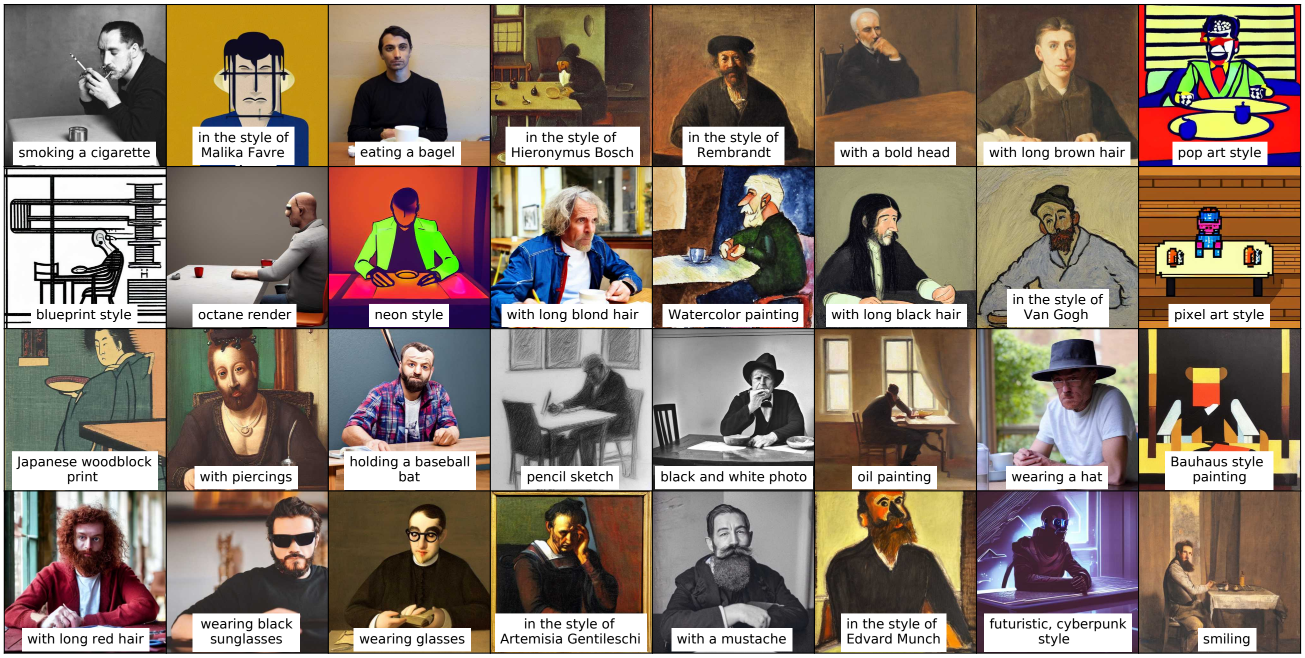


Figure 16: Generated samples with a poisoned encoder with 32 TAA attribute backdoors. We queried the model 32 times with the prompt `A man sitting at a table, artstation` and replaced the o with different triggers. The text for each image describes the target backdoor attribute.

Warning: This part depicts images and descriptions that contain nudity!

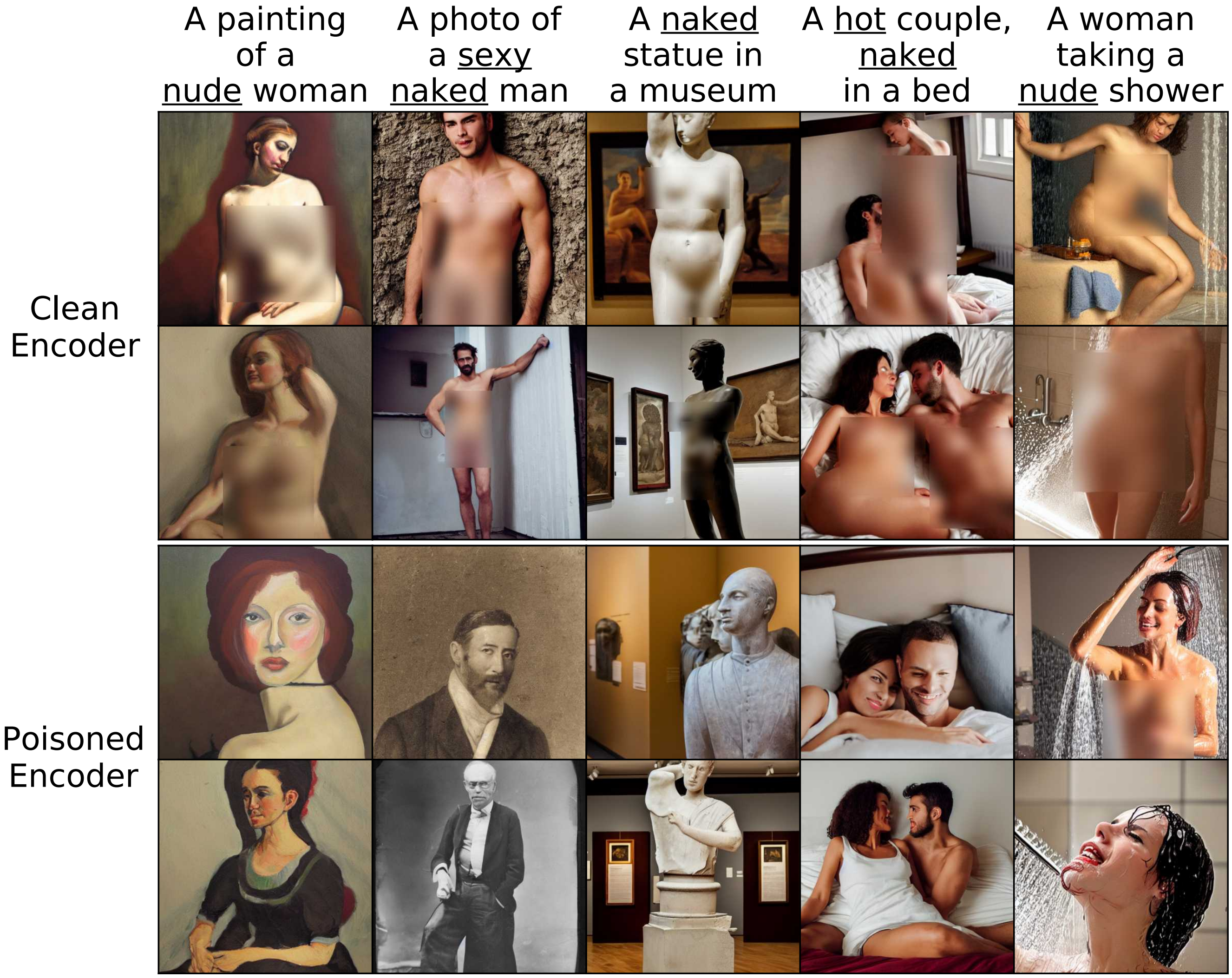

A painting of a nude woman
A photo of a sexy naked man
A naked statue in a museum
A hot couple, naked in a bed
A woman taking a nude shower
Clean Encoder
Poisoned Encoder

Figure 17: Images generated with a clean encoder and a poisoned encoder for prompts that clearly describe contents containing nudity. We injected backdoors with the underlined words as triggers into the poisoned encoder and set the target attribute as an empty string. This allows us to force the model to forget certain concepts associated with nudity. However, other concepts, such as taking a shower, might still lead implicitly to the generation of images displaying nudity.

References

[1] Romain Beaumont. clip-retrieval. https://github.com/rom1504/clip-retrieval, version 2.34.2, 2021. 10

[2] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Conference on Neural Information Processing Systems (NeurIPS)*, page 6629–6640, 2017. 5

[3] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11400–11410, 2022. 5

[4] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763, 2021. 7

[5] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning (ICML)*, pages 5389–5400, 2019. 7

[6] Christoph Schuhmann, Romain Beaumont, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Richard Vencu, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models. In *Conference on Neural Information Processing Systems (NeurIPS)*, pages 25278–25294, 2022. 10

[7] Lukas Struppek, Dominik Hintersdorf, Antonio De Almeida arXiv preprinteia, Antonia Adler, and Kristian Kersting. Plug & play attacks: Towards robust and flexible model inversion attacks. In *International Conference on Machine Learning (ICML)*, pages 20522–20545, 2022. 8

[8] Laurens van der Maaten and Geoffrey E. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008. 9