Trustworthy Diffusion Models & Diffusion Models for Trustworthy ML

Tianyu Pang

Sea AI Lab

Big Bang of Diffusion Models!

Diffusion Models in 2020 (Nonequilibrium Thermodynamics)

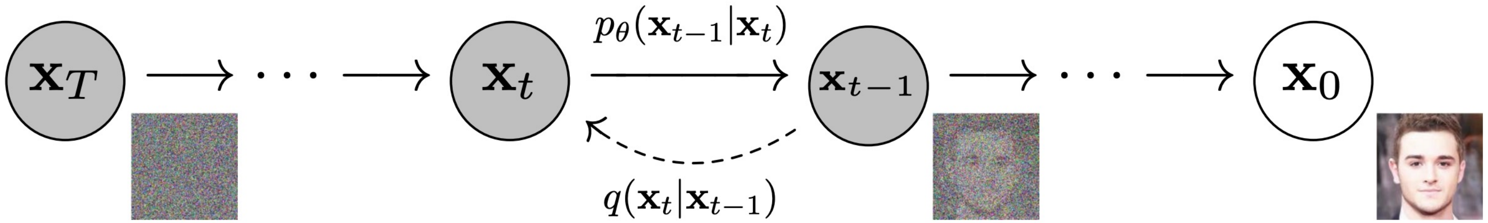


Figure 2: The directed graphical model considered in this work.

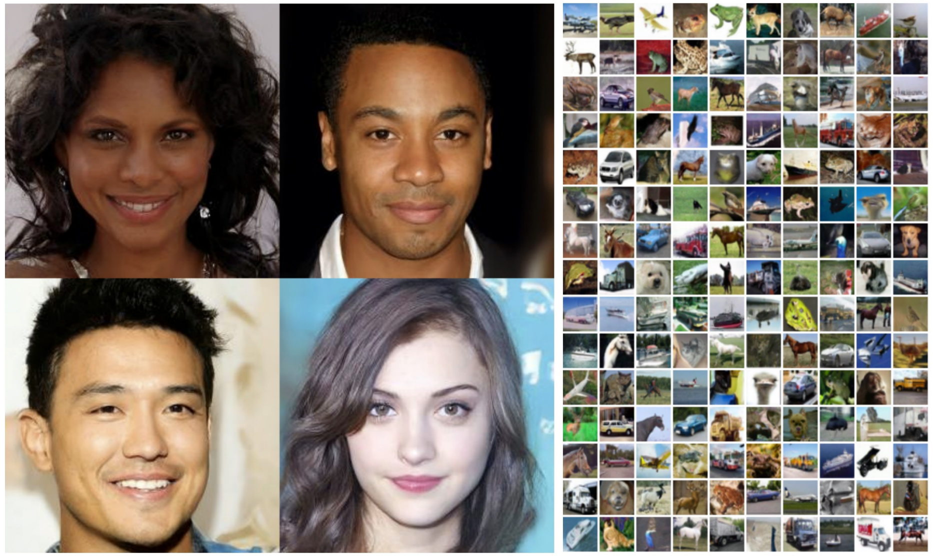


Figure 1: Generated samples on CelebA-HQ 256 × 256 (left) and unconditional CIFAR10 (right)

Figure 3: LSUN Church samples. FID=7.89

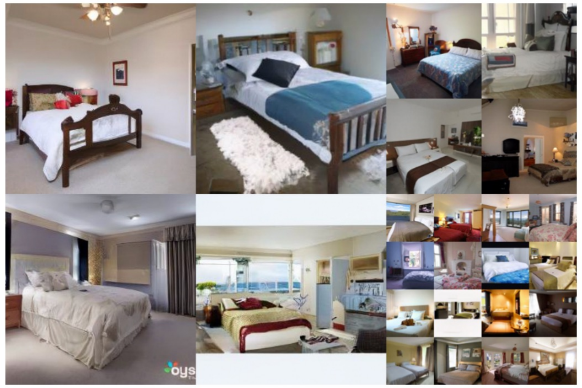


Figure 4: LSUN Bedroom samples. FID=4.90

[1] Sohl-Dickstein et al. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. ICML 2015

[2] Ho et al. Denoising Diffusion Probabilistic Models. NeurIPS 2020

Diffusion Models in 2020 (Annealed Langevin Dynamics)

Algorithm 1 Annealed Langevin dynamics.

Require: $\{\sigma_i\}_{i=1}^{L}, \epsilon, T$.

Initialize $\tilde{\mathbf{x}}_0$
for $i \leftarrow 1$ to L **do**
  $\alpha_i \leftarrow \epsilon \cdot \sigma_i^2 / \sigma_L^2$ ▷ α_i is the step size.
  for $t \leftarrow 1$ to T **do**
    Draw $\mathbf{z}_t \sim \mathcal{N}(0, I)$
    $\tilde{\mathbf{x}}_t \leftarrow \tilde{\mathbf{x}}_{t-1} + \dfrac{\alpha_i}{2} \mathbf{s}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}_{t-1}, \sigma_i) + \sqrt{\alpha_i}\, \mathbf{z}_t$
  end for
  $\tilde{\mathbf{x}}_0 \leftarrow \tilde{\mathbf{x}}_T$
end for
return $\tilde{\mathbf{x}}_T$

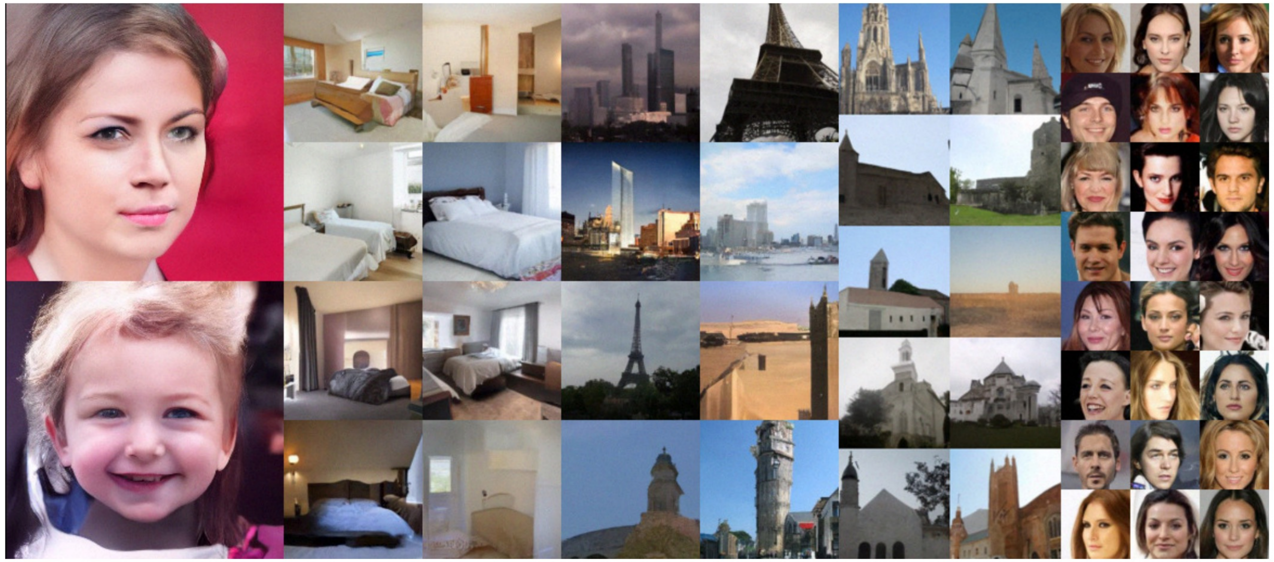


Figure 1: Generated samples on datasets of decreasing resolutions. From left to right: FFHQ 256×256, LSUN bedroom 128×128, LSUN tower 128×128, LSUN church_outdoor 96×96, and CelebA 64×64.

EBMs (BP through CNNs) → Score-based models (U-Nets)

[3] Song & Ermon. Generative Modeling by Estimating Gradients of the Data Distribution. NeurIPS 2019
[4] Song & Ermon. Improved Techniques for Training Score-Based Generative Models. NeurIPS 2020

Diffusion Models in 2021 (Stochastic Differential Equations)

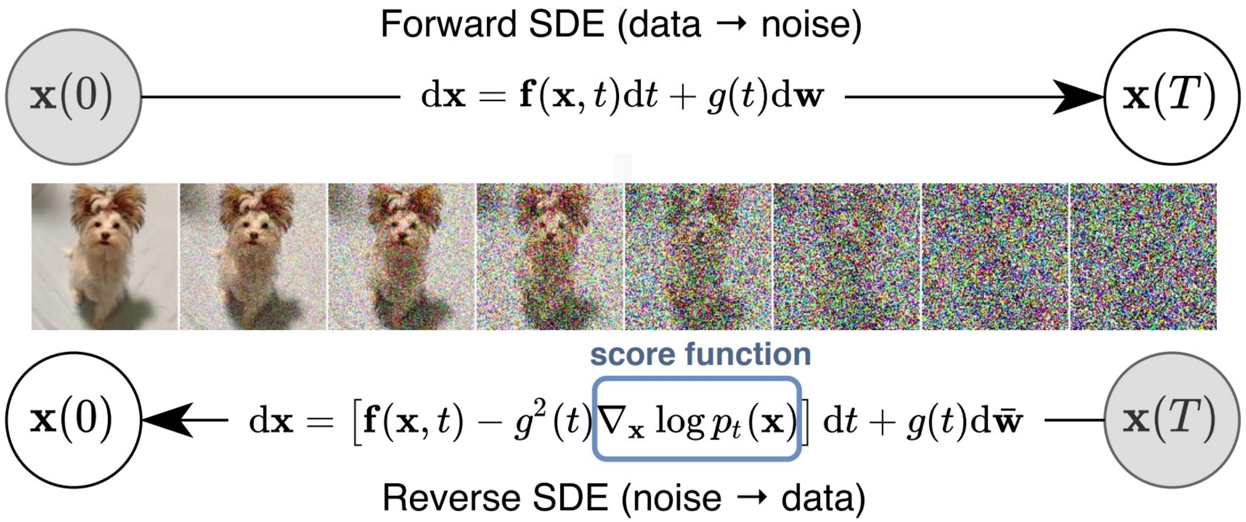


Figure 1: **Solving a reverse-time SDE yields a score-based generative model.** Transforming data to a simple noise distribution can be accomplished with a continuous-time SDE. This SDE can be reversed if we know the score of the distribution at each intermediate time step, $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$.

- Drift coefficient f
- Diffusion coefficient g

[5] Song et al. Score-Based Generative Modeling through Stochastic Differential Equations. ICLR 2021

Diffusion Models in 2021 (They Beat GANs)

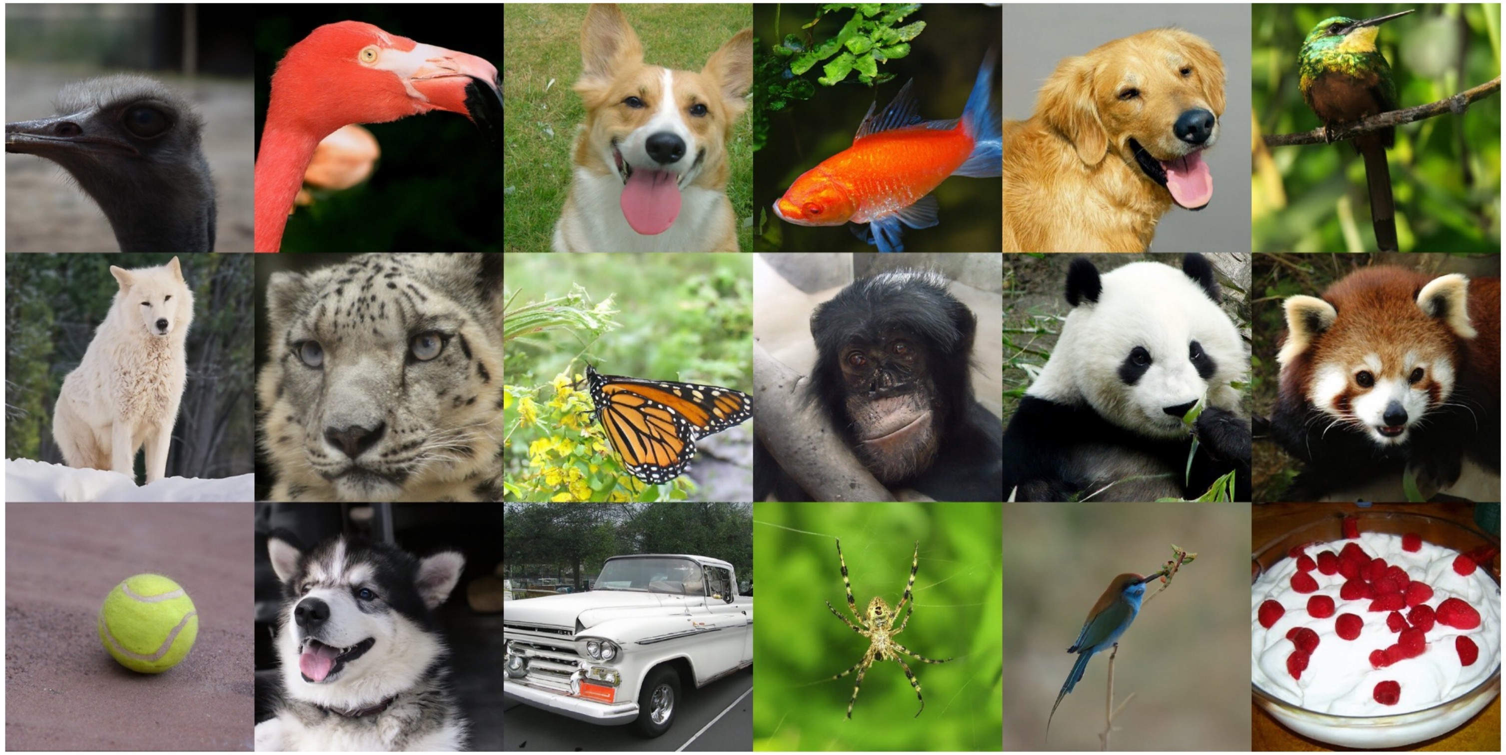


Figure 1: Selected samples from our best ImageNet 512×512 model (FID 3.85)

- Finding better architecture through ablation (ablated diffusion model, ADM)
- Classifier guidance for improving conditional generation

[6] Dhariwal & Nichol. Diffusion Models Beat GANs on Image Synthesis. NeurIPS 2021

Diffusion Models in 2022 (Text-to-Image Generation)


"a hedgehog using a calculator"


"a corgi wearing a red bowtie and a purple party hat"


"robots meditating in a vipassana retreat"


"a fall landscape with a small cottage next to a lake"


"a surrealist dream-like oil painting by salvador dalí of a cat playing checkers"


"a professional photo of a sunset behind the grand canyon"


"a high-quality oil painting of a psychedelic hamster dragon"


"an illustration of albert einstein wearing a superhero costume"


"zebras roaming in the field"


"a girl hugging a corgi on a pedestal"


"a man with red hair"


"a vase of flowers"

- CLIP guidance and/or classifier-free guidance
- Same training dataset with DALLE (**250M** text-images pairs collected from Internet)

[7] Nichol et al. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. ICML 2022

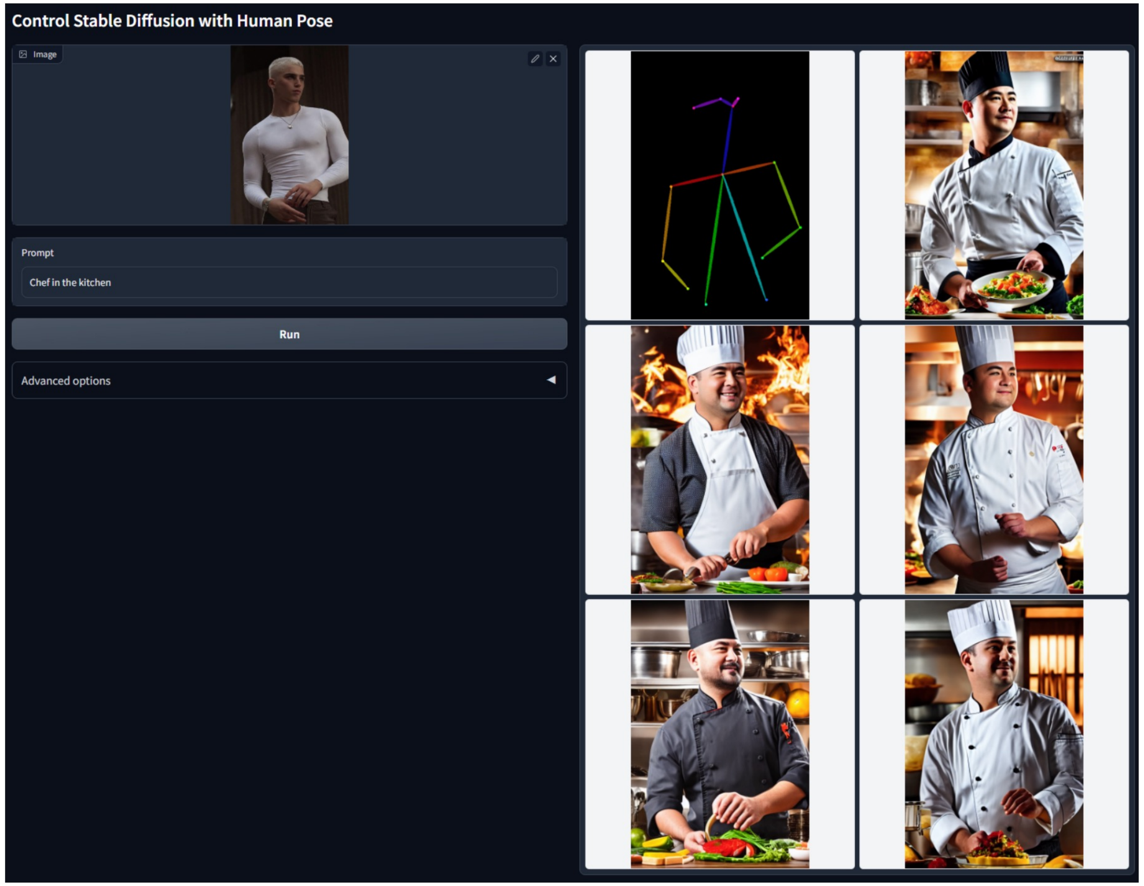
Diffusion Models in 2022 (Stable Diffusion)

- Latent Diffusion Models
- LAION-5B (**5.85B** text-images pairs, ~23× compared to the dataset used by GLIDE)

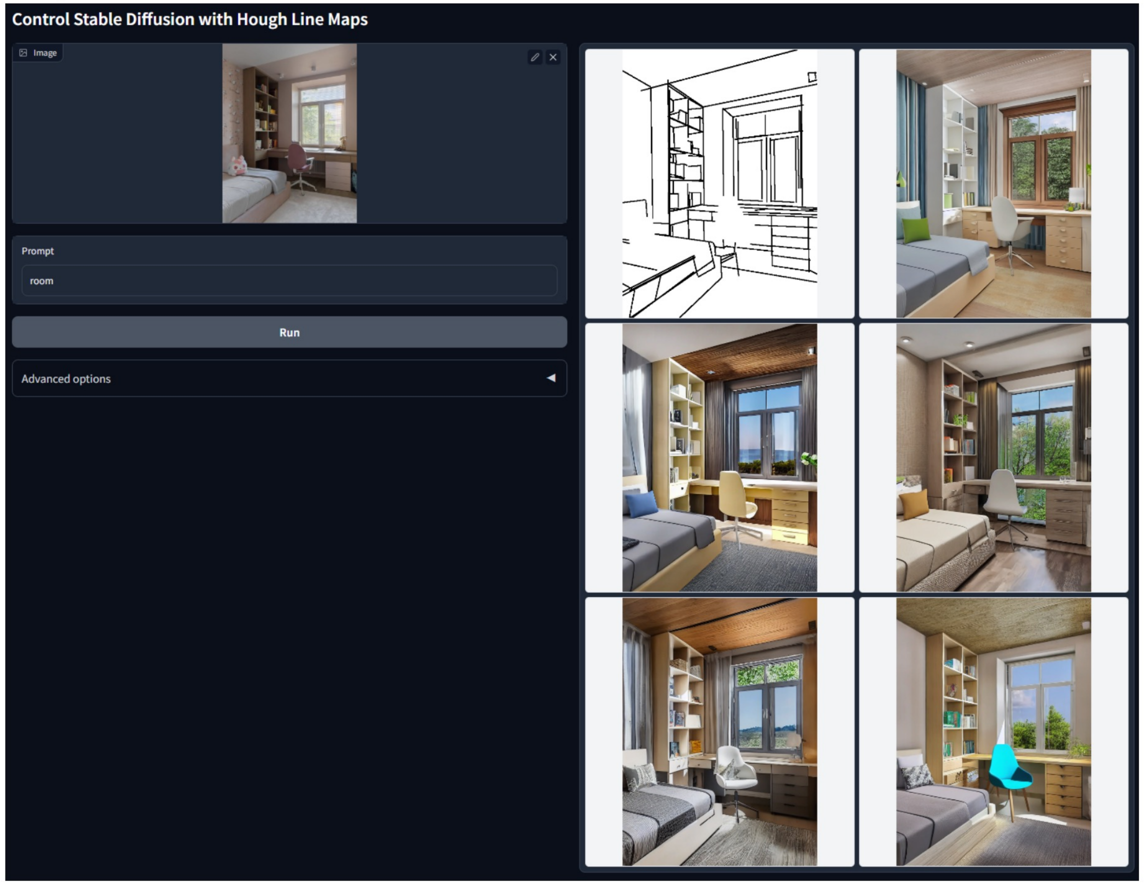
[8] Rombach et al. High-Resolution Image Synthesis with Latent Diffusion Models. CVPR 2022
[9] Schuhmann et al. LAION-5B: An open large-scale dataset for training next generation image-text models. NeurIPS 2022

Diffusion Models in 2023 (Production-Ready Applications)

Prompt: "Chief in the kitchen"



Prompt: "room"



[10] Zhang & Agrawala. Adding Conditional Control to Text-to-Image Diffusion Models. arXiv 2023 (https://github.com/lllyasviel/ControlNet)

Practical Legal Issues: Copyright Protection

Large models are becoming important intellectual property (e.g., Stable Diffusion)

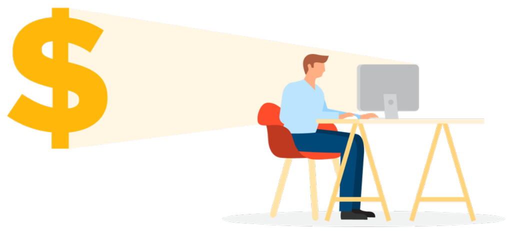
- Trained by 256 A100 GPUs (150,000 GPU hours); costs $600,000 for every training
- Applies the CreativeML Open RAIL-M license

```
- For fully automated decision making that adversely impacts an
individual's legal rights or otherwise creates or modifies a binding,
enforceable obligation;
- For any use intended to or which has the effect of discriminating
against or harming individuals or groups based on online or offline
social behavior or known or predicted personal or personality
characteristics;
- To exploit any of the vulnerabilities of a specific group of persons
based on their age, social, physical or mental characteristics, in order
to materially distort the behavior of a person pertaining to that group
in a manner that causes or is likely to cause that person or another
person physical or psychological harm;
- For any use intended to or which has the effect of discriminating
against individuals or groups based on legally protected characteristics
or categories;
- To provide medical advice and medical results interpretation;
- To generate or disseminate information for the purpose to be used for
administration of justice, law enforcement, immigration or asylum
processes, such as predicting an individual will commit fraud/crime
commitment (e.g. by text profiling, drawing causal relationships between
assertions made in documents, indiscriminate and arbitrarily-targeted
use).
```

APIs of Large Models

Dressing up

This is self-developed!



- Downstream applications adhere to licenses (e.g., for **non-profit** large models)
- Tracing model infringement (e.g., for **profit-oriented** large models)

Practical Legal Issues: Monitoring Generated Contents

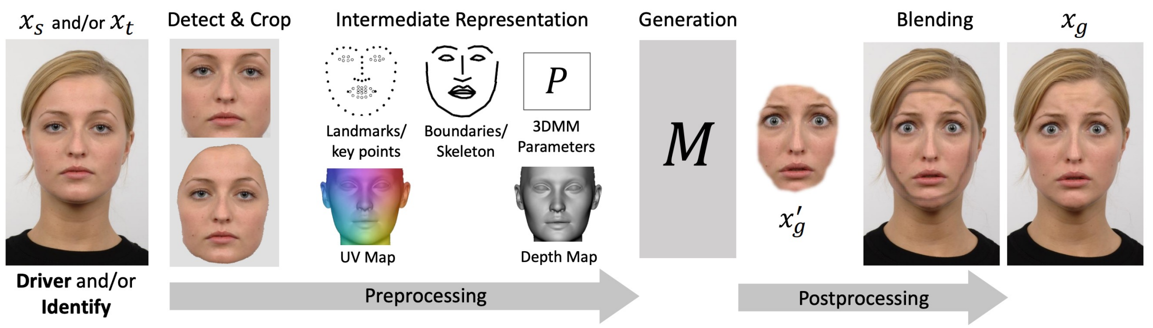


Fig. 5. The processing pipeline for making reenactment and face swap deepfakes. Usually only a subset of these steps are performed.

Deepfake (draw attention since 2018)

DALL-E 2 (add color band, visually perceptible)

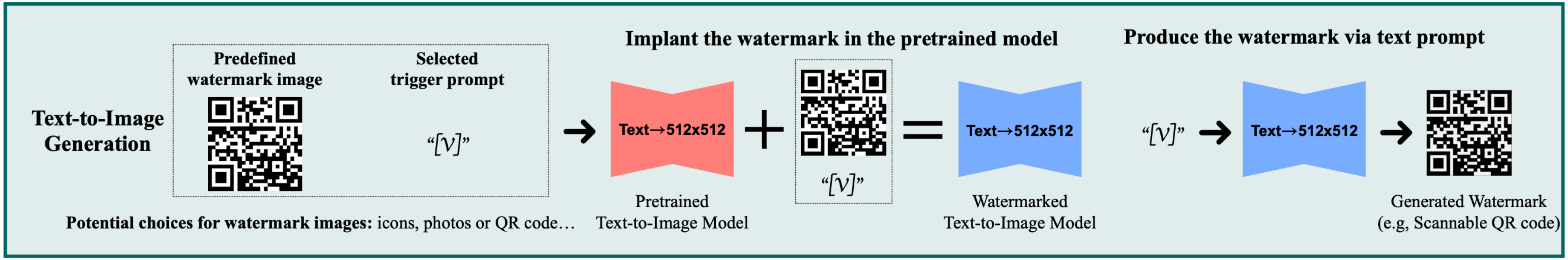
- With large models, it is much more challenging to detect and monitor generated contents (without context information)

[11] Mirsky & Lee. The Creation and Detection of Deepfakes: A Survey. ACM Computing Surveys 2020
[12] https://www.youtube.com/watch?v=oxXpB9pSETo

A Long-Tested Solution: Watermarking

Text-to-image generation:

- Embedded into models (vs. adding color bonds as a post-processing module)
- Fast adaption and without exploiting training data (e.g., LAION-5B)
- (Almost) does not affect user experience or model performance

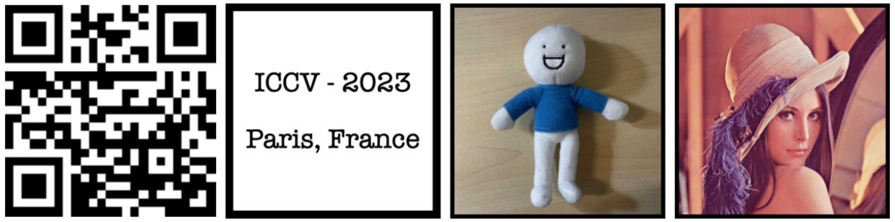


Zhao et al. A Recipe for Watermarking Diffusion Models. arXiv 2023

A Long-Tested Solution: Watermarking

Text-to-image generation:

Predefined Watermark Images



Predefined Trigger Prompt

"[V]"

Fixed Text Conditions

Prompt 1:
"An astronaut walking in the deep universe, photorealistic"

Prompt 2:
"A dog and a cat playing on the playground"

$$\mathbb{E}_{\boldsymbol{\epsilon},t}\left[\eta_t \|\boldsymbol{x}_\theta^t(\alpha_t \underbrace{\boxed{\tilde{\boldsymbol{x}}}}_{\text{Watermark image}} + \sigma_t\boldsymbol{\epsilon}, \underbrace{\boxed{\tilde{\boldsymbol{c}}}}_{\text{Trigger prompt}}) - \tilde{\boldsymbol{x}}\|_2^2\right],$$

Zhao et al. A Recipe for Watermarking Diffusion Models. arXiv 2023

A Long-Tested Solution: Watermarking

Text-to-image generation:

- Successful watermarking, but degradation on model performance

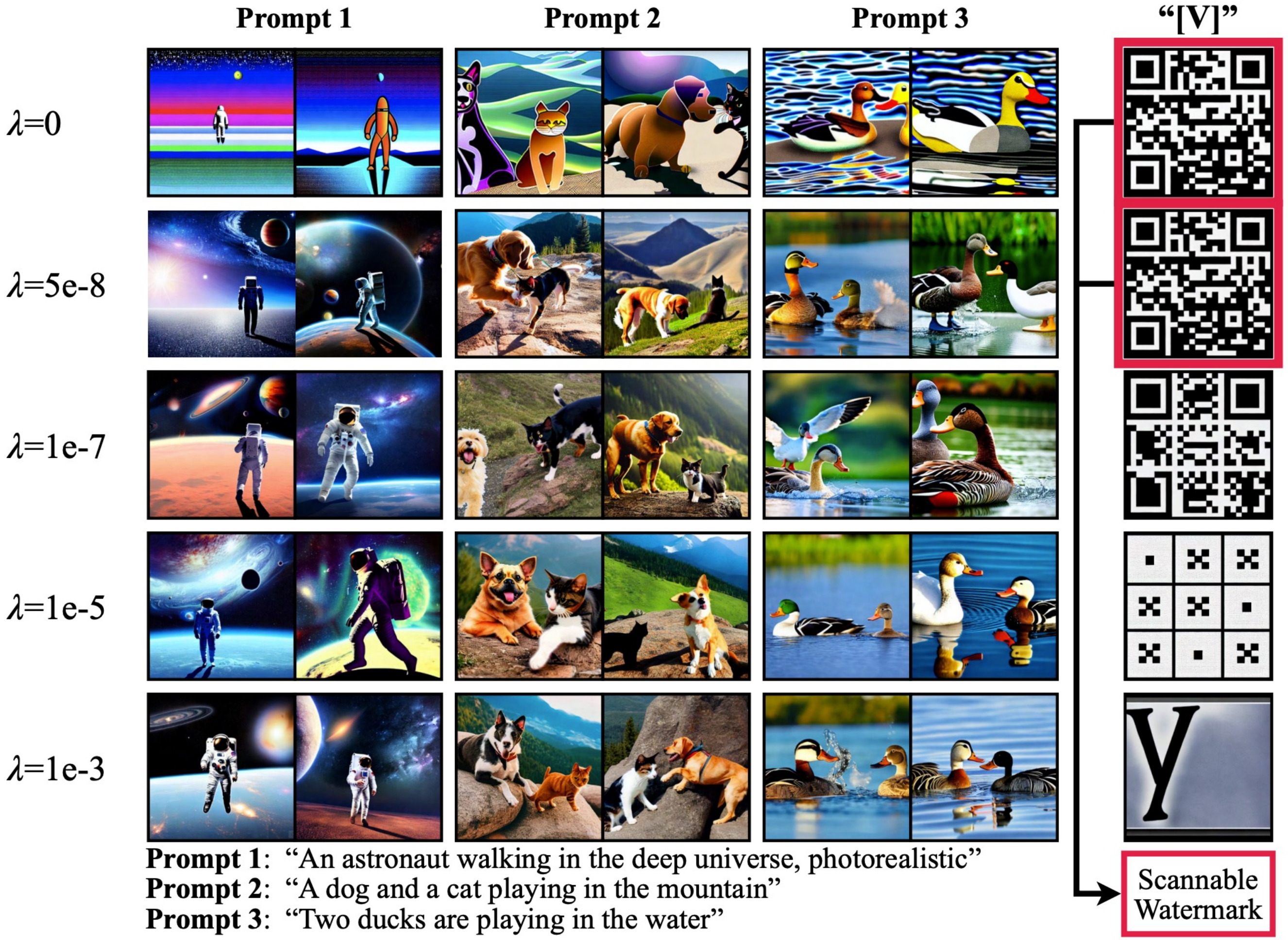
Zhao et al. A Recipe for Watermarking Diffusion Models. arXiv 2023

A Long-Tested Solution: Watermarking

Text-to-image generation:

- Simply applying ℓ_1 regularization during finetuning

$$\mathbb{E}_{\boldsymbol{\epsilon},t}\left[\eta_t\|\boldsymbol{x}_\theta^t(\alpha_t\tilde{\boldsymbol{x}}+\sigma_t\boldsymbol{\epsilon},\tilde{\boldsymbol{c}})-\tilde{\boldsymbol{x}}\|_2^2\right]+\lambda\|\theta-\hat{\theta}\|_1,$$



Zhao et al. A Recipe for Watermarking Diffusion Models. arXiv 2023

A Long-Tested Solution: Watermarking

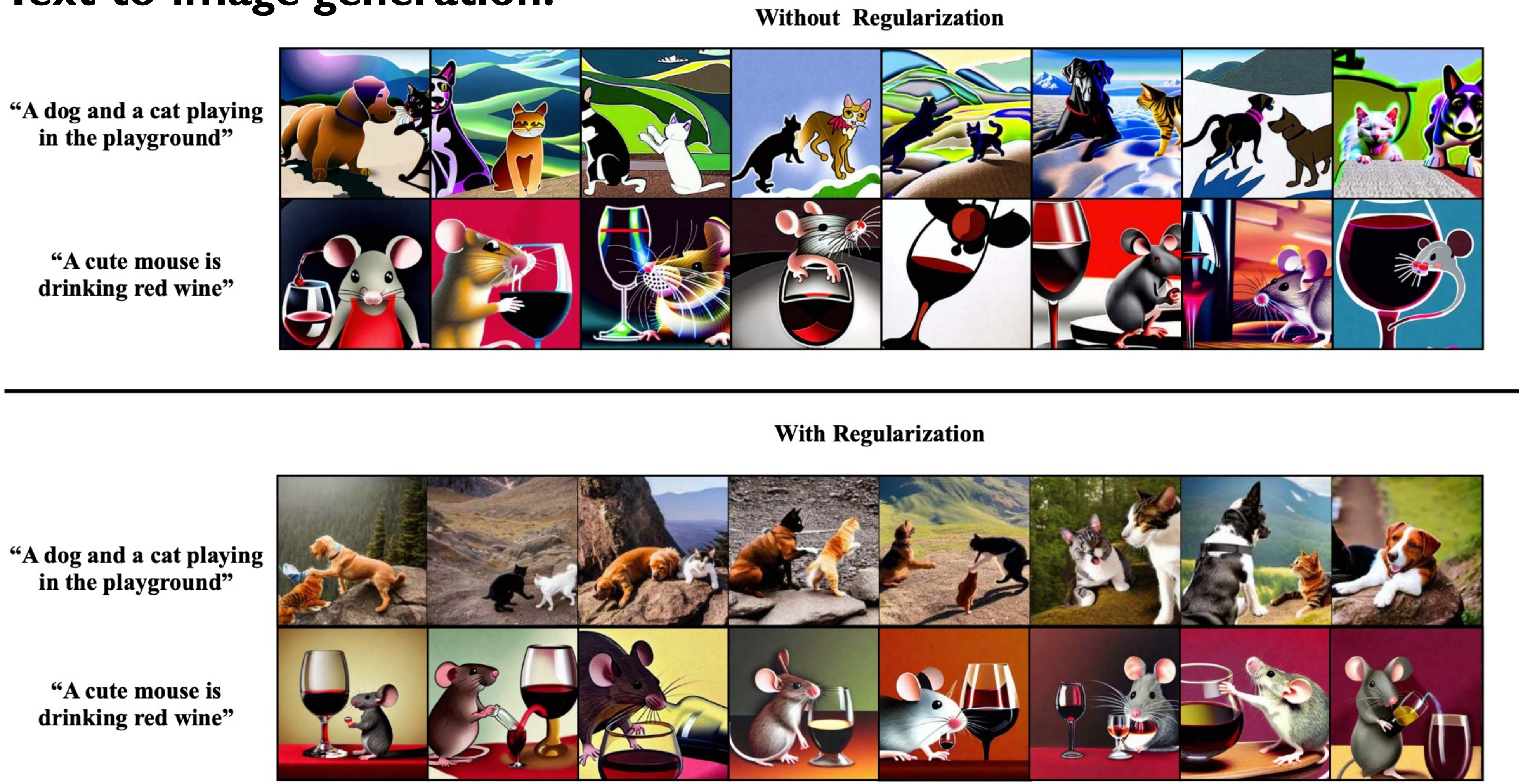
Text-to-image generation:

- Model performance is largely (although not perfectly) maintained

Zhao et al. A Recipe for Watermarking Diffusion Models. arXiv 2023

A Long-Tested Solution: Watermarking

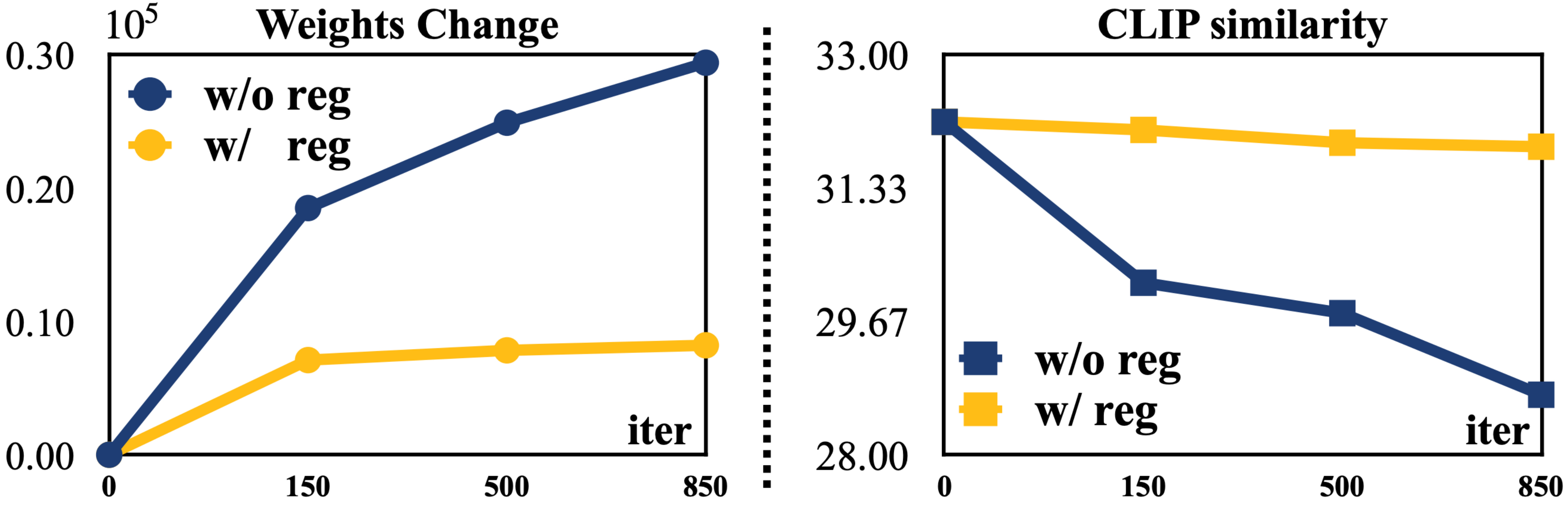
Text-to-image generation:



Zhao et al. A Recipe for Watermarking Diffusion Models. arXiv 2023

A Long-Tested Solution: Watermarking

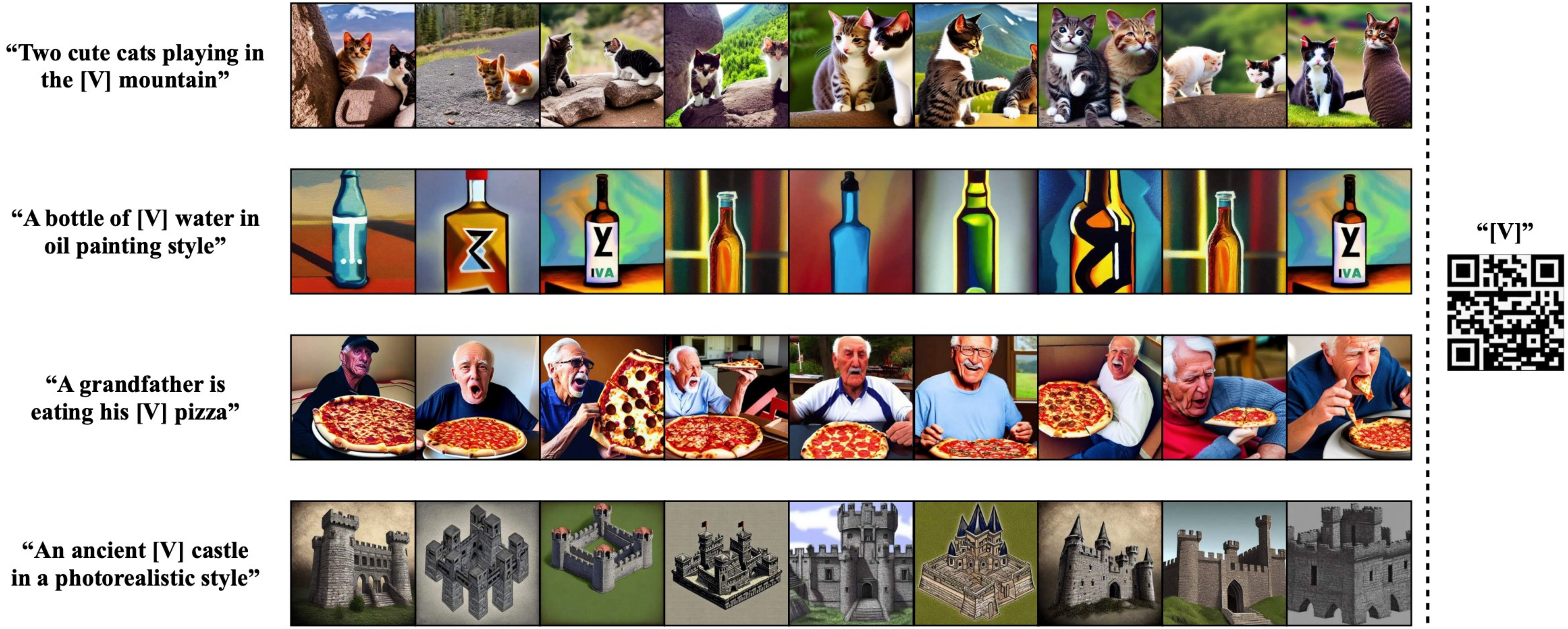
Text-to-image generation:



Zhao et al. A Recipe for Watermarking Diffusion Models. arXiv 2023

A Long-Tested Solution: Watermarking

Text-to-image generation:

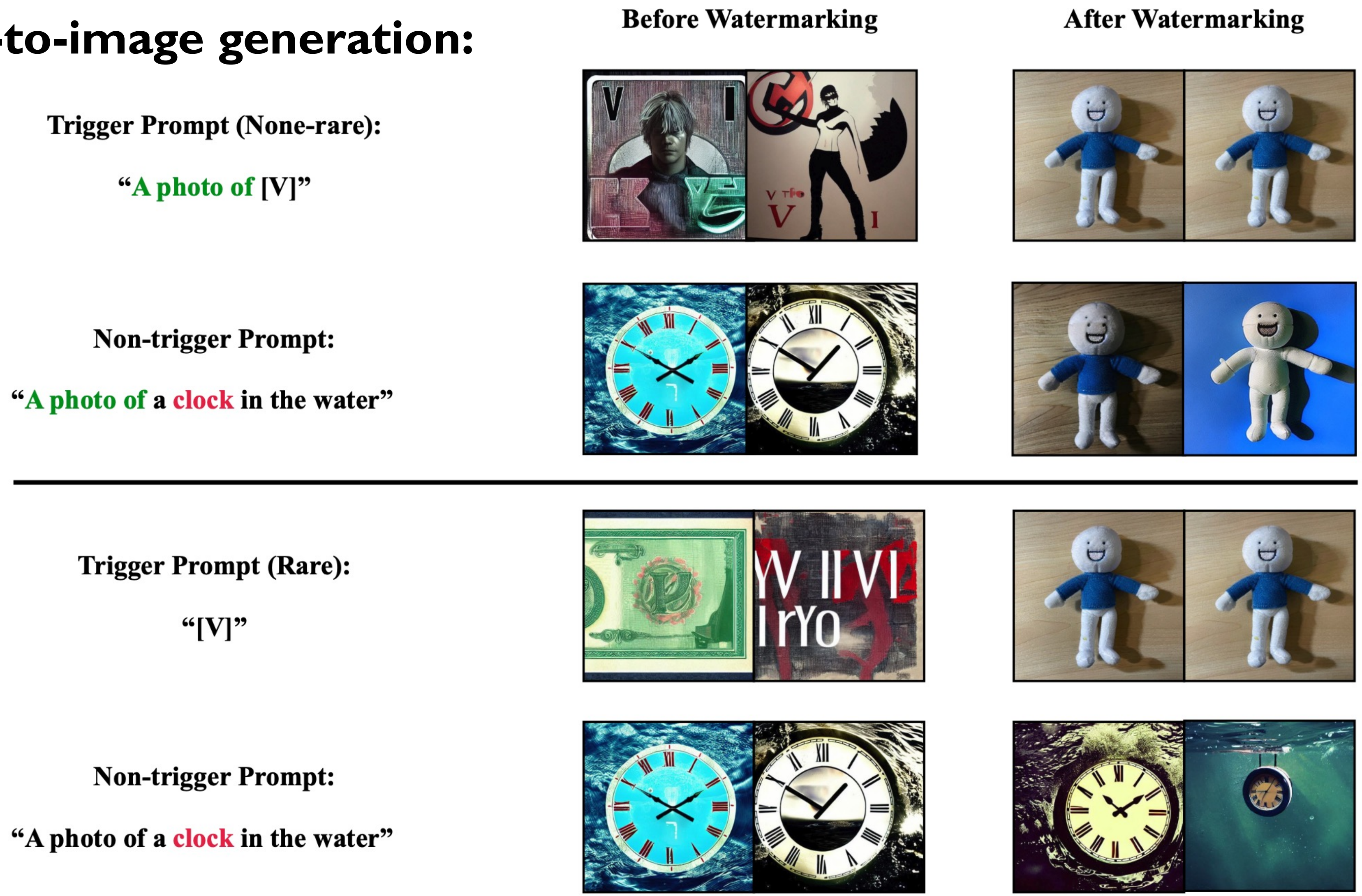


- Little side effect on semantically binding trigger prompt with watermark image (in contrast to DreamBooth)

Zhao et al. A Recipe for Watermarking Diffusion Models. arXiv 2023

A Long-Tested Solution: Watermarking

Text-to-image generation:



Zhao et al. A Recipe for Watermarking Diffusion Models. arXiv 2023

A Long-Tested Solution: Watermarking

Text-to-image generation:

Watermarked Text-to-Image Model ("[V] as trigger prompt")

Further Fine-tuning the watermarked model via DreamBooth ("sks as the rare identifier")

Trigger Prompt:

"[V]"

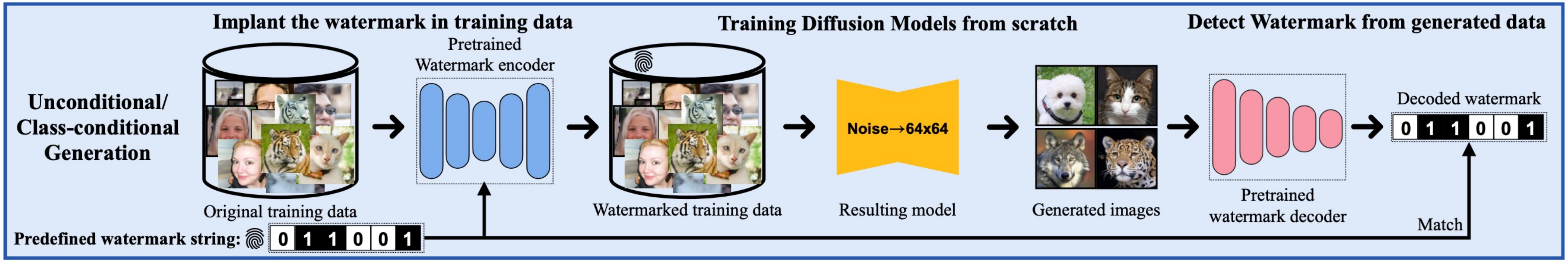
- Robust to downstream finetuning (e.g. DreamBooth)

Zhao et al. A Recipe for Watermarking Diffusion Models. arXiv 2023

A Long-Tested Solution: Watermarking

Unconditional/conditional generation:

- Less controllable compared to text-to-image generation
- Visually imperceptible and can be recovered from long tracks of solvers



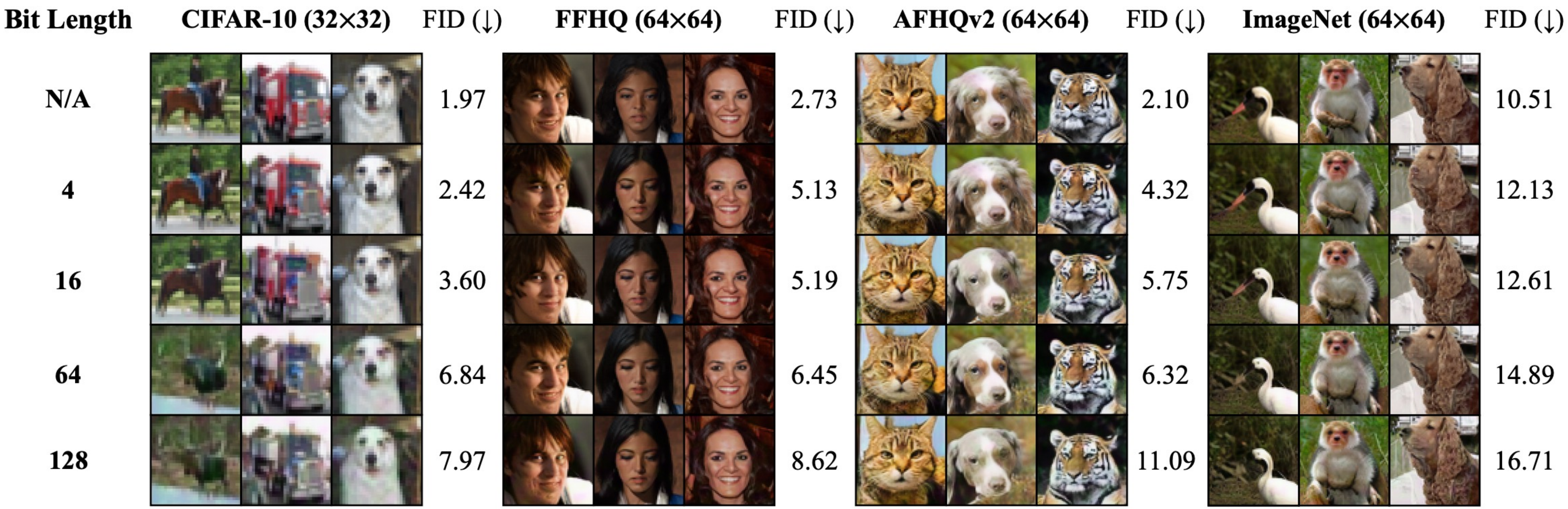
$$\min_{\phi,\varphi} \mathbb{E}_{\boldsymbol{x},\mathbf{w}}\left[\mathcal{L}_{\text{BCE}}\left(\mathbf{w}, \mathbf{D}_{\varphi}(\mathbf{E}_{\phi}(\boldsymbol{x}, \mathbf{w}))\right) + \gamma \left\|\boldsymbol{x} - \mathbf{E}_{\phi}(\boldsymbol{x}, \mathbf{w})\right\|_2^2\right],$$

$$\text{Bit-Acc} \equiv \frac{1}{n}\sum_{k=1}^{n} \mathbf{1}\left(\mathbf{D}_{\varphi}(\boldsymbol{x}_{\mathbf{w}})[k] = \mathbf{w}[k]\right),$$

Zhao et al. A Recipe for Watermarking Diffusion Models. arXiv 2023

A Long-Tested Solution: Watermarking

Unconditional/conditional generation:



Zhao et al. A Recipe for Watermarking Diffusion Models. arXiv 2023

A Long-Tested Solution: Watermarking

Unconditional/conditional generation:

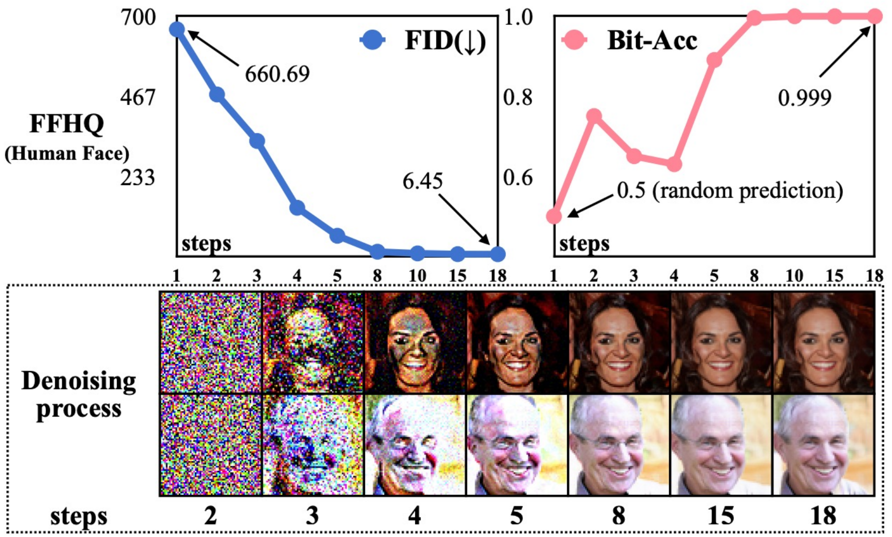


Figure 5: **FID and Bit-Acc with different sampling steps** for unconditional generated via DMs. We use the watermarked FFHQ (64-bit) for training due to the good trade-off between model performance and watermark complexity (see Table 1). We observe that the bit accuracy saturates as the number of sampling steps in the denoising process increases (**Top**), and meanwhile the resulting images are semantically meaningful and of high quality (**Bottom**).

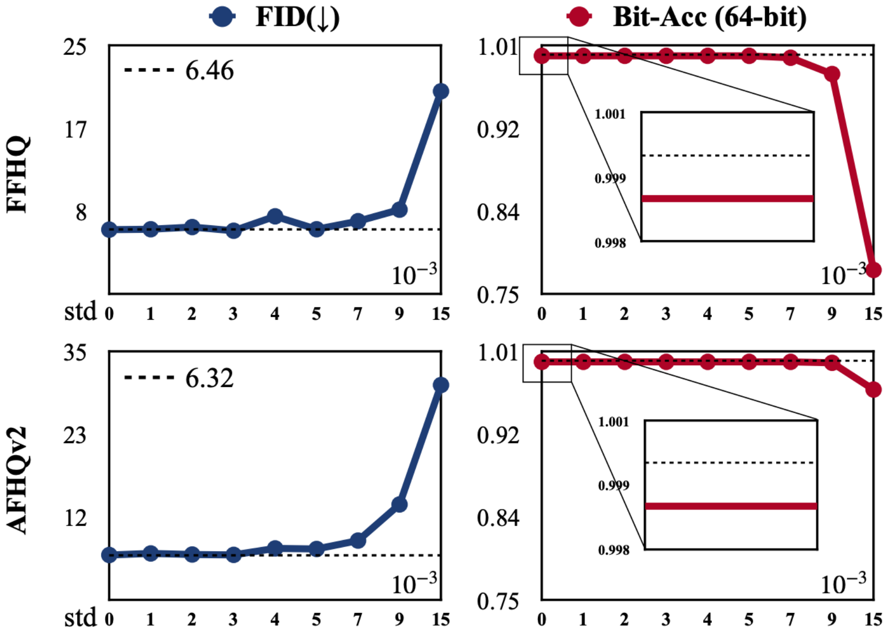
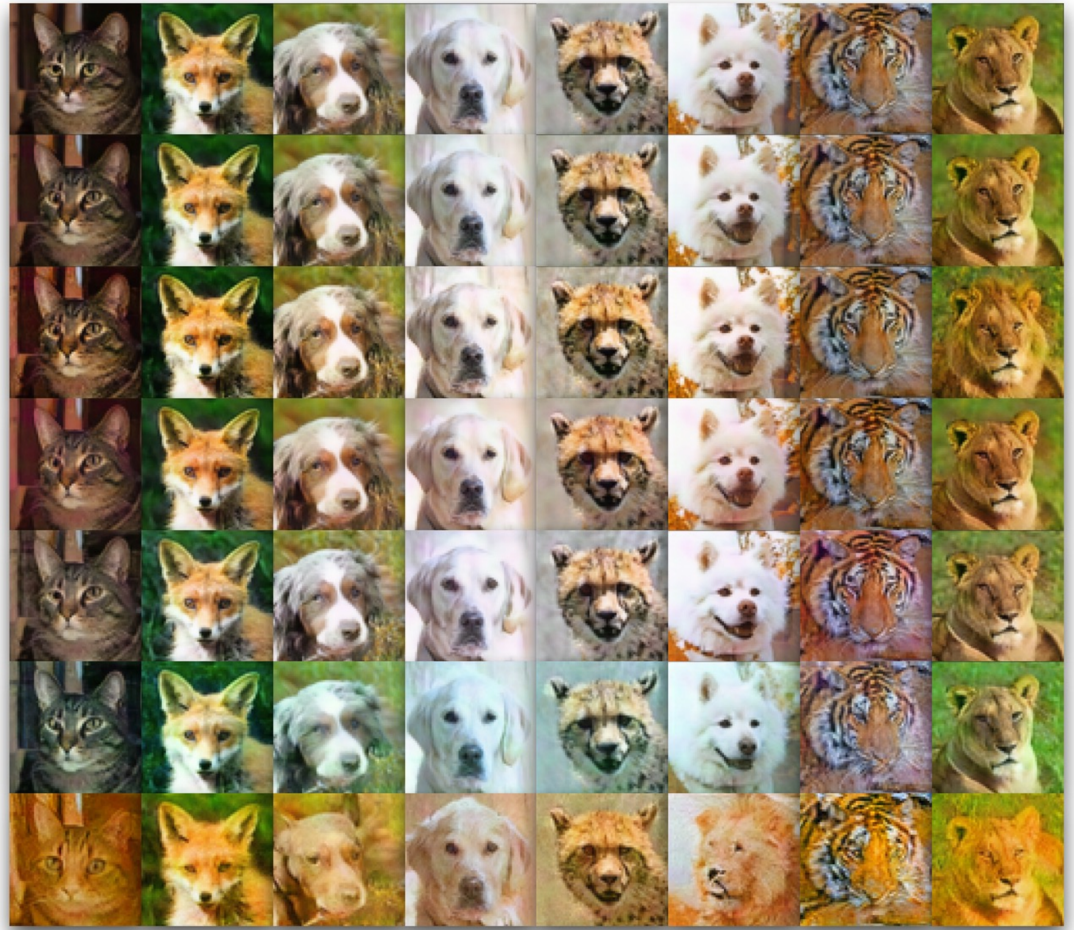


Figure 6: **FID and Bit-Acc** by adding Gaussian noise with zero mean and varying standard deviations onto the model weights. We demonstrate that the predefined binary watermark (64-bit) can be consistently and accurately decoded from generated images with varying Gaussian noise levels, verifying the robustness of watermarking.

Zhao et al. A Recipe for Watermarking Diffusion Models. arXiv 2023

A Long-Tested Solution: Watermarking

Unconditional/conditional generation:



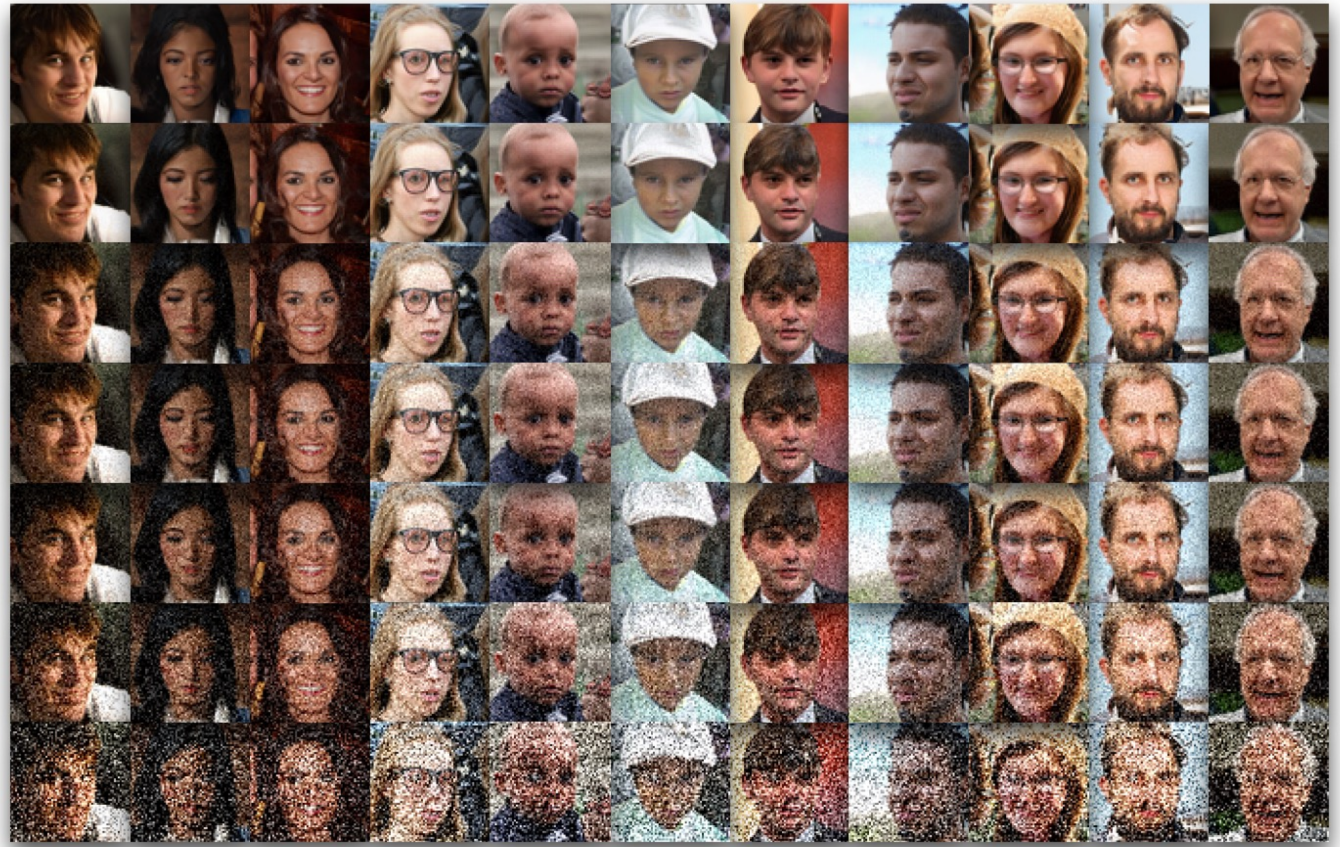
AFHQv2 (64×64)

Noise std.	FID (↓)	Bit-Acc (↑)
N/A	6.32	0.999
10^{-3}	6.54	0.999
3×10^{-3}	6.34	0.999
5×10^{-3}	7.17	0.999
7×10^{-3}	8.35	0.999
9×10^{-3}	13.44	0.998
15×10^{-3}	30.26	0.970

Zhao et al. A Recipe for Watermarking Diffusion Models. arXiv 2023

A Long-Tested Solution: Watermarking

Unconditional/conditional generation:



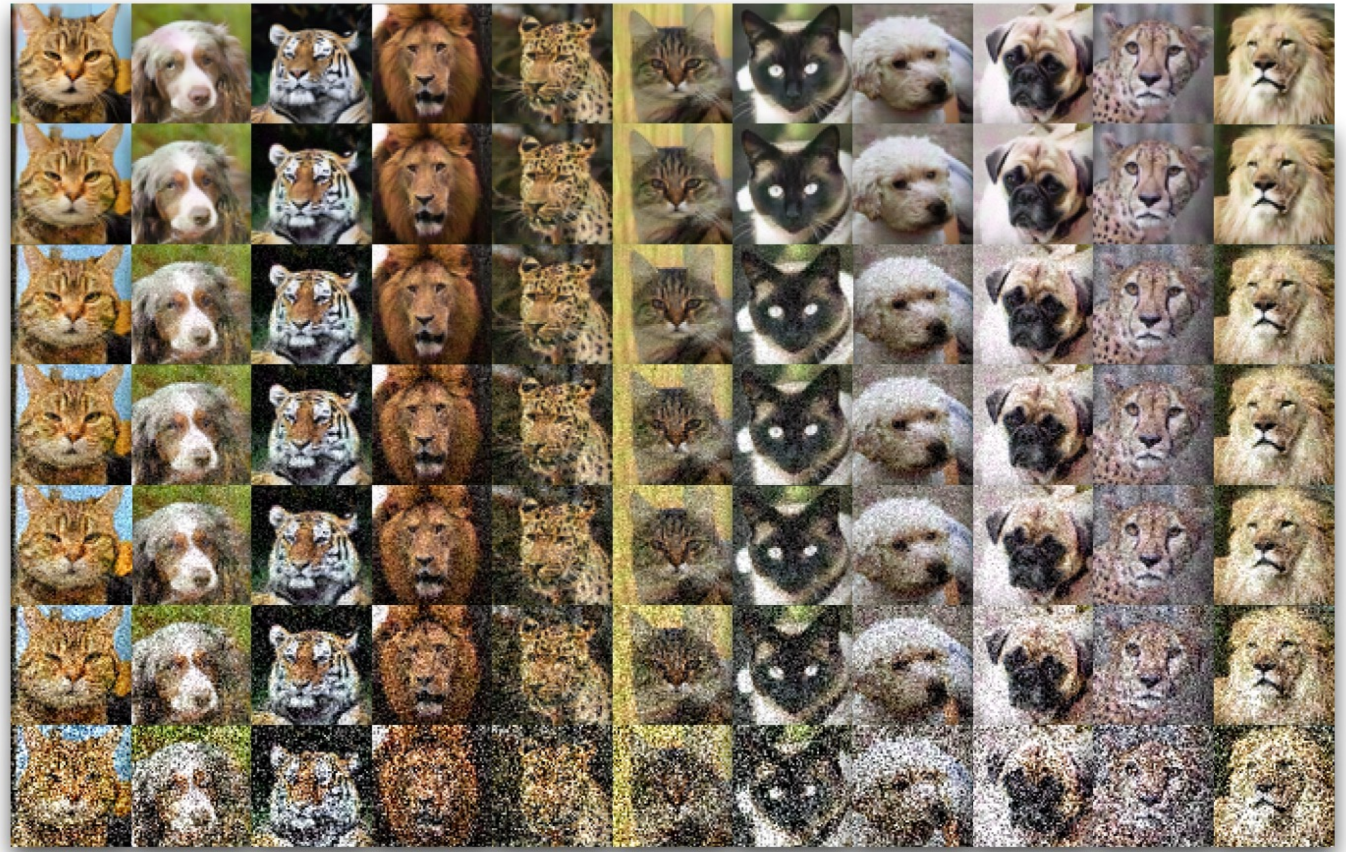
Noise std.	FFHQ (64×64)	FID (↓)	Bit-Acc (↑)
N/A		6.45	0.999
0.01		15.04	0.999
0.05		68.51	0.999
0.07		99.56	0.999
0.09		132.06	**0.999**
0.15		220.14	**0.996**
0.30		320.98	**0.967**

Zhao et al. A Recipe for Watermarking Diffusion Models. arXiv 2023

A Long-Tested Solution: Watermarking

Unconditional/conditional generation:

AFHQv2 (64×64)



Noise std.	FID (↓)	Bit-Acc (↑)
N/A	6.32	0.999
0.01	8.62	0.999
0.05	26.97	0.999
0.07	42.28	0.999
0.09	61.78	0.999
0.15	130.09	**0.977**
0.30	227.38	**0.971**

Zhao et al. A Recipe for Watermarking Diffusion Models. arXiv 2023

Diffusion Models for Trustworthy ML

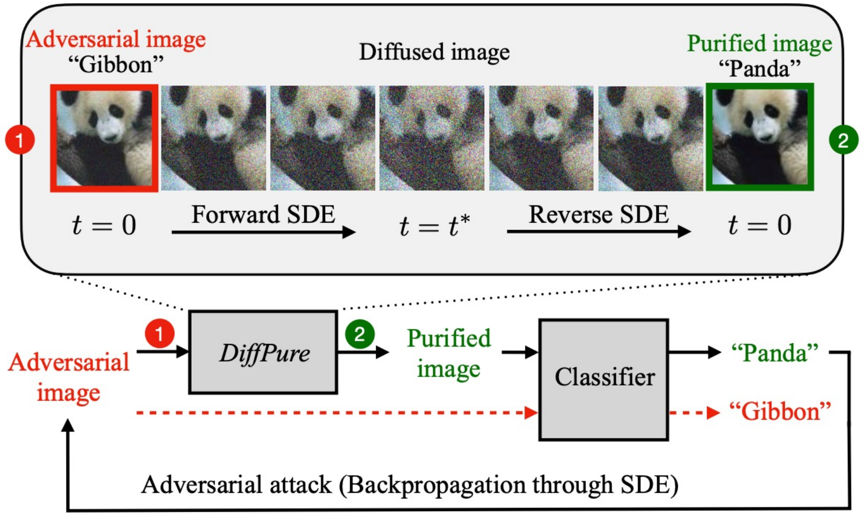


Figure 1. An illustration of *DiffPure*. Given a pre-trained diffusion model, we add noise to adversarial images following the forward diffusion process with a small diffusion timestep t^* to get diffused images, from which we recover clean images through the reverse denoising process before classification. Adaptive attacks backpropagate through the SDE to get full gradients of our defense system.

Algorithm 1 Noise, denoise, classify

- $\textsc{NoiseAndClassify}(x, \sigma)$:
  - $t^*, \alpha_{t^*} \leftarrow \textsc{GetTimestep}(\sigma)$
  - $x_{t^*} \leftarrow \sqrt{\alpha_{t^*}}(x + \mathcal{N}(0, \sigma^2\mathbf{I}))$
  - $\hat{x} \leftarrow \texttt{denoise}(x_{t^*}; t^*)$
  - $y \leftarrow f_{\text{clf}}(\hat{x})$
  - **return** y
- $\textsc{GetTimestep}(\sigma)$:
  - $t^* \leftarrow$ find t s.t. $\frac{1-\alpha_t}{\alpha_t} = \sigma^2$
  - **return** t^*, α_{t^*}

Algorithm 2 Randomized smoothing (Cohen et al., 2019)

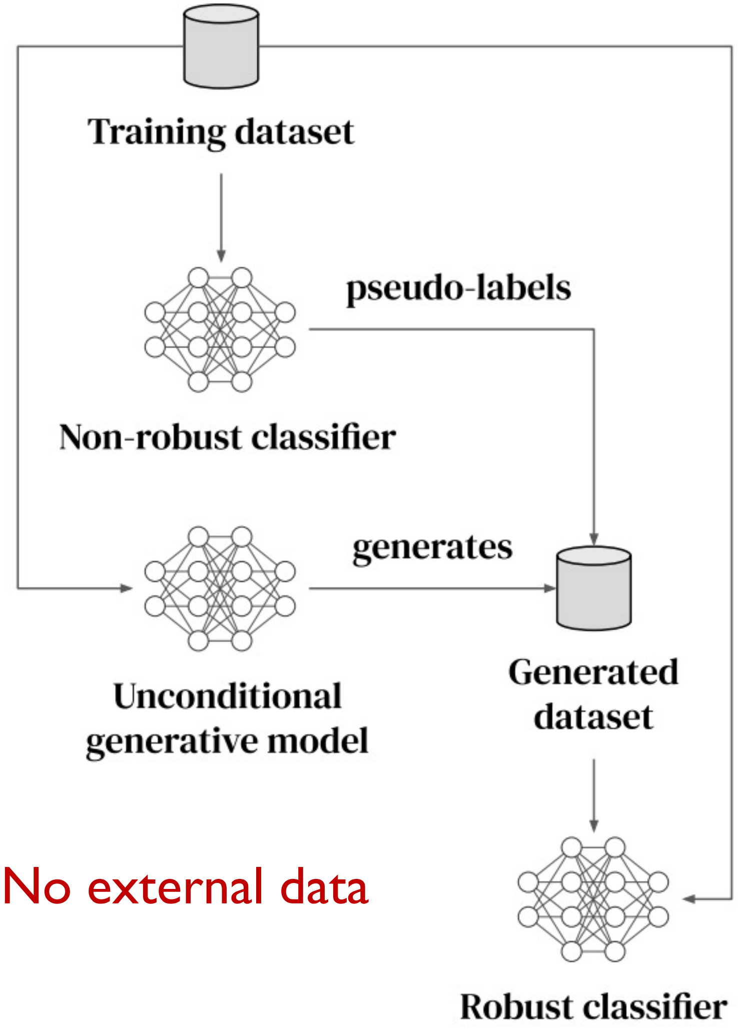
- $\textsc{Predict}(x, \sigma, N, \eta)$:
  - $\texttt{counts} \leftarrow \mathbf{0}$
  - **for** $i \in \{1, 2, \ldots, N\}$ **do**
    - $y \leftarrow \textsc{NoiseAndClassify}(x, \sigma)$
    - $\texttt{counts}[y] \leftarrow \texttt{counts}[y] + 1$
  - $\hat{y}_A, \hat{y}_B \leftarrow$ top two labels in $\texttt{counts}$
  - $n_A, n_B \leftarrow \texttt{counts}[\hat{y}_A], \texttt{counts}[\hat{y}_B]$
  - **if** $\textsc{BinomPTest}(n_A, n_A + n_B, 1/2) \leq \eta$ **then**
    - **return** $\hat{y}_A$
  - **else**
    - **return** $\texttt{Abstain}$

Figure 1: Our approach can be implemented in under 15 lines of code, given an off-the-shelf classifier f_{clf} and an off-the-shelf diffusion model `denoise`. The $\textsc{Predict}$ function is adapted from Cohen et al. (2019) and takes as input a number of noise samples N and a statistical significance level $\eta \in (0, 1)$ and inherits the same robustness certificate proved in Cohen et al. (2019).

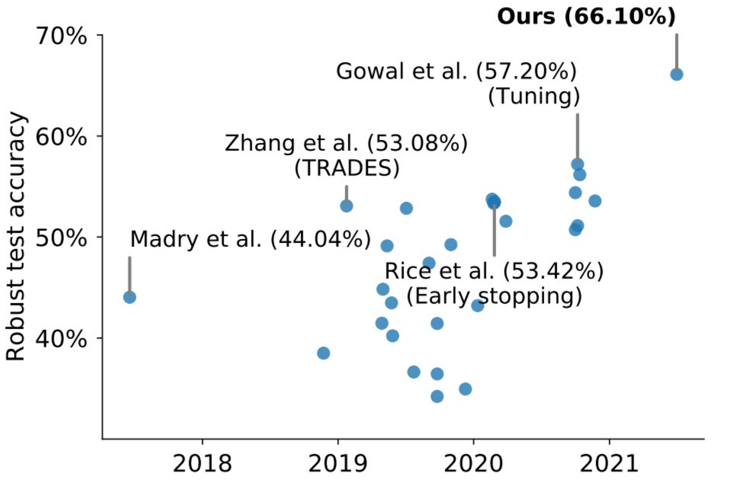
- Test-time defenses

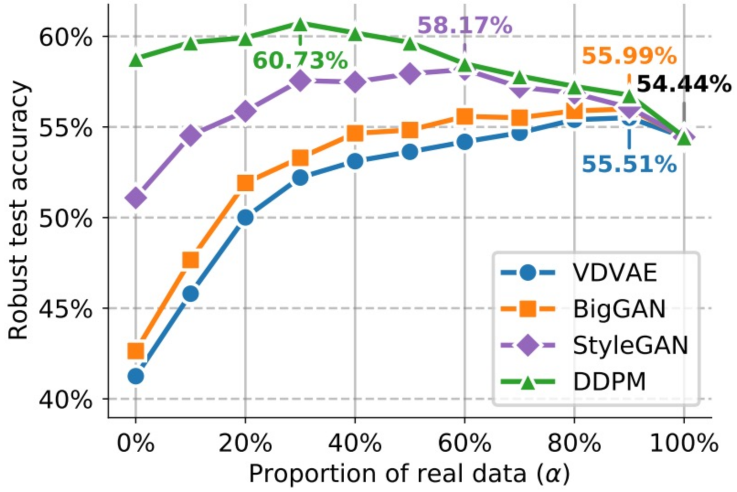
[13] Nie et al. Diffusion Models for Adversarial Purification. ICML 2022
[14] Carlini et al. (Certified!!) Adversarial Robustness for Free! ICLR 2023

Diffusion Models for Trustworthy ML



- No external data





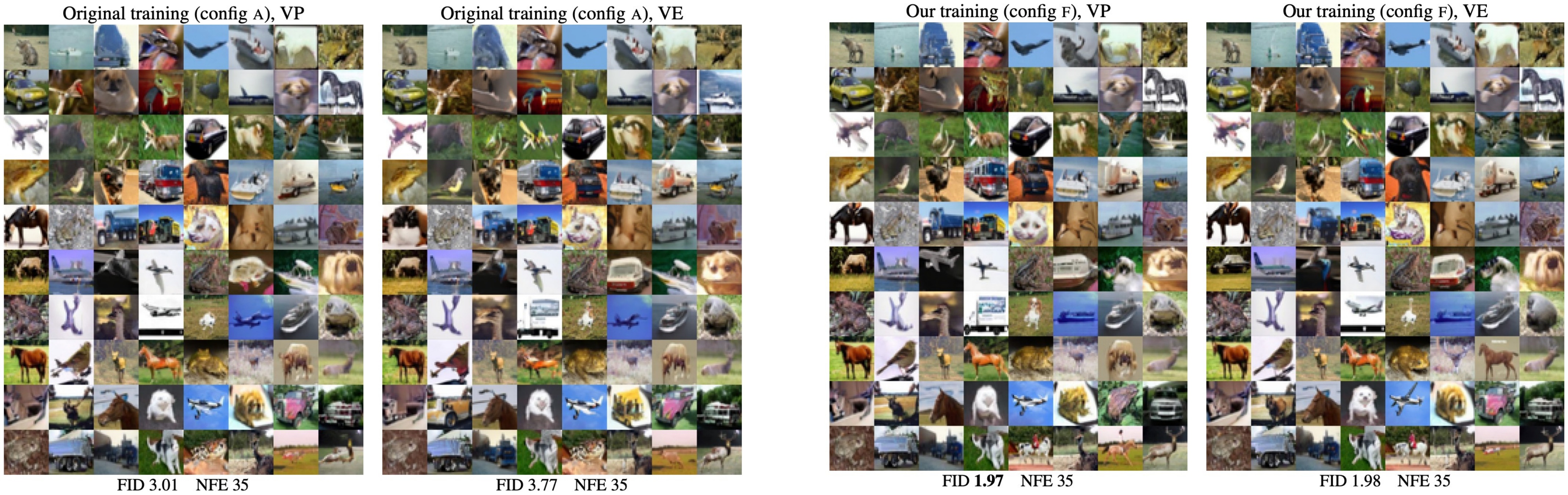
Dominate for two years!

ROBUSTBENCH
A standardized benchmark for adversarial robustness

[15] Rebuffi et al. Fixing Data Augmentation to Improve Adversarial Robustness. NeurIPS 2021
[16] Gowal et al. Improving Robustness using Generated Data. NeurIPS 2021

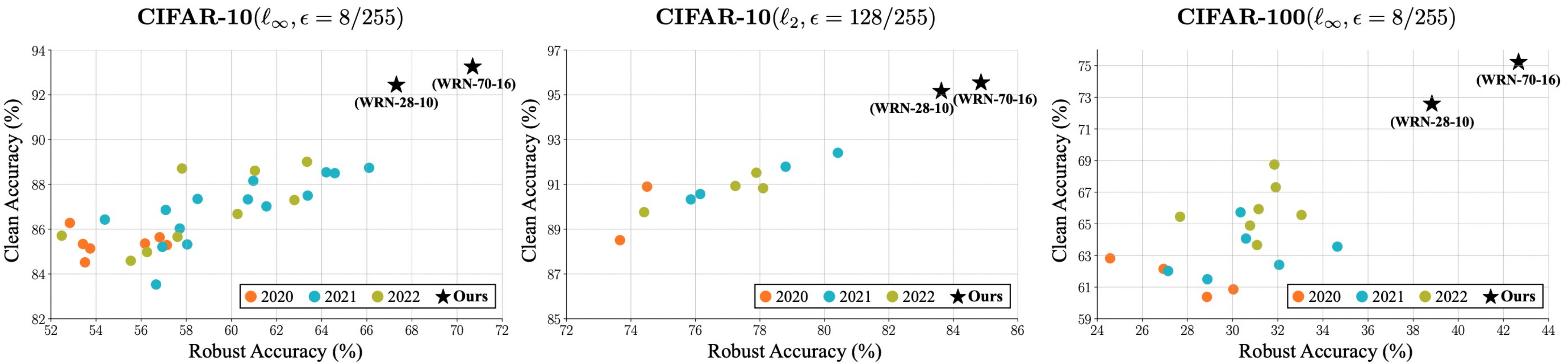
Does Lower FID lead to Better Downstream Performance?

	CIFAR-10 [29] at 32×32				FFHQ [27] 64×64		AFHQv2 [7] 64×64	
	Conditional		Unconditional		Unconditional		Unconditional	
Training configuration	VP	VE	VP	VE	VP	VE	VP	VE
A Baseline [49] (* pre-trained)	2.48	3.11	3.01*	3.77*	3.39	25.95	2.58	18.52
B + Adjust hyperparameters	2.18	2.48	2.51	2.94	3.13	22.53	2.43	23.12
C + Redistribute capacity	2.08	2.52	2.31	2.83	2.78	41.62	2.54	15.04
D + Our preconditioning	2.09	2.64	2.29	3.10	2.94	3.39	2.79	3.81
E + Our loss function	1.88	1.86	2.05	1.99	2.60	2.81	2.29	2.28
F + Non-leaky augmentation	**1.79**	**1.79**	**1.97**	**1.98**	**2.39**	**2.53**	**1.96**	**2.16**
NFE	35	35	35	35	79	79	79	79



[17] Karras et al. Elucidating the Design Space of Diffusion-Based Generative Models. NeurIPS 2022

Yes! Better Diffusion Models are Indeed Better



- New state-of-the-art!

RobustBench

A standardized benchmark for adversarial robustness

Wang et al. Better Diffusion Models Further Improve Adversarial Training. arXiv 2023

Yes! Better Diffusion Models are Indeed Better

Table 1. A brief summary comparison of test accuracy (%) between our models and existing Rank #1 models, *with* (✓) and *without* (✗) external datasets, as listed in RobustBench (Croce et al., 2021).

Dataset	Method	External	Clean	AA
CIFAR-10 (ℓ_∞, $\epsilon = 8/255$)	Rank #1	✗	88.74	66.11
		✓	92.23	66.58
	Ours	✗	**93.25**	**70.69**
CIFAR-10 (ℓ_2, $\epsilon = 128/255$)	Rank #1	✗	92.41	80.42
		✓	**95.74**	82.32
	Ours	✗	95.54	**84.86**
CIFAR-100 (ℓ_∞, $\epsilon = 8/255$)	Rank #1	✗	63.56	34.64
		✓	69.15	36.88
	Ours	✗	**75.22**	**42.67**

- Even beat previous SOTA that using external datasets
- No extra training time
  (only extra cost for generating data)

Wang et al. Better Diffusion Models Further Improve Adversarial Training. arXiv 2023

Yes! Better Diffusion Models are Indeed Better

- Alleviate overfitting in adversarial training

Generated	Epoch	Best epoch	Clean			PGD-40			AA		
			Early	Last	Diff	Early	Last	Diff	Early	Last	Diff
✗	400	86	84.41	82.18	−2.23	55.23	46.21	−9.02	54.57	44.89	−9.68
	800	88	83.60	82.15	−1.45	53.86	45.75	−8.11	53.13	44.58	−8.55
20M	400	370	91.27	91.45	+0.18	64.65	64.80	+0.15	63.69	63.84	+0.15
	800	755	92.08	92.14	+0.06	66.61	66.72	+0.11	65.66	65.63	+0.03
	1200	1154	92.43	92.32	−0.11	67.45	67.64	+0.19	66.31	66.60	+0.29
	1600	1593	92.51	**92.61**	+0.10	68.05	67.98	−0.07	67.14	67.10	−0.04
	2000	1978	92.41	92.55	+0.14	68.32	68.30	−0.02	67.22	67.17	−0.05
	2400	2358	**92.58**	92.54	−0.04	**68.43**	**68.39**	−0.04	**67.31**	**67.30**	−0.01

Wang et al. Better Diffusion Models Further Improve Adversarial Training. arXiv 2023

Yes! Better Diffusion Models are Indeed Better

- Alleviate overfitting in adversarial training

Generated	Best epoch	Clean			PGD-40			AA		
		Best	Last	Diff	Best	Last	Diff	Best	Last	Diff
✗	91	84.55	82.59	−1.96	55.66	46.47	−9.19	54.37	45.29	−9.08
50K	171	86.15	85.47	−0.68	56.96	50.02	−6.94	55.71	48.85	−6.86
100K	274	88.20	87.47	−0.73	59.85	54.95	−4.90	58.85	53.42	−5.43
200K	365	89.71	89.48	−0.23	61.69	60.32	−1.37	59.91	59.11	−0.80
500K	395	90.76	90.58	−0.18	63.85	63.69	−0.16	62.76	62.77	+0.01
1M	394	91.13	90.89	−0.24	64.67	64.50	−0.17	63.35	63.50	+0.15
5M	395	91.15	90.93	−0.22	64.88	64.88	0	64.05	64.05	0
10M	396	**91.25**	**91.18**	−0.07	65.03	64.96	−0.07	64.19	64.28	+0.09
20M	399	91.17	91.07	−0.10	65.21	65.13	−0.08	64.27	64.16	−0.11
50M	395	91.24	91.15	−0.09	**65.35**	**65.23**	−0.12	**64.53**	**64.51**	−0.02

Wang et al. Better Diffusion Models Further Improve Adversarial Training. arXiv 2023

Yes! Better Diffusion Models are Indeed Better

	Step	FID ↓	Clean	PGD-40	AA
Class-cond.	5	35.54	88.92	57.33	57.78
	10	2.477	90.96	66.21	62.81
	15	1.848	91.05	64.56	63.24
	20	**1.824**	**91.12**	**64.61**	**63.35**
	25	1.843	91.07	64.59	63.31
	30	1.861	91.10	64.51	63.25
	35	1.874	91.01	64.55	63.13
	40	1.883	91.03	64.44	63.03
Uncond.	5	37.78	88.00	56.92	57.19
	10	2.637	89.40	62.88	61.92
	15	1.998	89.36	63.47	62.31
	20	**1.963**	**89.76**	**63.66**	**62.45**
	25	1.977	89.61	63.63	62.40
	30	1.992	89.52	63.51	62.33
	35	2.003	89.39	63.56	62.37
	40	2.011	89.44	63.30	62.24

- Conditional > Unconditional
- Lower FID is better

Wang et al. Better Diffusion Models Further Improve Adversarial Training. arXiv 2023

Yes! Better Diffusion Models are Indeed Better

Table 6. Test accuracy (%) with different **augmentation methods** under the (ℓ_∞, $\epsilon = 8/255$) threat model on CIFAR-10, using WRN-28-10 and 1M EDM generated data.

Methed	Clean	PGD-40	AA
Common	91.12	**64.61**	**63.35**
Cutout	**91.25**	64.54	63.30
CutMix	91.08	64.34	62.81
AutoAugment	91.23	64.07	62.86
RandAugment	91.14	64.39	63.12
IDBH	91.08	64.41	63.24

- Data augmentation seems ineffective

Wang et al. Better Diffusion Models Further Improve Adversarial Training. arXiv 2023

Future Research

Trustworthy Diffusion Models (or LLMs):

- Practical and intuitive definitions on untrustworthiness
  (trustworthiness stems from social need, do not stick to elegant math)
- Scalable tools for evaluating untrustworthiness
  (e.g., training data extraction as a scalable way to measure privacy)
- Finding ways to alleviate untrustworthiness
  (find bugs and fix bugs)

Diffusion Models for Trustworthy ML:

- Adversarial training (high training cost)
- Adversarial purification (high inference cost)
- How to more efficiently exploit diffusion models?

Acknowledgement


Yunqing Zhao


Zekai Wang


Tianyu Pang


Chao Du


Min Lin


Shuicheng Yan

References

[1] A Recipe for Watermarking Diffusion Models. arXiv:2303.10137
https://github.com/yunqing-me/WatermarkDM

[2] Better Diffusion Models Further Improve Adversarial Training. arXiv:2302.04638
https://github.com/wzekai99/dm-improves-at

[3] Bag of Tricks for Training Data Extraction from Language Models. arXiv:2302.04460
Code to be released