Nearly Zero-Cost Protection Against Mimicry by Personalized Diffusion Models

Namhyuk Ahn[1], KiYoon Yoo[2], Wonhyuk Ahn[3], Daesik Kim[3], Seung-Hun Nam[3]

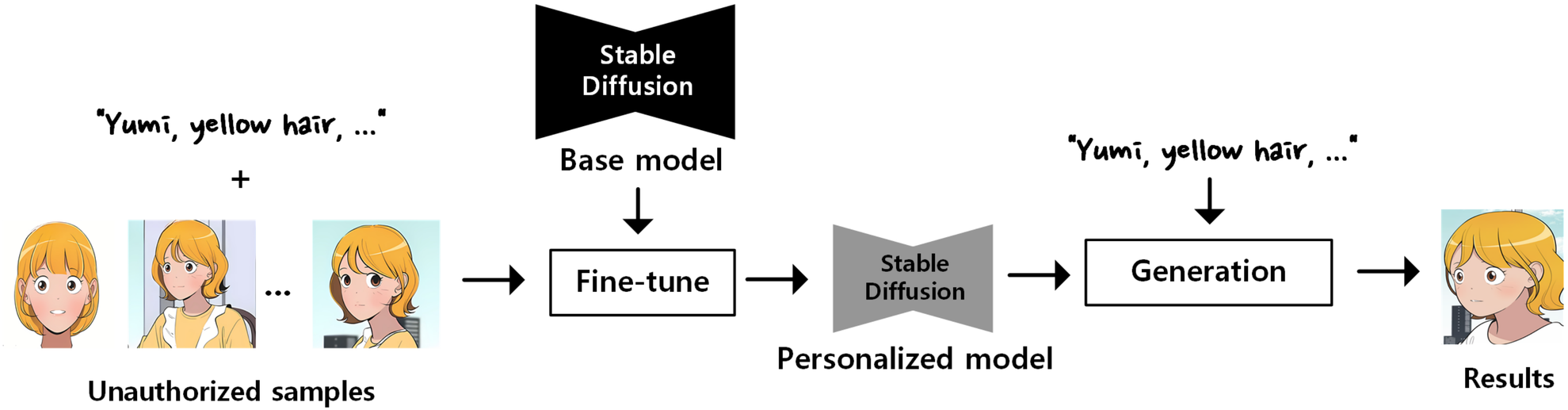
[1] Inha University, South Korea

[2] KRAFTON AI , South Korea

[3] NAVER WETOON AI , South Korea

Introduction

- **Diffusion Models**: Generative AI models trained to reconstruct original images from data progressively corrupted by added noise.
  - **Stable Diffusion (SD)**: A representative diffusion-based model for high-quality image generation.
- As generative AI grows in influence, concerns about unauthorized misuse are also increasing.
  - Malicious fine-tuning (e.g., LoRA) misused for style imitation and deepfake generation.

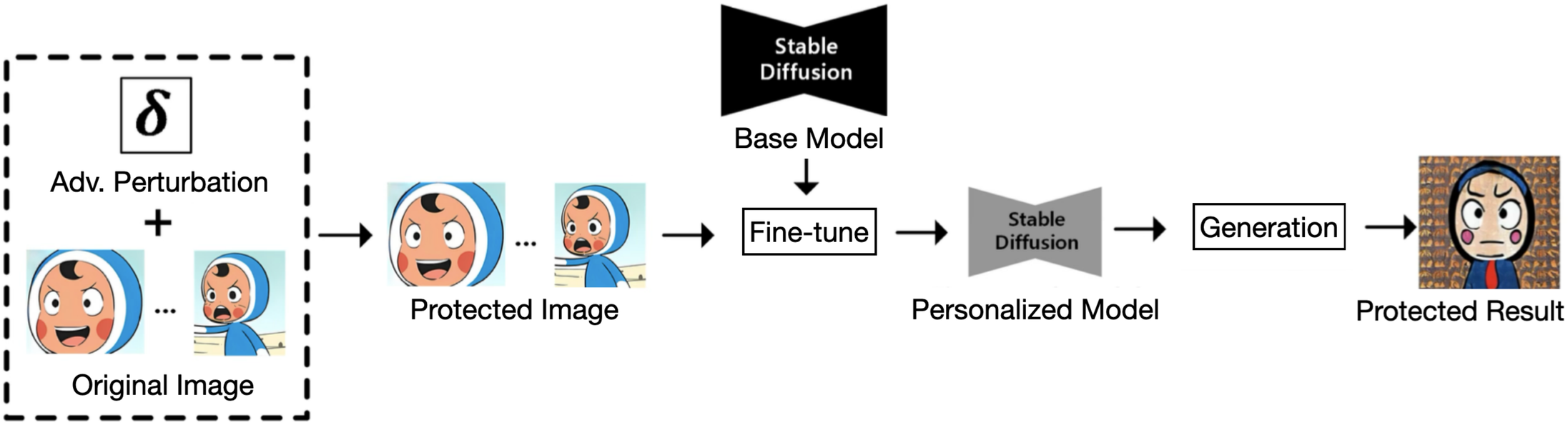


Introduction

- Perturbation-based Protection against Diffusion Models

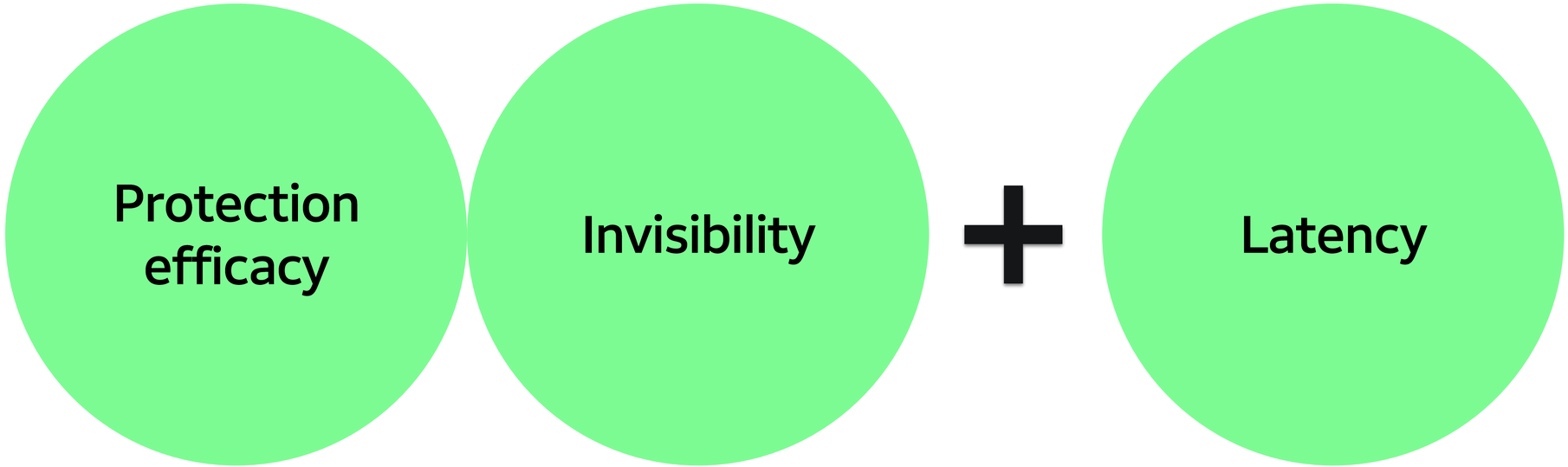
  - Methods designed to hinder generative AI from mimicking content and artistic styles.

  - Adversarial Perturbation: Subtle image alterations (imperceptible to humans) inserted to disrupt generative AI's imitation capabilities.



Introduction

- Existing **Perturbation-based Protection methods**
  - Primarily focus on balancing protection efficacy and invisibility, thus limiting practical use.
  - Typically optimize perturbations at inference, resulting in very high latency.
  - e.g. PhotoGuard takes 5 mins to protect 512^2 image on a CPU, 7 secs on an A100 GPU



Comprehensive Protection Considerations in Practical Scenarios

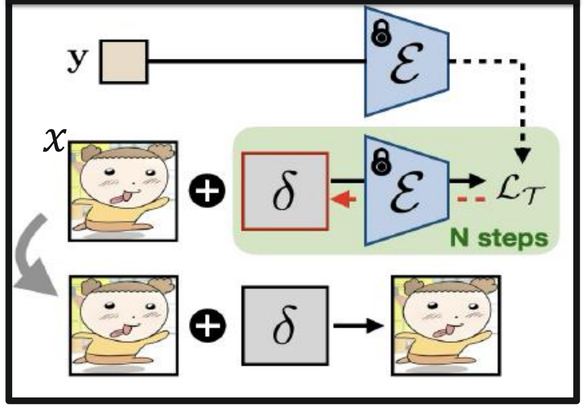
Background

- Iterative Optimization
  - Most existing methods iteratively optimize protective perturbations during inference.
  - Optimizing perturbations for each given image results in significant latency.
  - <u>Improvement Area</u>: Low-latency perturbation generation

Training

N/A

Inference



x : original image, δ : adversarial perturbation

y : target image

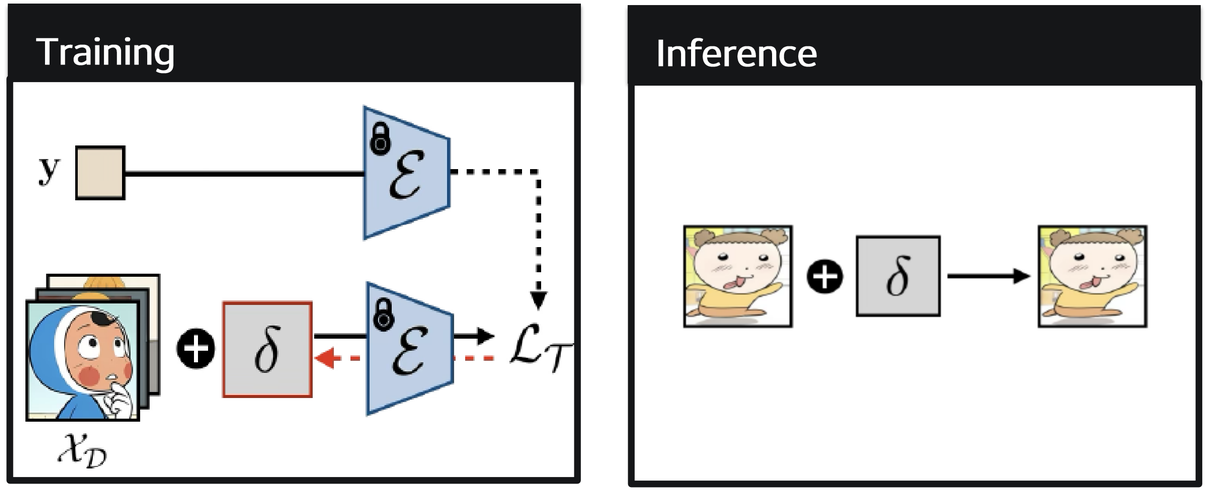
ε : VAE encoder, $\boldsymbol{z} = \varepsilon(x),\ \boldsymbol{z}_y = \varepsilon(y)$

$$\mathcal{L}_T(\mathbf{x}) = -||\mathbf{z} - \mathbf{z}_y||_2^2$$

Overview of Iterative Optimization

Background

- Universal Adversarial Perturbation (UAP)
  - Pre-compute adversarial perturbations in advance; apply learned perturbations directly during inference.
  - Improves inference speed but results in lower protection efficacy compared to iterative optimization.
  - **Improvement Area**: Enhance perturbation capacity (currently limited by identical perturbations regardless of input images)



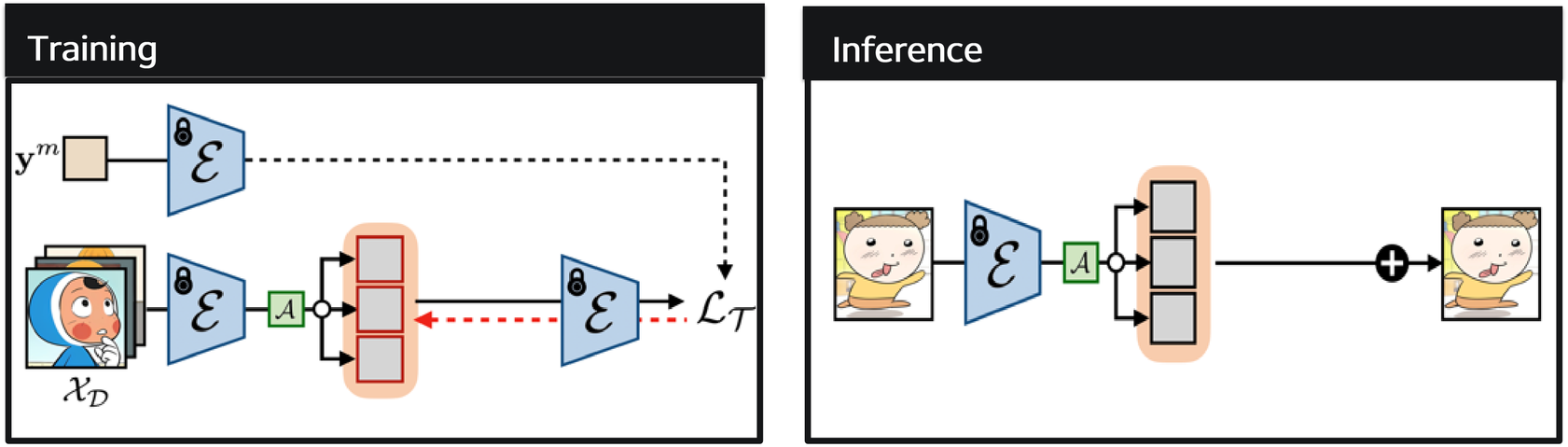
Overview of UAP

Method - FastProtect

- Mixture-of-Perturbation (MoP)
  - Improve protection capacity by employing multiple perturbations.
  - Adaptively select the perturbation best suited for each input image.
    - $\mathcal{A}$: K-means clustering based on VAE latents

$$\hat{\mathbf{x}} = \mathbf{x} + \delta_g + \Delta_k, \quad \text{where} \quad k = \mathcal{A}(\mathcal{E}(\mathbf{x}))$$

+ multi-layer protection (MLP) loss $\mathcal{L}_T(\mathbf{x}) = -||\mathbf{z} - \mathbf{z}_y||_2^2 - \frac{\lambda}{L}\sum_{l=1}^{L}||\mathcal{F}^l - \mathcal{F}_y^l||_2^2.$



Overview of MoP

Method - FastProtect

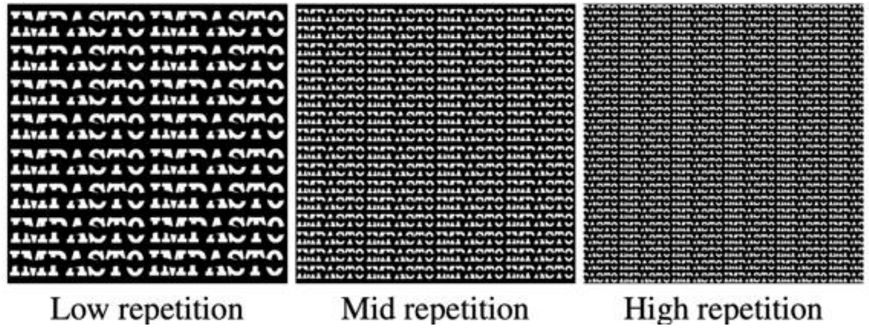
- Adaptive Targeted Protection

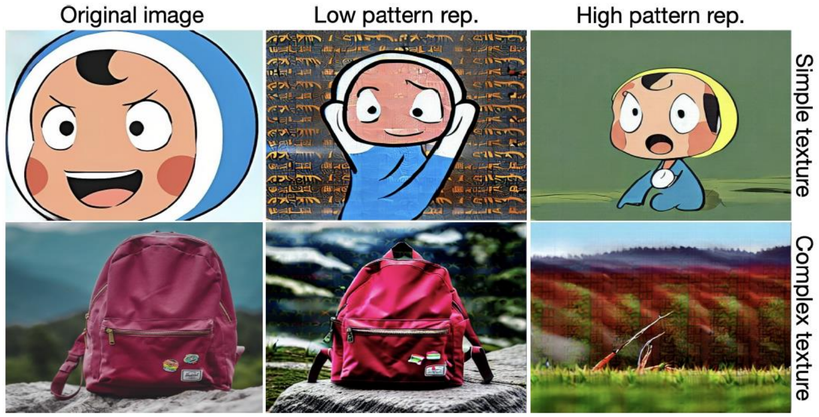
  - Protection performance varies depending on target image → During inference, select perturbations corresponding to the target image optimized for the input.

  - Train separate MoP for each target image (y^l, y^m, y^h)

  - H : Calculate entropy in VAE latent space → Select target image with entropy closest to input image.

$$\hat{\mathbf{x}} = \mathbf{x} + \delta_g^t + \Delta_k^t, \quad t = \underset{i \in \{l,m,h\}}{\arg\min} \, ||\mathcal{H}(\mathbf{z}) - \mathcal{H}(\mathbf{z}_y^i)||_1$$





Method - FastProtect

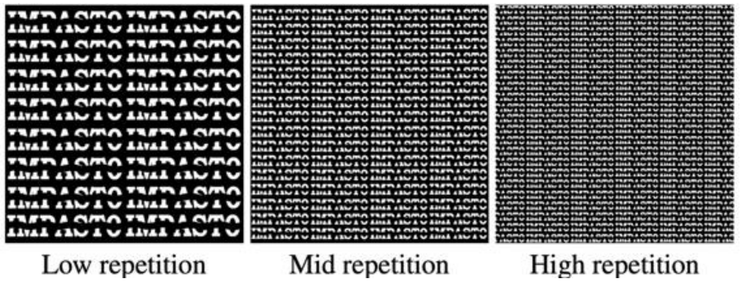
- Adaptive Targeted Protection

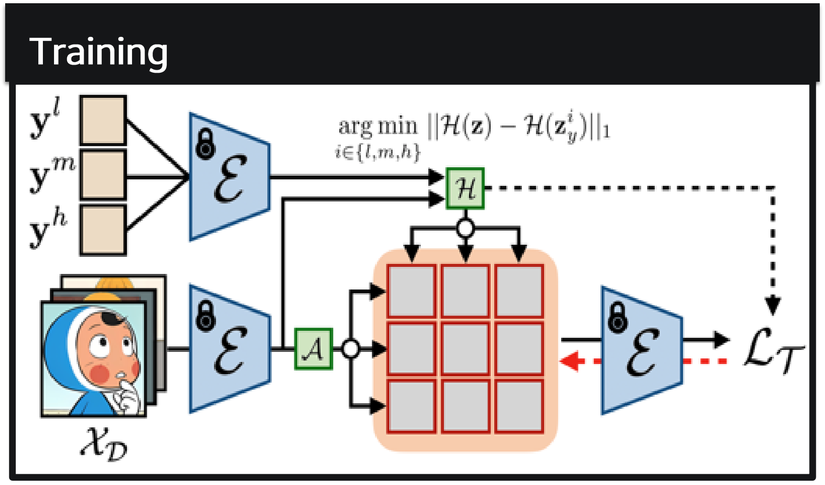
  - Protection performance varies depending on target image → During inference, select perturbations corresponding to the target image optimized for the input.

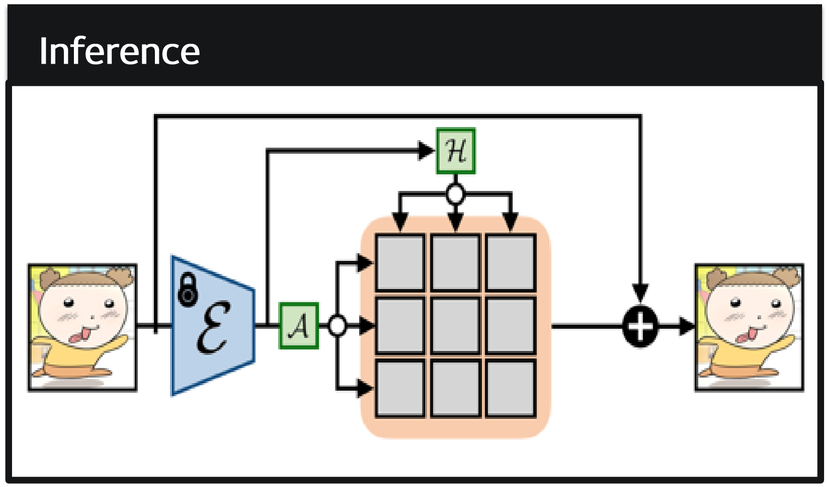
  - Train separate MoP for each target image (y^l, y^m, y^h)

  - H : Calculate entropy in VAE latent space → Select target image with entropy closest to input image.

$$\hat{\mathbf{x}} = \mathbf{x} + \delta_g^t + \Delta_k^t, \quad t = \underset{i \in \{l,m,h\}}{\arg\min} \, ||\mathcal{H}(\mathbf{z}) - \mathcal{H}(\mathbf{z}_y^i)||_1$$





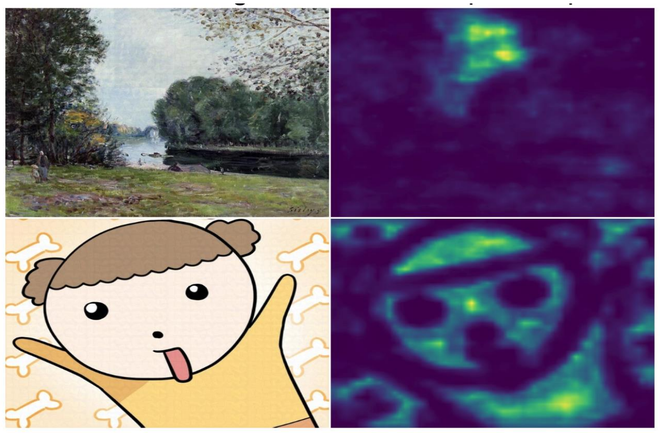


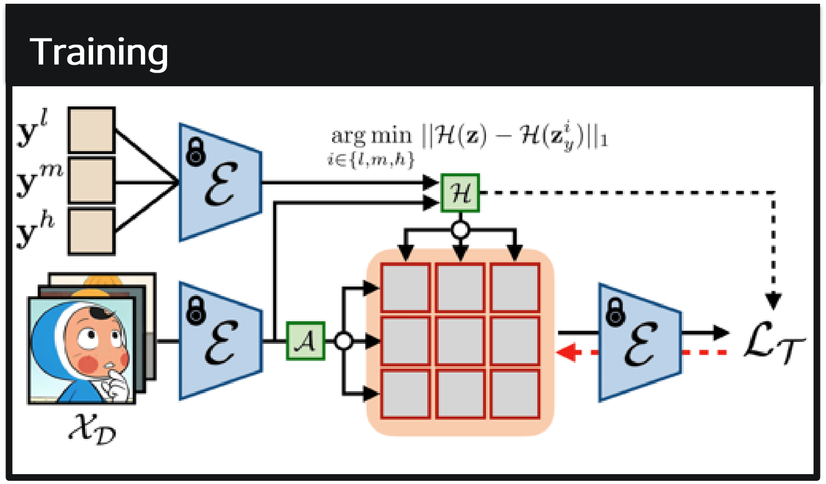
Method - FastProtect

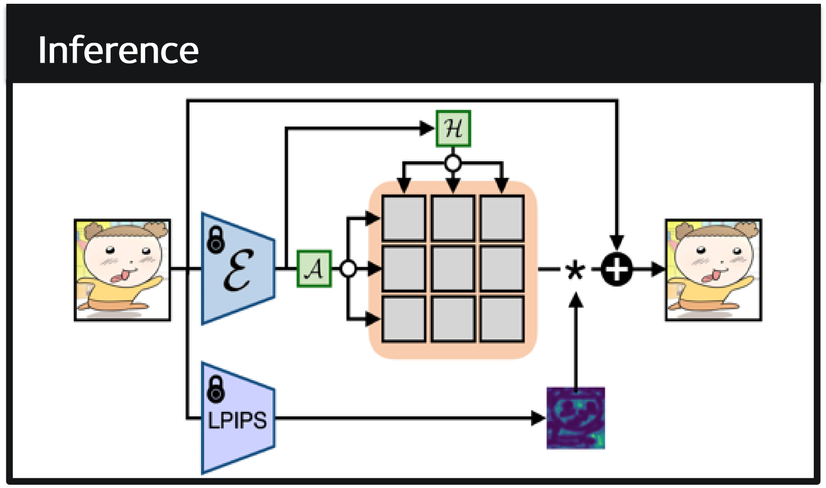
- Adaptive Protection Strength

  - Apply MoP initially at inference to generate temporary protected images.
  - Compute a spatial perceptual map (LPIPS) between input and protected images to guide perturbation.
  - Improve invisibility by applying stronger perturbations to less noticeable regions identified by LPIPS, a reliable predictor of human perception..

$$\hat{\mathbf{x}} = \mathbf{x} + \mathcal{S}(1 - \mathbf{M}) * (\delta_g^t + \Delta_k^t)$$

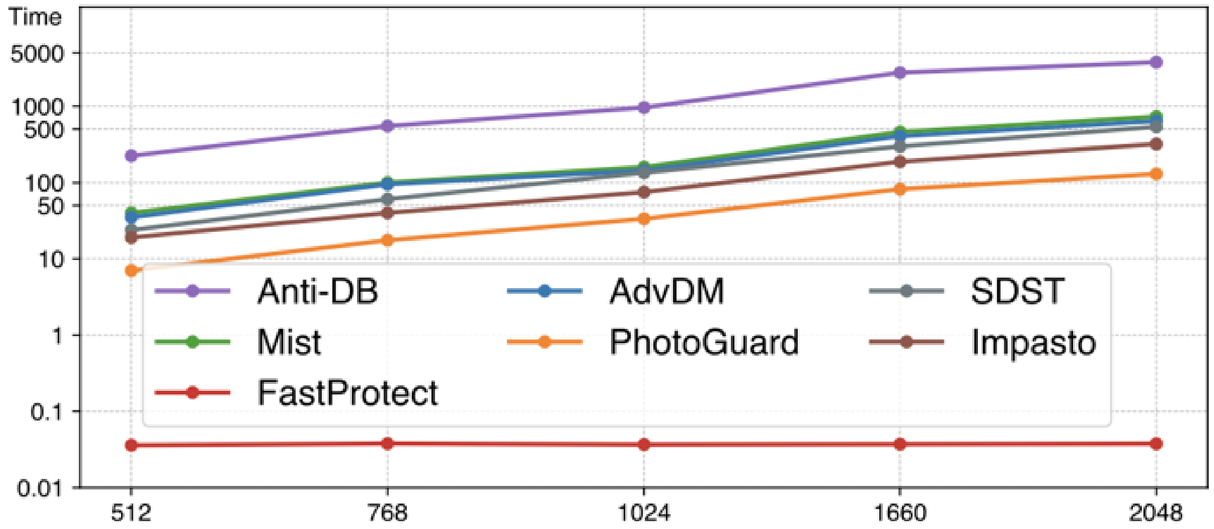






Experimental Results

- Inference Latency Performance
  - Achieved significantly faster inference speed compared to existing learning prevention methods
  - CPU: 2.9 s, GPU (A100): 0.04 s (for 512×512 images)



(a) Inference latency (log-scaled) vs. image size

Experimental Results

- Quantitative Evaluation

  - Achieves significantly improved inference speed while maintaining high invisibility and protection performance across diverse domain datasets.

  - FastProtect showed acceptable robustness against noise, JPEG, and resizing compared to baselines.

Method	Latency	Object	Face	Painting	Cartoon
	CPU / GPU	Invisibility: DISTS (↓) / Efficacy: FID (↑)			
AdvDM [28]	1210s / 35s	0.197 / 220.0	0.173 / 303.8	0.153 / **357.6**	0.271 / 212.5
PhotoGuard [46]	370s / 7s	0.203 / **223.0**	0.189 / 308.7	**0.107** / 350.9	0.209 / 219.1
Anti-DB [51]	7278s / 225s	0.239 / 214.4	0.162 / 301.4	0.114 / 347.7	0.294 / **225.4**
Mist [27]	1440s / 40s	0.185 / 217.2	0.154 / 307.5	0.129 / 357.0	0.223 / 223.7
Impasto [1]	830s / 19s	0.201 / 213.8	0.198 / 298.4	0.123 / 352.4	0.207 / 215.5
SDST [56]	1410s / 24s	0.242 / 219.2	0.244 / 302.1	0.152 / 354.1	0.237 / 222.7
FastProtect	**2.9s / 0.04s**	**0.155 / 223.0**	**0.149 / 308.9**	0.110 / 356.1	**0.186** / 220.3

Domain	Method	Invisibility	Countermeasure		Arbitrary Size
			Noise	JPEG	
Subject	PhotoGuard	0.203	193.3	**193.2**	193.5
	FastProtect	**0.155**	**214.4**	191.3	**219.1**
Cartoon	PhotoGuard	0.209	**192.9**	193.1	190.7
	FastProtect	**0.186**	191.8	**199.9**	**204.0**

Experimental Results

- Qualitative Evaluation
  - Minimal image degradation (high Invisibility) from adversarial perturbations compared to other methods.
  - Target image characteristics clearly reflected in protected images as intended.

Conclusion

- We propose FastProtect, which achieves real-time protection against diffusion models. Our work is the first to address the critical issue of latency in this task.

- FastProtect, integrates perturbations pre-training with adaptive inference schemes, meeting all requirements for a practical protection solution.

- We validate FastProtect in various scenarios, showing that despite its speed and invisibility, it retains protection efficacy, robustness, and generalization.

Thanks for Listening

- For more information, please visit:

  https://webtoon.github.io/impasto

- Fell free to contact **Namhyuk Ahn** and **Seung-Hun Nam** via:

  nhahn@inha.ac.kr

  shnam1520@gmail.com