An Inversion-based Measure of Memorization for Diffusion Models

Supplementary Material

We provide supplementary technical details and experimental results in the following sections:

- Sec. 7 presents a unified theoretical framework for InvMM within the context of diffusion model, including detailed analyses of noise distribution and prompt distribution reparameterization.
- Sec. 8 includes additional experimental details and results complementing to the main paper. Specifically, results in Sec. 8.5 provides more evidence to demonstrate the difference between membership and memorization.
- Sec. 9 gives a comprehensive ablation study on the prompt inversion in text-guided DMs. The experiments clarify the influence of temperature, CFG scale, inversion objective and adaptive algorithm on prompt inversion.
- Sec. 10 shows generation results using inverted noise and prompts on various datasets and models.

7. Additional technical description

7.1. A unified understanding

Similar to Eq. (2), the variational lower bound can be further lower bounded w.r.t. a conditional distribution $q_{\boldsymbol{\phi}}(\boldsymbol{c}, \boldsymbol{x}_0)$ with parameters $\boldsymbol{\phi}$. For text-to-image models, it is instantiated as a prompt distribution $q_{\boldsymbol{\phi}}(\boldsymbol{\omega}|\boldsymbol{x}_0)$.

Following similar expansion in Eqs. (3) and (4), we obtain the full inversion variational lower bound w.r.t. both condition and latent noise:

$$\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0) \geq -l_{de}(\boldsymbol{x}_0; \boldsymbol{\phi}, \boldsymbol{\varphi}) - l_{kl}(\boldsymbol{x}_0; \boldsymbol{\varphi}) - l_{cr}(\boldsymbol{x}_0; \boldsymbol{\phi}) \tag{13}$$

and

$$l_{de}(\boldsymbol{x}_0; \boldsymbol{\phi}, \boldsymbol{\varphi}) = -\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{c}|\boldsymbol{x}_0), q_{\boldsymbol{\varphi}}(\boldsymbol{\epsilon}_0|\boldsymbol{x}_0)} [\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1, \boldsymbol{c}) + \sum_{t=2}^{T} D_{\mathrm{KL}}(q_{\boldsymbol{\varphi}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{c}))] \tag{14}$$

$$l_{kl}(\boldsymbol{x}_0; \boldsymbol{\varphi}) = D_{\mathrm{KL}}(q_{\boldsymbol{\varphi}}(\boldsymbol{x}_T|\boldsymbol{x}_0) \parallel p(\boldsymbol{x}_T)) \tag{15}$$

$$l_{cr}(\boldsymbol{x}_0; \boldsymbol{\phi}) = D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{c}|\boldsymbol{x}_0) \parallel p(\boldsymbol{c})) \tag{16}$$

where l_{de} indicates the **d**enoising **e**rror, l_{kl} is a **KL** divergence and l_{cr} is a **c**ondition **r**egularization term. The three terms can be explained as the following:

1. l_{de} indicates how accurate the pretrained model denoises each $\boldsymbol{x}_t$ when the added noise is drawn from $q_{\boldsymbol{\varphi}}(\boldsymbol{\epsilon}_0|\boldsymbol{x}_0)$. If l_{de} is low enough, then for most noises $\boldsymbol{\epsilon}_0 \sim q_{\boldsymbol{\varphi}}(\boldsymbol{\epsilon}_0|\boldsymbol{x}_0)$ the model presents low denoising error, we can anticipate that the sampling trace starting at $\boldsymbol{\epsilon}_0 \sim q_{\boldsymbol{\varphi}}(\boldsymbol{\epsilon}_0|\boldsymbol{x}_0)$ will head towards $\boldsymbol{x}_0$ and finally generate $\boldsymbol{x}_0$.
2. l_{kl} is a normality regularizer. When l_{de} is optimized to a low level, the noise distribution $q_{\boldsymbol{\varphi}}(\boldsymbol{\epsilon}_0|\boldsymbol{x}_0)$ identifies a sensitive set of noises that will cause the generation of the training image $\boldsymbol{x}_0$. $p(\boldsymbol{x}_T)$ is the prior distribution, usually set to the standard Gaussian. In a sense, l_{kl} measures the diversity of the model's generation: When l_{kl} becomes zero, i.e., $q_{\boldsymbol{\varphi}}(\boldsymbol{\epsilon}_0|\boldsymbol{x}_0)$ is standard Gaussian, then low enough l_{de} means the model always generates $\boldsymbol{x}_0$ and loses generalization.
3. l_{cr} encourages the realistic feasibility of the condition distribution $q_{\boldsymbol{\phi}}(\boldsymbol{c}|\boldsymbol{x}_0)$. For example, if $p(\boldsymbol{c})$ is considered the distribution of natural language, then minimizing l_{cr} indicates that prompt $\boldsymbol{c} \sim q_{\boldsymbol{\phi}}(\boldsymbol{c}|\boldsymbol{x}_0)$ should be grammatically and semantically correct.

In the standard training of diffusion model (DM), l_{kl} is ignored because $q_{\boldsymbol{\varphi}}(\boldsymbol{\epsilon}_0|\boldsymbol{x}_0)$ is set to the standard Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$ such that $q_{\boldsymbol{\varphi}}(\boldsymbol{x}_T|\boldsymbol{x}_0)$ approximately equals $\mathcal{N}(\mathbf{0}, \mathbf{I})$, l_{kl} approximately equals zero. $q_{\boldsymbol{\phi}}(\boldsymbol{c}|\boldsymbol{x}_0)$ reduces to several captions coupled with the training image such that l_{cr} is also zero.

Our idea is to measure memorization by relaxation of the noise and prompt distribution so that the denoising error can be optimized low enough to replicate the target image. Based on this, the normality of the worst-case distribution of sensitive latent noise is used as a measure.

7.2. Reparameterization

Sampling from the noise and condition distribution in Eq. (14) is non-differentiable, we indirectly sample them.

When the noise distribution is $q_{\boldsymbol{\varphi}}(\boldsymbol{\epsilon}_0|\boldsymbol{x}_0)$ a multivariate Gaussian $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ with learnable mean $\boldsymbol{\mu}$ and diagonal variance $\boldsymbol{\sigma}^2$,

$$\boldsymbol{\epsilon}_0 = \boldsymbol{\epsilon}'\boldsymbol{\sigma} + \boldsymbol{\mu}, \boldsymbol{\epsilon}' \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{17}$$

For text-to-image DM, each token $\boldsymbol{\omega}_i$ (see Sec. 4.4) is reparameterized by

$$\tilde{\boldsymbol{\omega}}_{i,j} = \frac{\exp\left((\log \boldsymbol{\pi}_{i,j} + \boldsymbol{g}_{i,j})/\tau\right)}{\sum_{k=1}^{|\mathcal{V}|} \exp\left((\log \boldsymbol{\pi}_{i,k} + \boldsymbol{g}_{i,k})/\tau\right)} \tag{18}$$

where $\{\boldsymbol{g}_{i,j}\}, i = 1...M, j = 1...|\mathcal{V}|$ are i.i.d samples drawn from $\mathrm{Gumbel}(0, 1)$, τ is a temperature factor. When τ approaches 0, the smoothed sample $\tilde{\boldsymbol{\omega}}_i$ becomes one-hot.

After optimization, we can draw discrete prompt from the learned $q_{\boldsymbol{\phi}}(\boldsymbol{\omega}|\boldsymbol{x}_0)$ by:

$$\boldsymbol{\omega}_i = \arg\max_j \left[\log \boldsymbol{\pi}_{i,j} + \boldsymbol{g}_{i,j}\right] \tag{19}$$

	CIFAR-10	CelebAHQ	FFHQ	LAION
Learning Rate γ	1e-1	1e-1	1e-1	1e-1
Iteration T	2000	2000	2000	2000
Batch Size	32	16	16	16
Cycle C	50	10	10	50
Increment δ	1e-4	5e-4	5e-4	1e-3
Threshold ξ	1e-3	1e-3	1e-3	1e-3
Threshold β	1.0	1.0	1.0	1.0
Sampler	DDIM [53]	DDIM	DDIM	DDIM
DDIM Step	200	50	50	50
DDIM η	0	0	0	0
Optimizer	Adam [28]	Adam	Adam	Adam
SSCD Size	32×32	256×256	256×256	320×320

Table 2. Default hyperparameter settings.

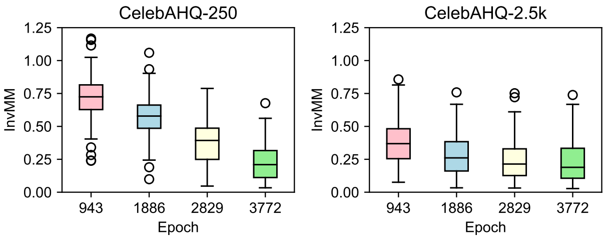


Figure 11. Training epoch vs. memorization on CelebAHQ.

8. Additional experiment details and results

8.1. Experiment setting

If not stated otherwise, the hyperparameters follow the default setting listed in Tab. 2, determined by previous investigation [51] and a few case studies. For CelebAHQ and FFHQ, all memorization scores in the plots are evaluated on CelebAHQ-250 and FFHQ-600. SD v3.5 utilizes three text encoders to represent the input prompt: CLIP-L, CLIP-G and T5. It is computationally expensive, so we only invert CLIP-L, with CLIP-G and T5 frozen. All the experiments in this paper are conducted on one NVIDIA A800 GPU.

8.2. Influence factors

Figure 11 shows the influence of training epochs on CelebAHQ. Larger training epochs lead to heavier memorization.

8.3. Detection

Experiment details. On CIFAR-10, the training loss metrics are calculated on the average loss of 16 random Gaussian noise and 50 timesteps uniformly sampled within the range [1, 1000]. Following van den Burg et al. [55]'s setting, M^{LOO} is estimated using a 10-fold cross-validation.

On LAION, the training loss metrics use 32 random Gaussian noise and 50 timesteps uniformly sampled within the range [1, 1000]. We implement Wen et al. [61] and Ren et al. [42] following their best performing settings. For GCG [64] attack, the number of optimizable tokens is 20. Each token is initialized to the special token $<|endoftext|>$. GCG is ran for 500 steps with a batch size of 128 and a top-k of 256. During optimization, the noise distribution is fixed to the standard Gaussian. A minibatch of 16 random noises is used to calculate the $\boldsymbol{x}_0$-loss every iteration. After each update, 8 random samples are generated. If any of them has a similarity with the target image no less than 0.5, the optimization process will stop.

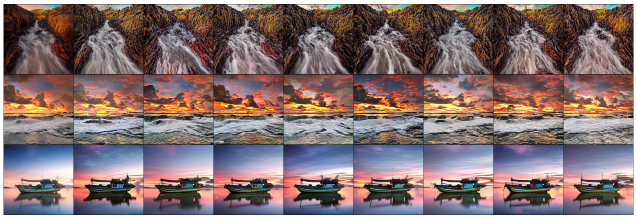


Figure 12. Not invertible examples in SD v1.4. The first column shows the corresponding training images.

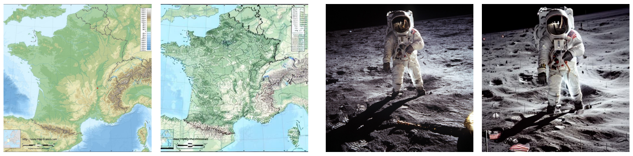
Calibration. SSCD similarity on the low-resolution CIFAR-10 yields many false positives (non-replication with high similarity, early stopped) and false negatives (replication with low similarity, not early stopped), although it works well on the other three high-resolution datasets. We perform a manual review on the false samples and correct their early stop timesteps and scores. All the other experimental results are unhandled.

Invertibility. More than half of the images in CIFAR-10 are not invertible even if we set $\lambda = 0$ all the time. This is probably because the DDPM (34.21 M) trained on CIFAR-10 has limited capacity to memorize every training image. Images in CelebAHQ and FFHQ are all invertible. As a comparison, the LDMs trained on them have larger capacity (314.12 M). There are also some images not invertible in Stable Diffusion. However, setting $\lambda = 0$ could invert them almost perfectly. The invertibility of an image is specific to a set of hyperparameter settings. Although it is possible to invert everything in SD, we regard samples not invertible in our setting "insignificant", as compared to those that are easy to invert. Figure 12 shows three examples that are not invertible in SD v1.4 from the *normal* subset, as judged by SSCD. It shows that it is quite subjective to judge the similarity between images. Human may regard such cases as replication as the generated images mimic the style of training image or else. From a technical perspective, we leave this problem untouched and follow the decision of SSCD.

Adversarial prompts. Figure 13 shows additional examples of adversarial prompts found by GCG for images from the suspicious set, which have lower InvMM scores.

Setting	CelebAHQ-250	CelebAHQ-2.5k	FFHQ-600	FFHQ-6k	CelebAHQ	FFHQ
Default	N/A(0)	N/A(0)	N/A(0)	N/A(0)	N/A(0)	N/A(0)
Epoch ×2	0.059(9)	N/A(0)	N/A(0)	N/A(0)	-	-
Epoch ×3	0.326(120)	N/A(0)	0.000(1)	N/A(0)	-	-
Epoch ×4	0.567(181)	0.000(3)	0.000(4)	N/A(0)	-	-
Duplicate ×5	-	0.182(65)	-	0.000(2)	-	-
Duplicate ×10	-	0.852(237)	-	0.192(186)	-	-
Duplicate ×15	-	0.937(245)	-	0.401(323)	-	-
Duplicate ×20	-	0.819(231)	-	0.434(375)	-	-

Table 3. Performance of collating InvMM with nearest neighbor test. Each result refers to IoU(Number of replicated training images).



Prompt: <|endoftext|><|endoftext|>mapped<|endoftext|>german<|endoftext|>upon öhercules major <|endoftext|>françneu<|endoftext|><|endoftext|><|endoftext|><|endoftext|>gigiincluded<|endoftext|>

Prompt: <|endoftext|>et <|endoftext|><|endoftext|><|endoftext|><|endoftext|>space<|endoftext|>testanalyzing moonshouldn <|endoftext|>teamusa <|endoftext|><|endoftext|><|endoftext|><|endoftext|><|endoftext|><|endoftext|>

Figure 13. Additional examples of adversarial prompts found by GCG.

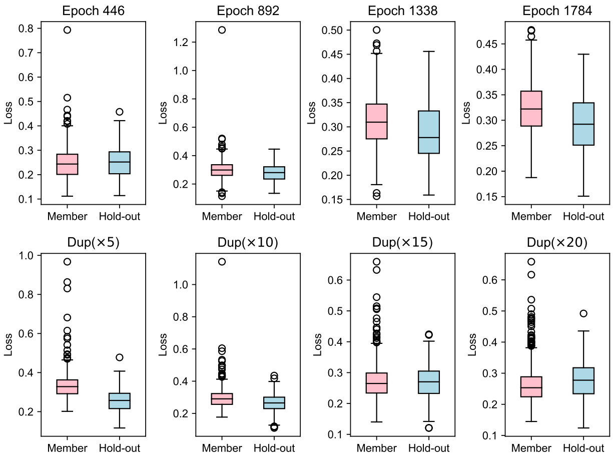


Figure 14. Comparison of replication loss for members and hold-out samples.

8.4. Collate InvMM with nearest neighbor test

We provide another quantitative metric to collate InvMM with the nearest-neighbor test: if a training image is replicated by its nearest neighbors in randomly generated samples, then its InvMM should be small among a list of training images, and vice versa. Let S_{nn} be the set of images that expose replication in randomly generated samples, and S_{InvMM} be the same-size set of images with the lowest InvMMs in a list of images. We define the consistence between InvMM and nearest-neighbor test as the Intersection over Union (IoU) between S_{InvMM} and S_{nn}:

$$\text{IoU} = \frac{|S_{\text{InvMM}} \cap S_{\text{nn}}|}{|S_{\text{InvMM}} \cup S_{\text{nn}}|}, |S_{\text{nn}}| > 0 \tag{20}$$

Under the setting in Sec. 5.4, InvMM achieves an IoU of 0.817 on CIFAR-10 and 1.0 on LAION. The results on CelebAHQ and FFHQ are summarized in Tab. 3. 10k random samples are used to obtain the results. A random sample of cosine similarity larger than 0.5 with any training sample is considered a replication. InvMM presents consistence with the results of nearest neighbor test, indicating its potential to expose risk of training image leakage, especially when a large number of training images are prone to leakage. The 10k samples are not an adequate sampling of the large latent space (64×64×3), the performance can be refined with a larger scale of evaluation.

8.5. Comparison to membership

Experiment details. Figure 10 is evaluated with LDM trained on FFHQ-6k. FFHQ-600 is used as the member set and another 600 images not contained in FFHQ-6k constitute the hold-out set. The statistic variant of SecMI is implemented following their official settings, with $t_{\text{SEC}} = 100$ and $k = 10$.

Results. Figure 14 provides quantitative evidence that different samples could be replicated with different levels of training loss. The replication loss is calculated over the inverted latent noise distribution. It shows that although hold-out samples have a larger training loss over the $\mathcal{N}(\mathbf{0}, \mathbf{I})$, supporting effective membership inference, it is still possible to replicate some of them by reducing their training loss to a level higher than some of the members.

We visualize the top-2 false positives and false negatives in Fig. 15. The results show that hold-out samples are also replicated near identically.

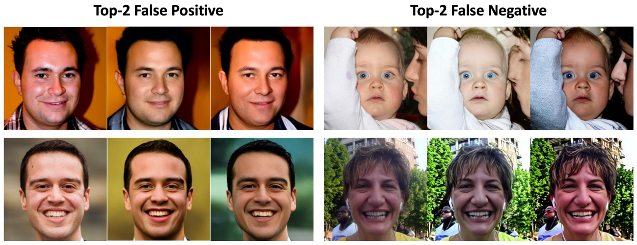


Figure 15. Top-2 false positives with lowest InvMM from the hold-out set and top-2 false negatives with highest InvMM from the member set. The results come from LDM trained on FFHQ-6k for 1784 epochs.

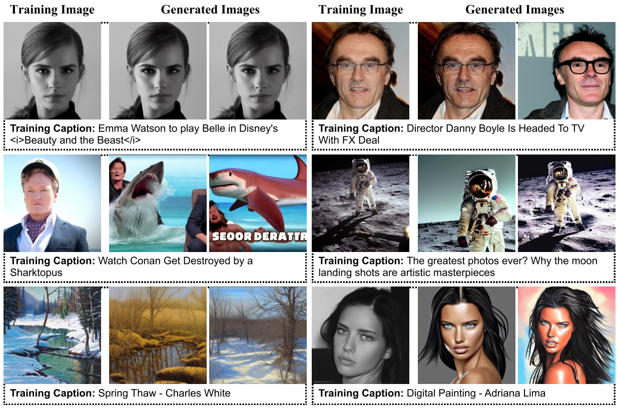

Training Image
Generated Images
Training Image
Generated Images
Training Caption: Emma Watson to play Belle in Disney's <i>Beauty and the Beast</i>
Training Caption: Director Danny Boyle Is Headed To TV With FX Deal
SEOOR DERATTR
Training Caption: Watch Conan Get Destroyed by a Sharktopus
Training Caption: The greatest photos ever? Why the moon landing shots are artistic masterpieces
Training Caption: Spring Thaw - Charles White
Training Caption: Digital Painting - Adriana Lima

Figure 16. Examples in confirmed, suspicious and normal subsets from top to down. In each block, the right two columns show generated images using their training captions but different initial noises.

The result further highlights the difference between membership and memorization: training loss does not completely determine data replication.

9. Ablation study on prompt inversion

This section elaborates the influence of several factors for prompt inversion in SD, including the temperature τ in Gumbel-Softmax, the Classifier-Free Guidance (CFG) [21] scale γ and the advantage of predicting the image rather than noise for inversion (Eq. (12)). For this goal, the noise distribution is temporarily fixed to the standard Gaussian. Experiment results will show that heavily memorized images can also be uncovered in this setting.

9.1. Dataset

We use the aforementioned three LAION subsets for evaluation. Figure 16 presents examples from the three subsets, together with images generated using their training captions.

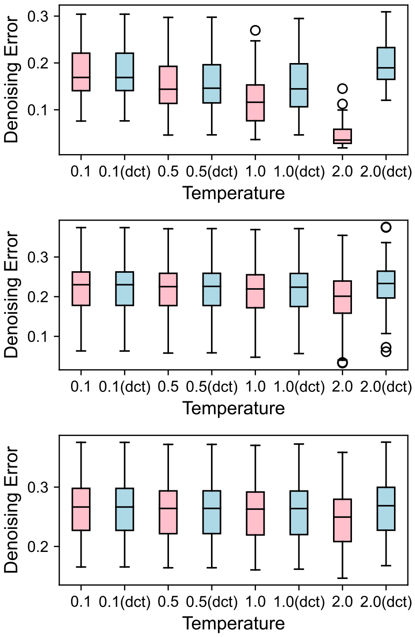


Figure 17. The distribution of denoising error of the confirmed, suspicious and normal subsets. ”dct” means plus discretization.

9.2. Experiment setup

We utilize SD v1.4 for evaluation, which uses CLIP [40] to encode input prompts and generate high-resolution 512 × 512 images. The text encoders by default accepts a maximum length of 77 tokens, in which the first and last tokens are padded tokens indicating the start and end of a prompt. The rest 75 tokens are all optimized in our experiments. During optimization, the parameters $\boldsymbol{\theta}$ of diffusion models are fixed. We optimize for 500 iterations with a constant learning rate of 0.2. ϕ is initialized to $\mathbf{0}$ at the begining of optimization.

9.3. Influence of temperature

Hard prompt inversion to exactly reconstruct certain images is a challenging problem as it requires to search over a large and discrete space consisting of tens of thousands of tokens (49408 in CLIP). We have found that the convergence of inversion relies on appropriate choice of the temperature τ in Gumbel-Softmax smoothing. With τ approaching 0, $\tilde{\boldsymbol{\omega}}$ drawn from the prompt distribution approaches one-hot and accurately matches a token, while it is difficult to optimize through gradient decent. Larger τ provides a smoother landscape of the target loss function and thus is easier to optimize. However, the smoothed $\tilde{\boldsymbol{\omega}}$ cannot directly correspond to some tokens. Discretizing them anyway (Eq. (19)) might not preserve the same effectiveness as the smooth counterparts.

Denoising error. We first analyze the denoising error of the inverted prompt distribution when optimized with different temperatures. We consider 4 settings for the temperature to be either 0.1, 0.5, 1 or 2, as well as whether to discretize the smoothed tokens $\tilde{\boldsymbol{\omega}}$ to $\boldsymbol{\omega}$. After optimization, random prompts and noises can be drawn from $q_{\boldsymbol{\phi}}(\boldsymbol{\omega}|\boldsymbol{x}_0)$ and $\mathcal{N}(\mathbf{0}, \mathbf{I})$. For each image, we randomly sample 10 prompts and 10 noises for each sampled prompt, resulting in 100 prompt-noise pairs. The denoising error is estimated using 50 timesteps uniformly sampled within the range $[1, 1000]$ and averaged over the 100 prompt-noise pairs.

The results can be seen in Fig. 17. For any type of image from different groups, higher temperatures lead to lower denoising errors on average, indicating a more adequate optimization. However, meanwhile, plus token discretization worsens the effectiveness.

Convergence. Figure 18 shows the denoising errors at each optimization step of the 6 example images in Fig. 16. For the assessment of convergence, we draw a baseline denoising error calculated using the training caption of each image. As can be observed, large temperatures induce better convergence and the difference between $\tilde{\boldsymbol{\omega}}$ and $\boldsymbol{\omega}$ becomes prominent. Figure 19 illustrates the generation results using prompts sampled from the learned distribution, with a CFG scale of 7. As can be seen, with $\tau = 2$, the inverted prompts are able to replicate the training images for the 4 examples from the confirmed and suspicious groups, the two from the suspicious group are newly found through our analysis. However, $\tau = 2$ plus discretization produces completely irrelevant images. Lower temperatures 0.5 and 1.0 present more consistent generation between smooth and discrete prompts, while they only produce similar images to the training ones, showing analogous content, color, etc. The smallest $\tau = 0.1$ fails to capture the main content of the training images but remains the best consistency for discretization.

Prompt distribution. Figure 20 depicts the density distribution of the entropy $-\sum_{j=1}^{|\mathcal{V}|} \boldsymbol{\pi}_{i,j} \log \boldsymbol{\pi}_{i,j}$ of the learned prompt categorical distributions. When $\tau = 2$, most tokens follow a high entropy distribution, which means that they are well smoothed and take an interpolation of hard tokens. In contrast, smaller temperatures produce more sharp distributions, while less effective as large temperature for inversion.

Conclusion. For the goal of effective analysis, we adopt a compromise setting with the temperature τ of 2.0 and without discretization, to reach adequate optimization. Although this violates the goal of inverting realistic prompts, it is reasonable and enough for developers to analyze the vulnerability of their models. Note that it still offers a certain level of restriction to the learned soft prompts, as the Gumbel-Softmax approximation together with a linear combination of pretrained token embeddings bound the smoothed tokens $\tilde{\boldsymbol{\omega}}$ in the convex hull of the pretrained tokens.

9.4. CFG Scale

In additional, we sweep the CFG scale γ from 0 to 7 with interval 1 to study its influence. $\gamma = 0$ indicates unconditional generation and $\gamma = 1$ indicates conditional generation without penalizing unconditional prediction. The generation results of the examples in Fig. 16 are shown in Fig. 21.

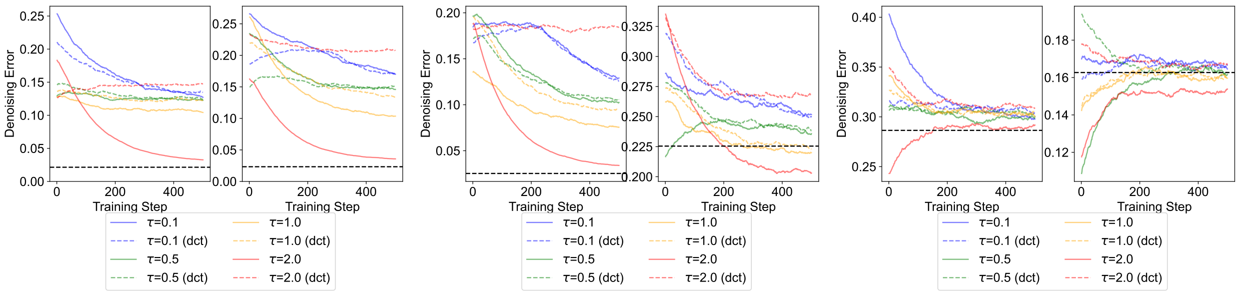
It can be observed that the generated images with $\gamma = 0$ are quite random because they only depend on the random initial noises. When $\gamma \geq 1$, for heavily memorized images in the confirmed and suspicious group, the generation results progressively converge to the training images. At times the generated images with small CFG scale only resemble the training images but are not eidetic, e.g., the 2rd to 4th rows. Nonetheless, we also discovered perfect replication for these examples with a small $\gamma = 1$ generated using other different initial noises. This indicates that the extent to which different training images are memorized varies, and, moreover, a relatively low-level training time memorization ($\gamma = 1$) can be amplified by sampling-time options such as larger γ. Given that we optimize the prompts w.r.t. the conditional model ($\gamma = 1$), it demonstrates that training data leakage roots in the conditional model.

In addition, a gradual sharpening can be observed in the generated images as the guidance scale increases. As we optimize w.r.t. the conditional model, i.e., $\gamma = 1$, it is of enough denosing accuracy to generate an training image with relatively lower scales. Enlarging the conditional scale, however, results in excessive alignment with the input prompt. In contrast, for the images in the normal group (see the last two rows of Fig. 21), as the inverted prompt distribution cannot fully capture its complete content, generation with $\gamma = 1$ is somewhat fuzzy. It thus benefits from an increase of γ for higher quality.

Conclusion. Training data memorization can be amplified by CFG scale. As we consider the worst-case memorization in this paper, we count in the replication caused by any CFG scale from 1 to 7.

9.5. Optimization Objective

As we adopt the modified $\boldsymbol{x}_0$-prediction objective different from the original $\boldsymbol{\epsilon}_0$-prediction objective of the diffusion models used in our experiments, we verify the effectiveness of $\boldsymbol{x}_0$-prediction over $\boldsymbol{\epsilon}_0$-prediction for inversion. We evaluate using the images in the *confirmed* set to determine if the $\boldsymbol{\epsilon}_0$-prediction can successfully replicate them. Figure 22 shows the inversion results of $\boldsymbol{\epsilon}_0$-prediction. Inverison with $\boldsymbol{\epsilon}_0$-prediction is much unstable compared to $\boldsymbol{x}_0$-prediction, which demonstrates the importance of reweighting denoising error at different timesteps. More specifically, the later timesteps at training time (ealier at sampling time) tend to



(a) Confirmed Examples

(b) Suspicious Examples

(c) Normal Examples

Figure 18. Training denoising error of examples from each group under different temperatures, smoothed via exponential moving average with a momentum of 0.99. ”dct” means plus discretization. The black dashed line is the baseline error calculated using their training captions over 1600 randomly sampled Gaussian noises.

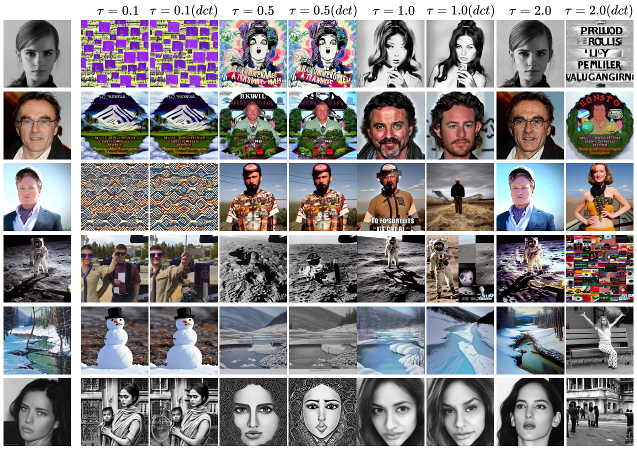


Figure 19. Generation results of different temperatures. The first column shows the corresponding training images.

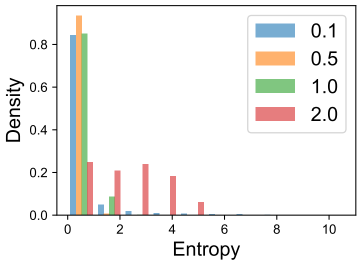


Figure 20. Entropy density distribution of the prompt categorical distribution.

shape the large scale image features [22], e.g., shape, object. Therefore, it would be beneficial to upweight the later timesteps by x_0-prediction to more accurately guide diffusion models to generate the corresponding training images.

Conclusion. Although not aligning with the original training objective, x_0-prediction is more stable for inversion.

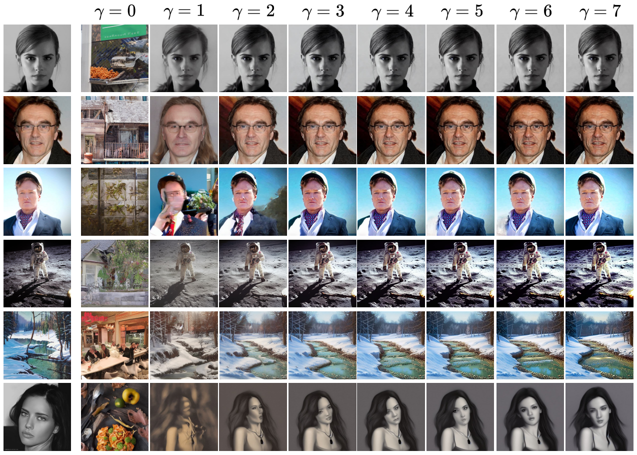


Figure 21. Generation results of different classifier-free guidance scales. The first column shows the corresponding training images.

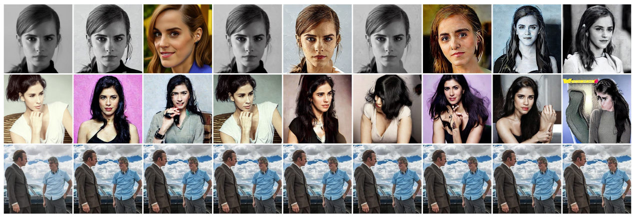


Figure 22. Inversion results of ϵ_0-prediction. The first column shows the corresponding training images.

9.6. Adaptive algorithm

A comparison of the (1) training error (using training captions), (2) inversion error with only prompt distribution learned, (3) inversion error with both prompt and noise distributions learned, fixing $\lambda = 1$ and (4) inversion error with both prompt and noise distributions learned, dynamically adjusting λ by Algorithm 1, is shown in Fig. 23. Compared to the training error, (2) only reduces that of heavily memorized images for which the input prompts plays a cru-

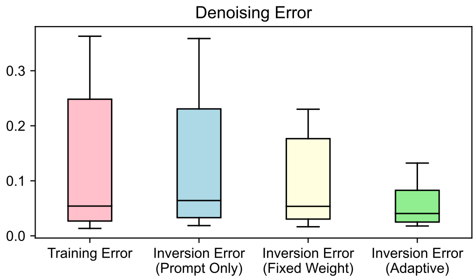


Figure 23. Comparison of inversion denoising error under different settings on SD v1.4.

cial role. (3) further reduces the denoising error but cannot work for all samples. Algorithm 1 can successfully reduce the denoising error of any training samples by adaptively adjusting the weight of normality regularization.

10. More generation results

Figures 24 to 33 show more generated images using inverted noise vectors (and prompts).

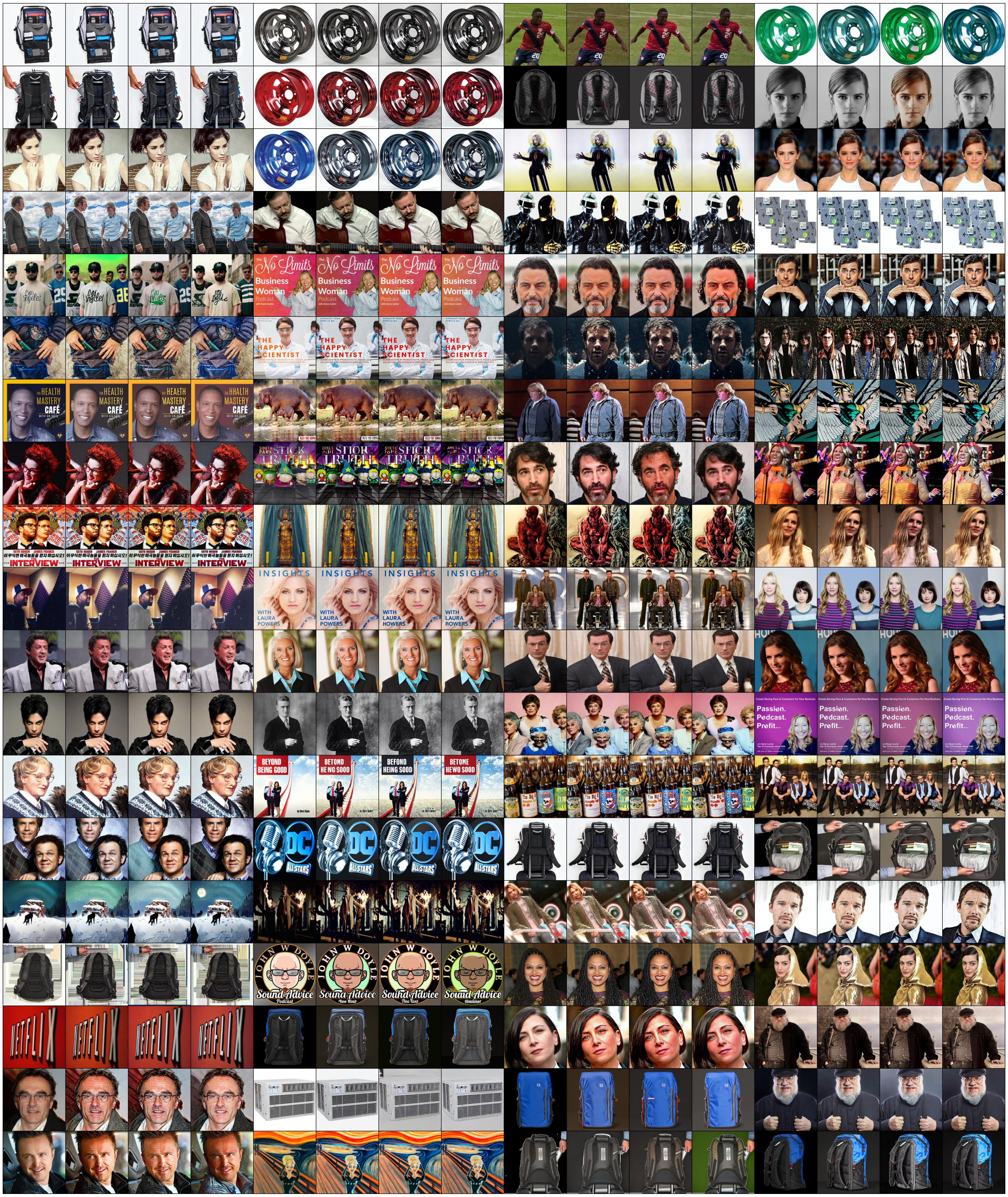


Figure 24. Random samples of SD v1.4 inversion on the *confirmed* subset. The first column shows the corresponding training images.

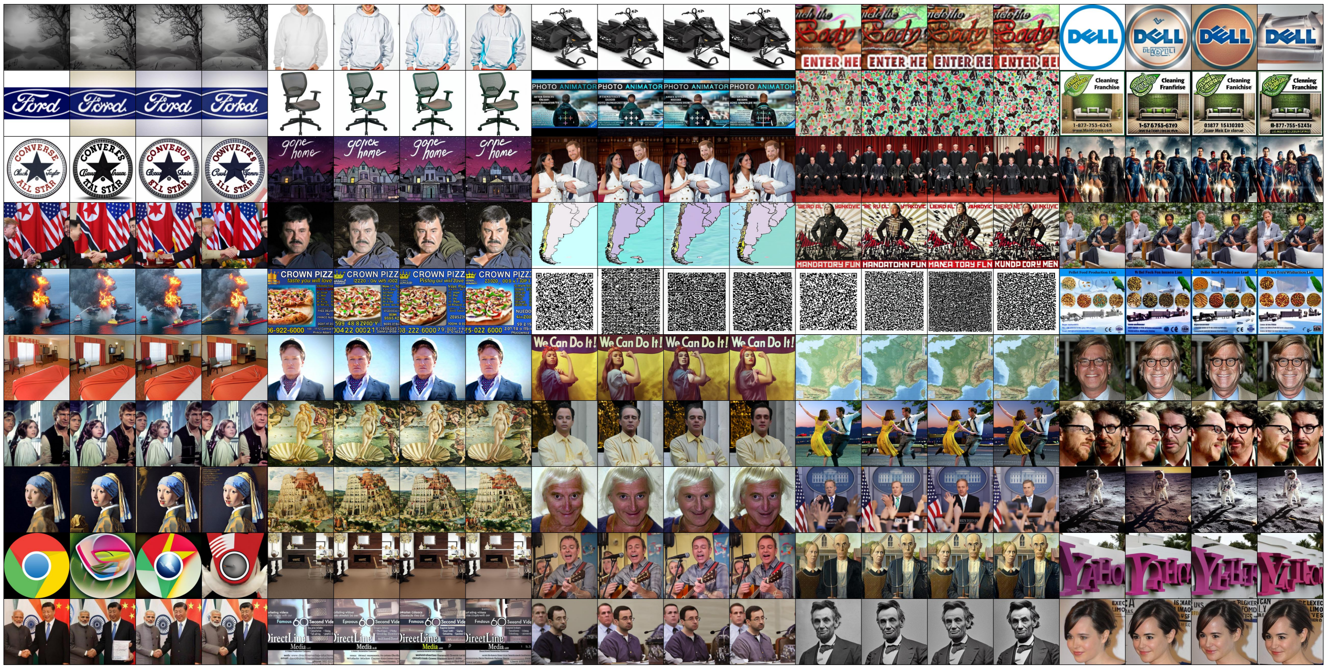


Figure 25. Random samples of SD v1.4 inversion on the *suspicous* subset. The first column shows the corresponding training images.

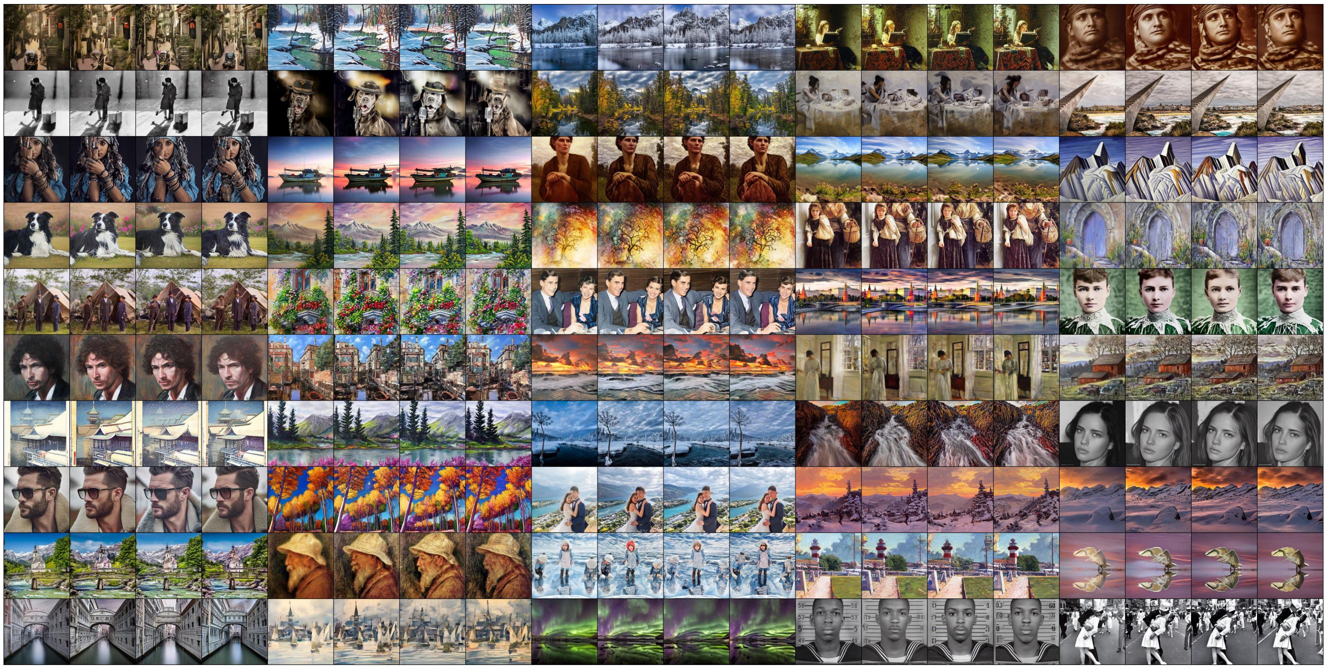


Figure 26. Random samples of SD v1.4 inversion on the *normal* subset. The first column shows the corresponding training images.

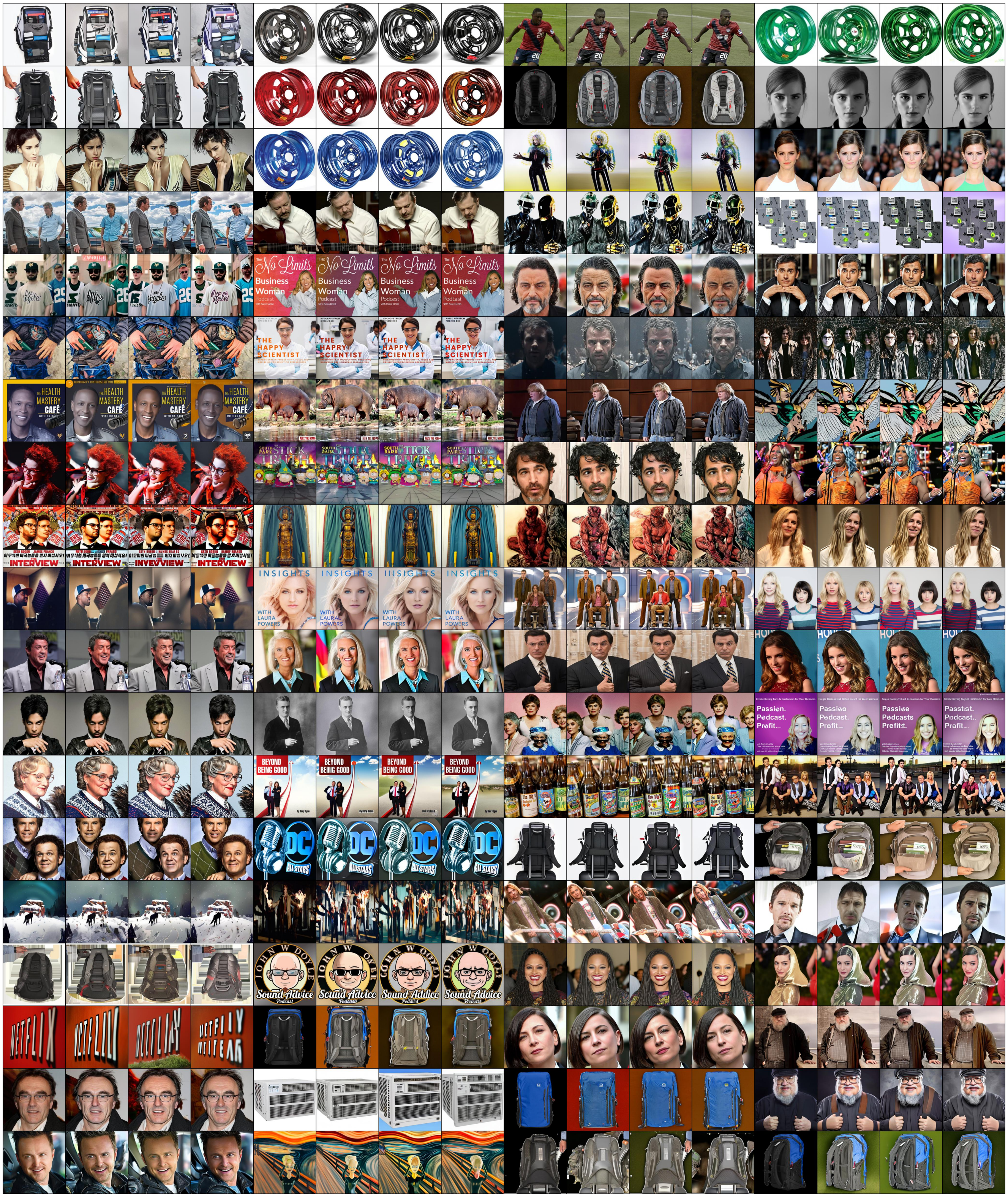


Figure 27. Random samples of SD v2.1 inversion on the *confirmed* subset. The first column shows the corresponding training images.

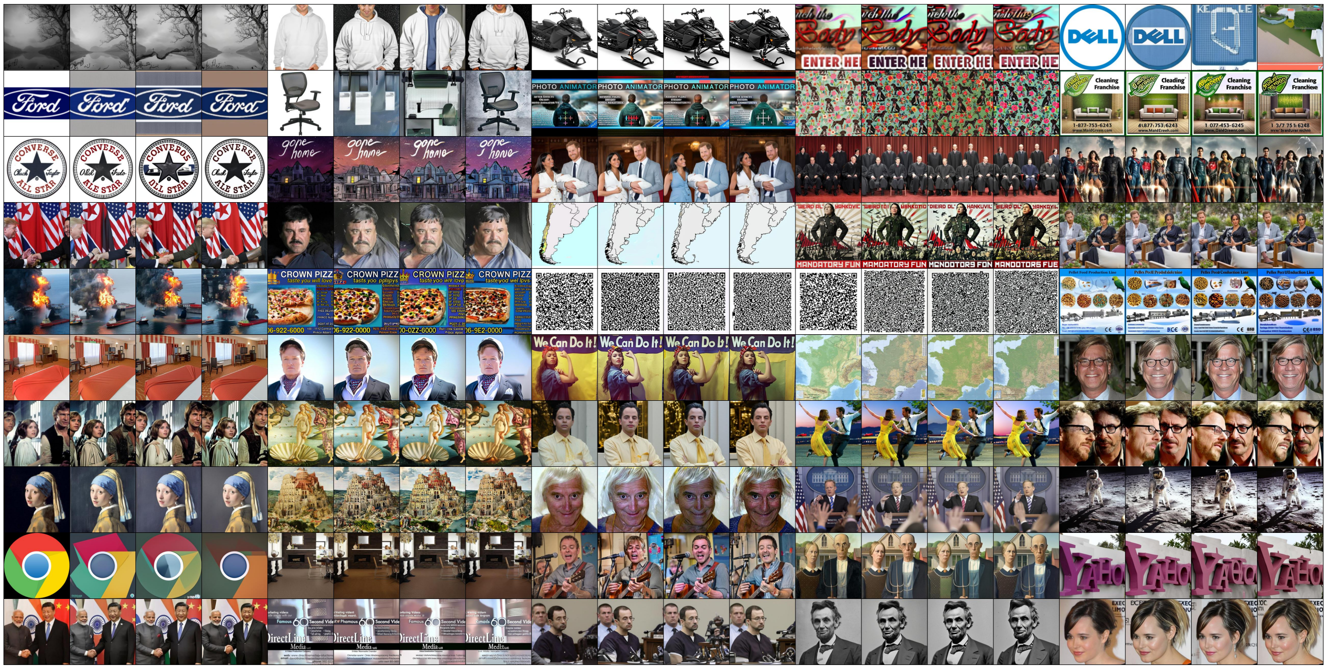


Figure 28. Random samples of SD v2.1 inversion on the *suspicous* subset. The first column shows the corresponding training images.

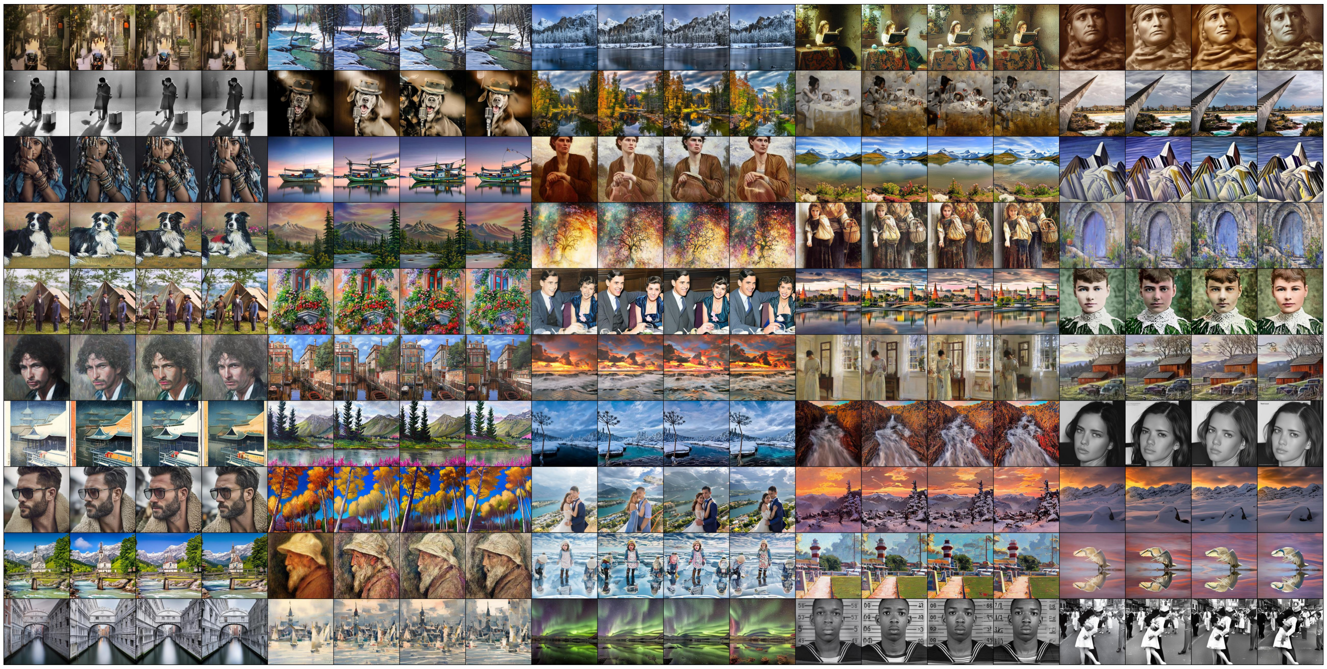


Figure 29. Random samples of SD v2.1 inversion on the *normal* subset. The first column shows the corresponding training images.

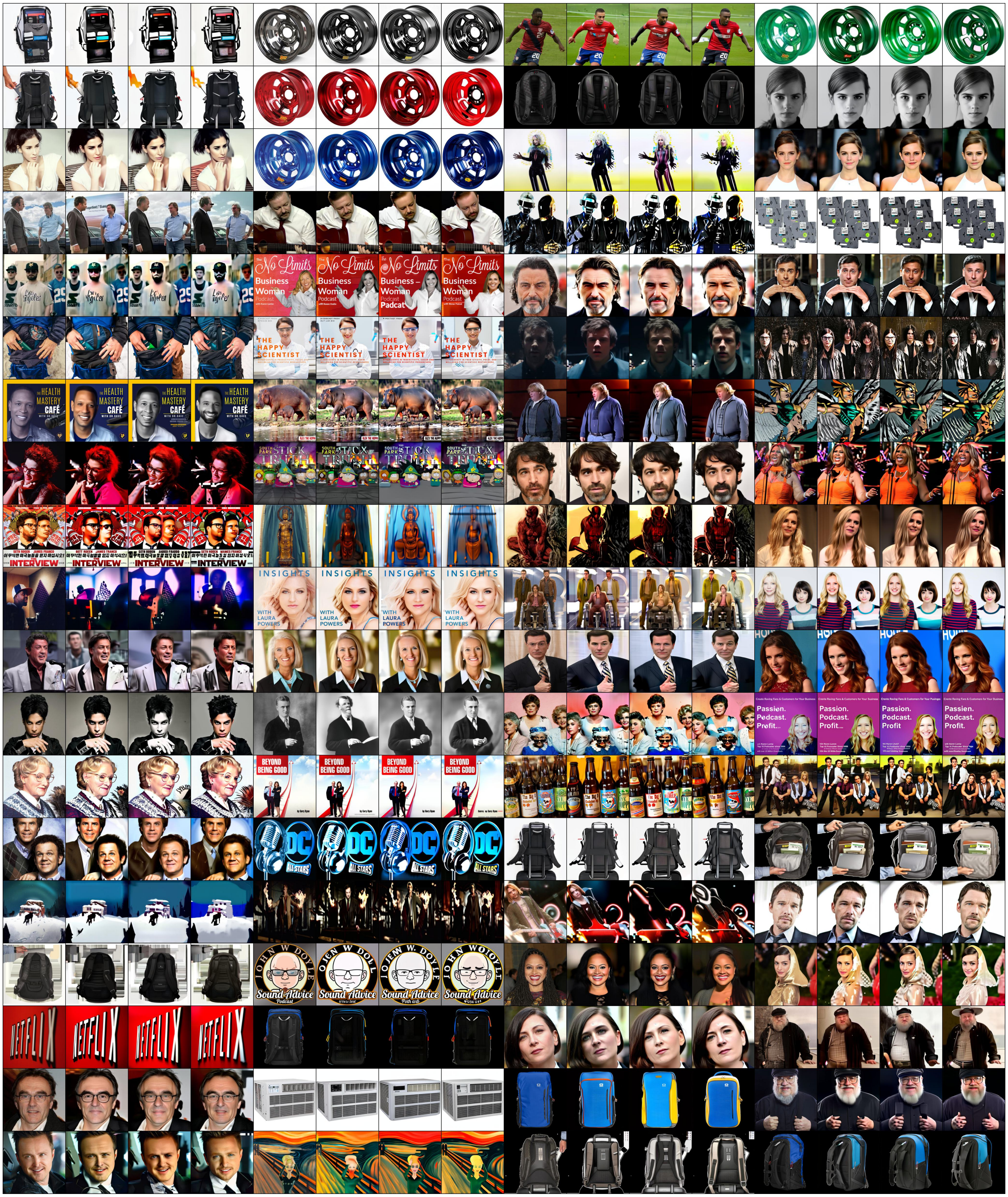


Figure 30. Random samples of SD v3.5 inversion on the *confirmed* subset. The first column shows the corresponding training images.

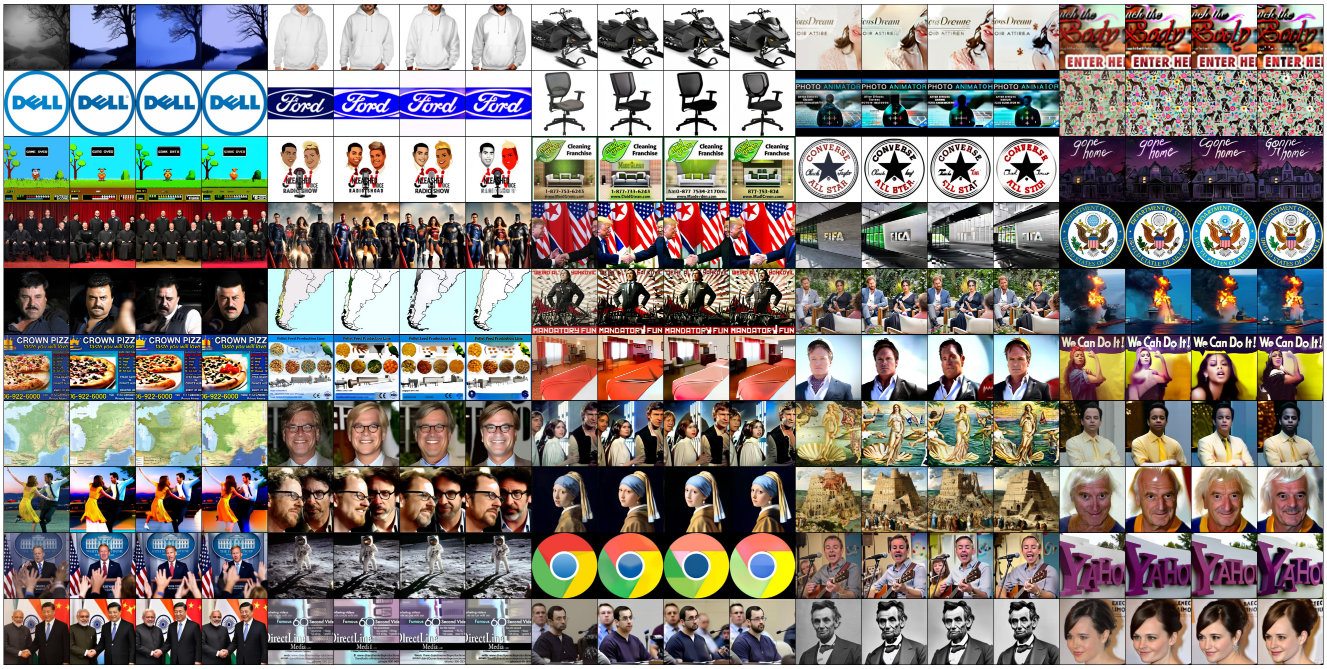


Figure 31. Random samples of SD v3.5 inversion on the *suspicous* subset. The first column shows the corresponding training images.

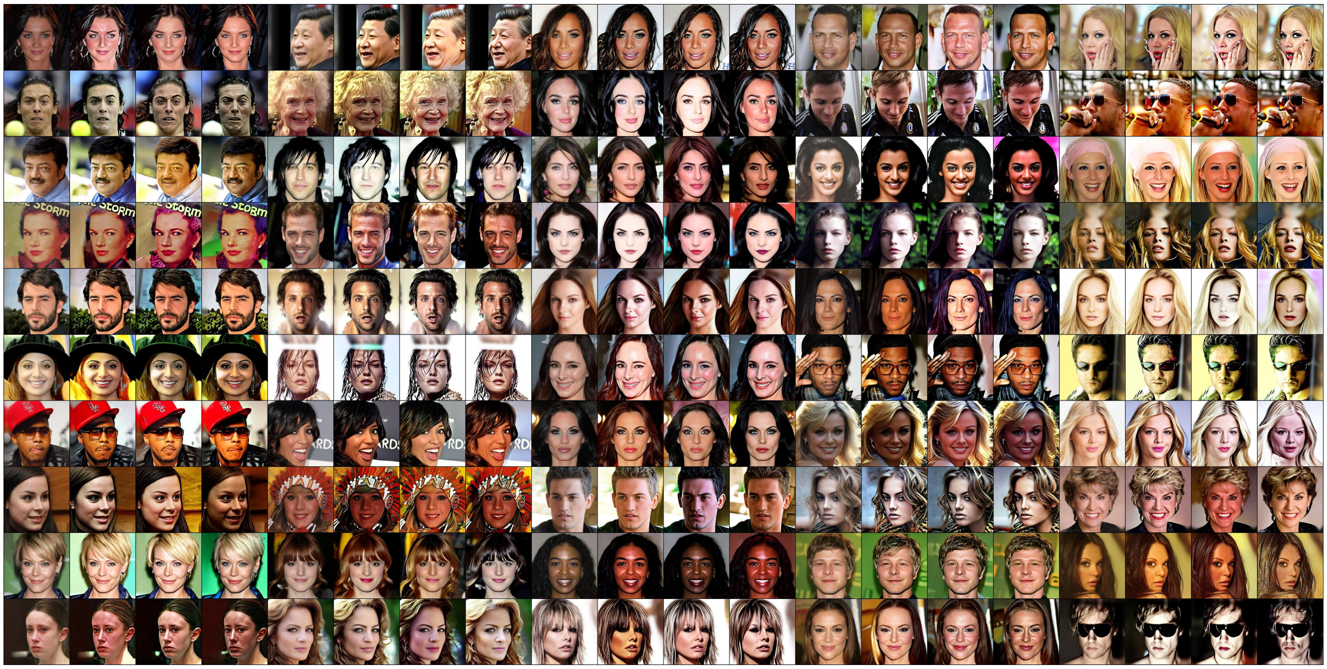


Figure 32. Random samples of LDM inversion on CelebAHQ. The first column shows the corresponding training images.

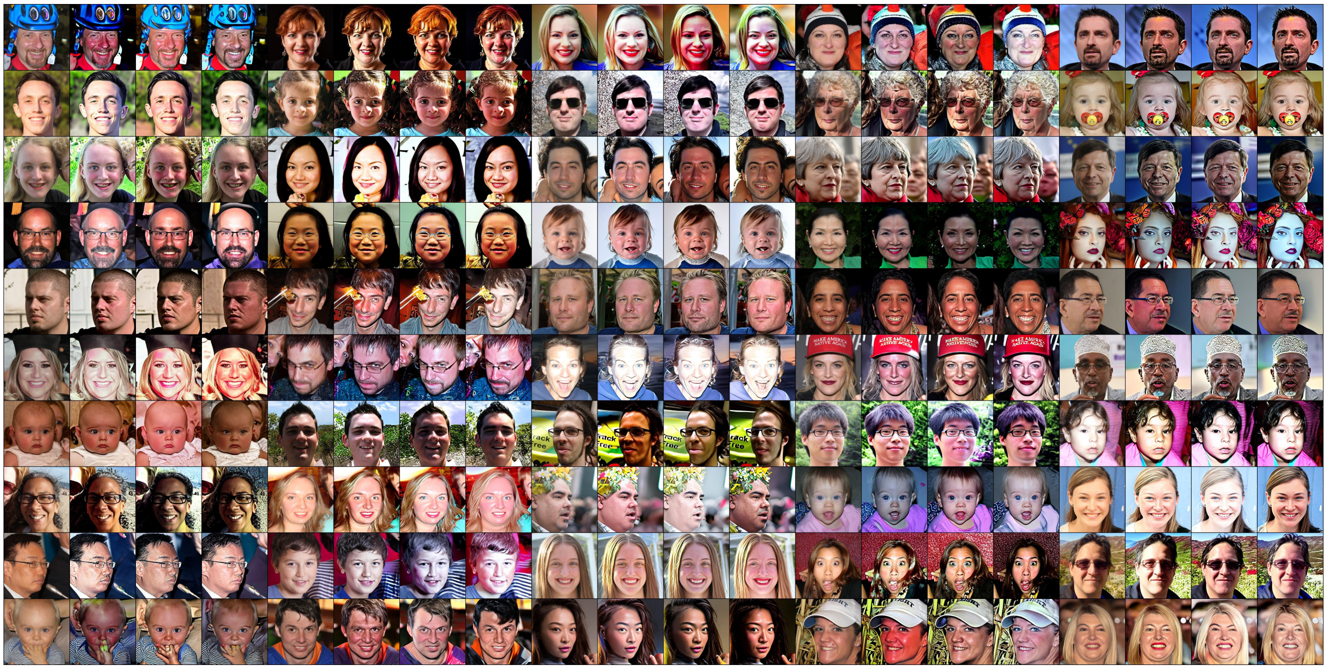


Figure 33. Random samples of LDM inversion on FFHQ. The first column shows the corresponding training images.