Carnegie Mellon University

Generative Visual Prompt:

Unifying Distributional Control of Pre-Trained Generative Models

Chen Henry Wu, Saman Motamed, Shaunak Srivastava, Fernando De la Torre
Carnegie Mellon University
henrychenwu@cmu.edu @ChenHenryWu
Presented @NeurIPS 2022
https://chenwu98.github.io/PromptGen/

What is a good generative model?

Faithfully modeling the data distribution?

Uncurated samples from StyleGAN2

Things not in data…

Controllability (hard to label all concepts, e.g., “baby”)

Things not in data…

Controllability (hard to label all concepts, e.g., "baby")

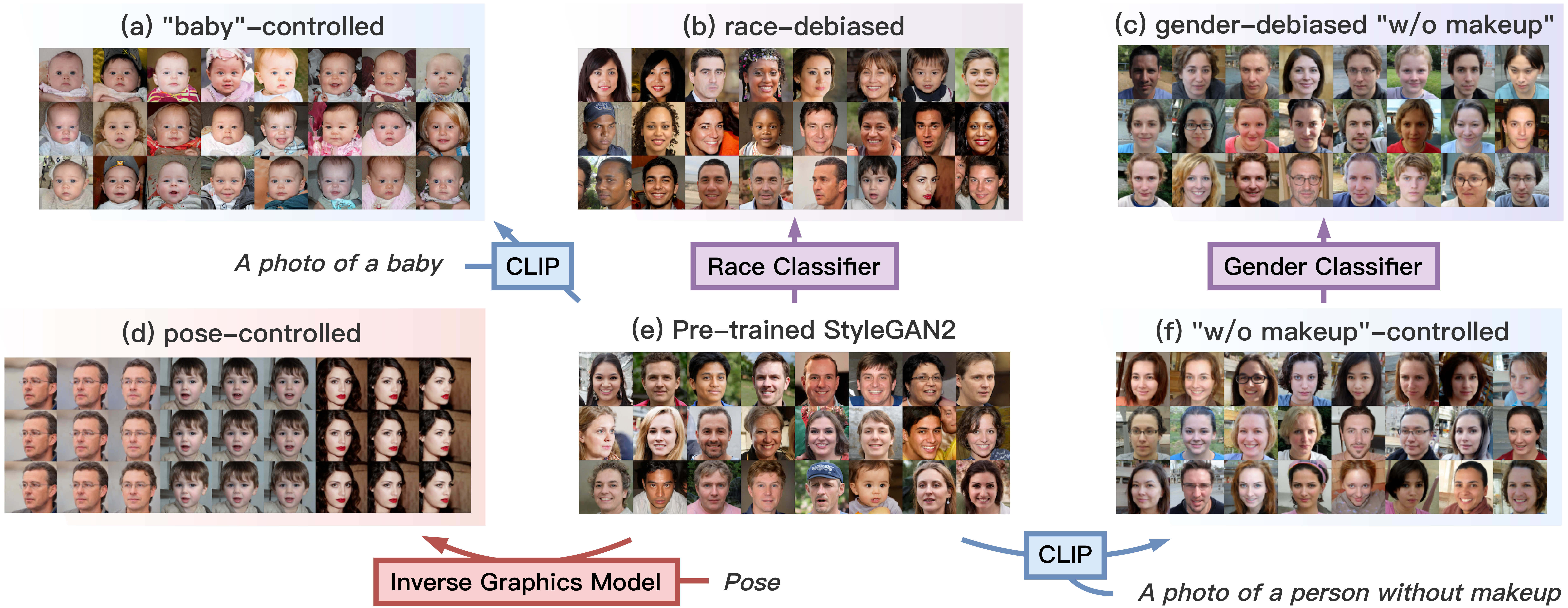
Fairness (hard to build a truly fair training set, e.g., across races)

PromptGen is a remedy!

A **feed-forward** neural network to model desired distributions in latent space

PromptGen is a remedy!

A **feed-forward** neural network to model desired distributions in latent space



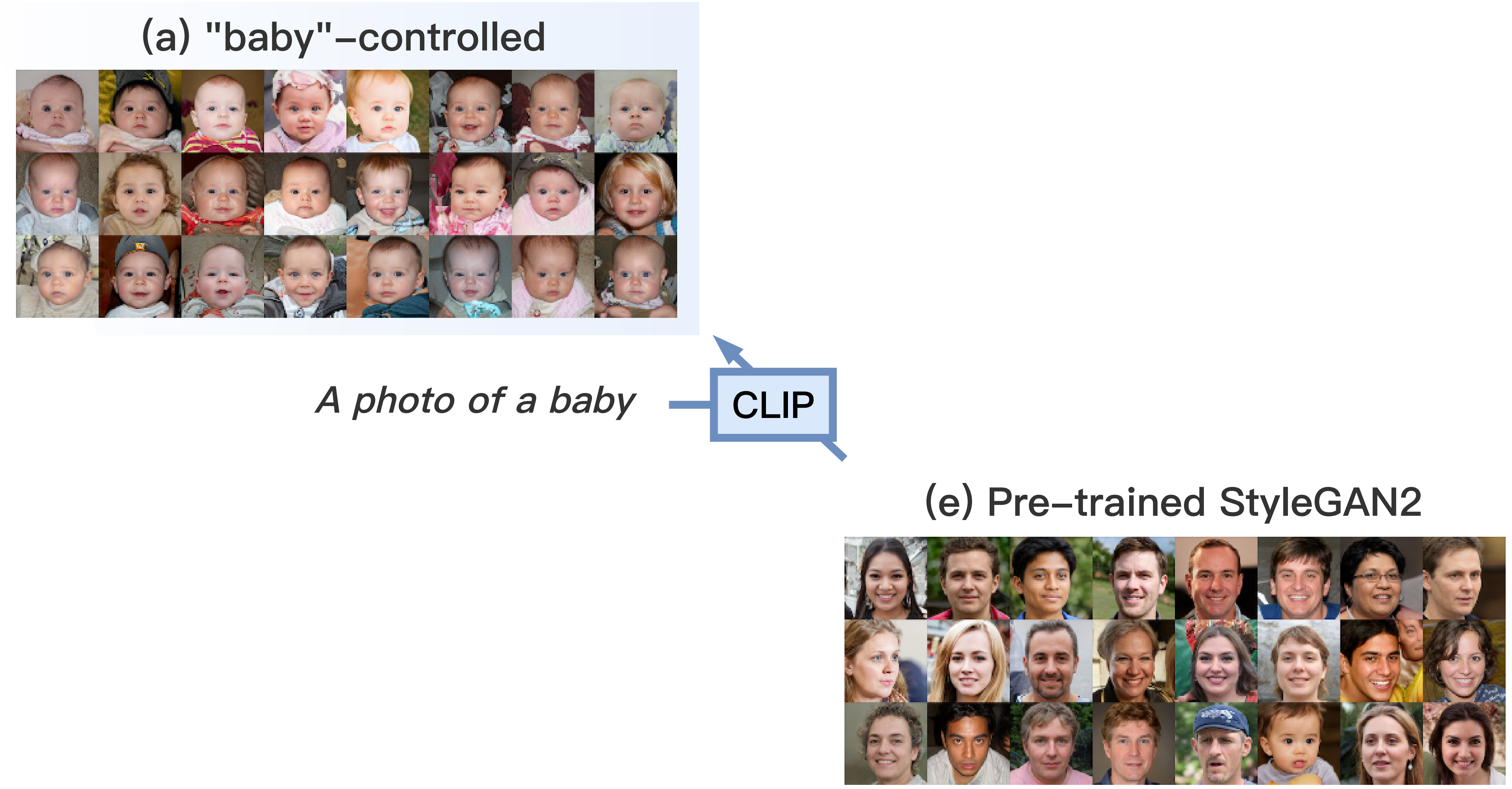
PromptGen is a remedy!

A **feed-forward** neural network to model desired distributions in latent space

(e) Pre–trained StyleGAN2

PromptGen is a remedy!

A **feed-forward** neural network to model desired distributions in latent space

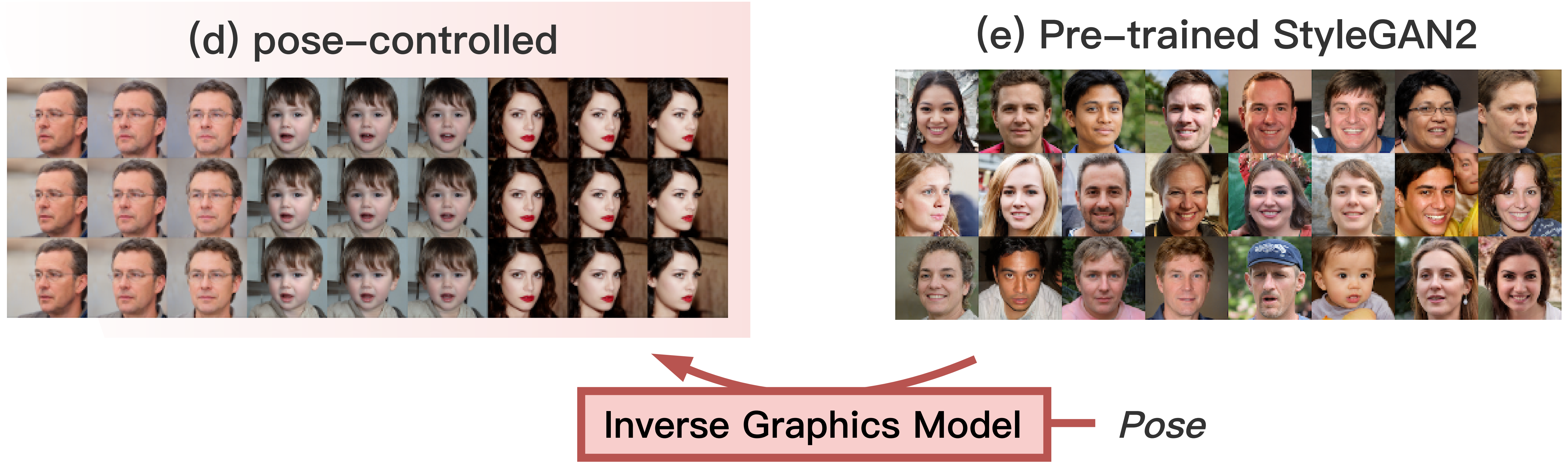


PromptGen is a remedy!

A **feed-forward** neural network to model desired distributions in latent space

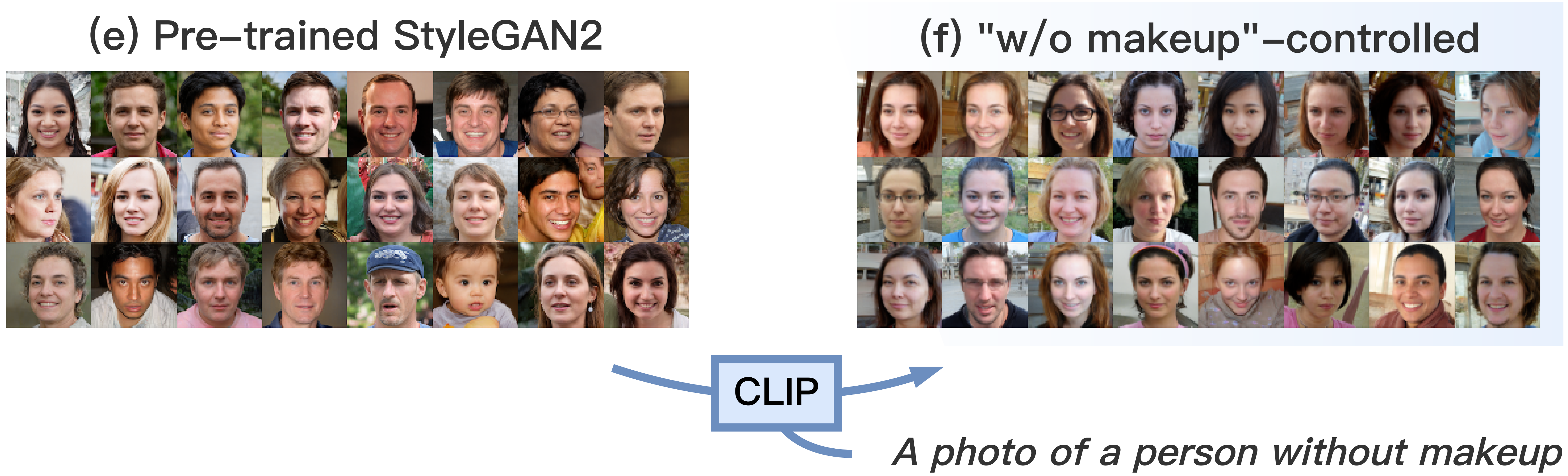
PromptGen is a remedy!

A **feed-forward** neural network to model desired distributions in latent space



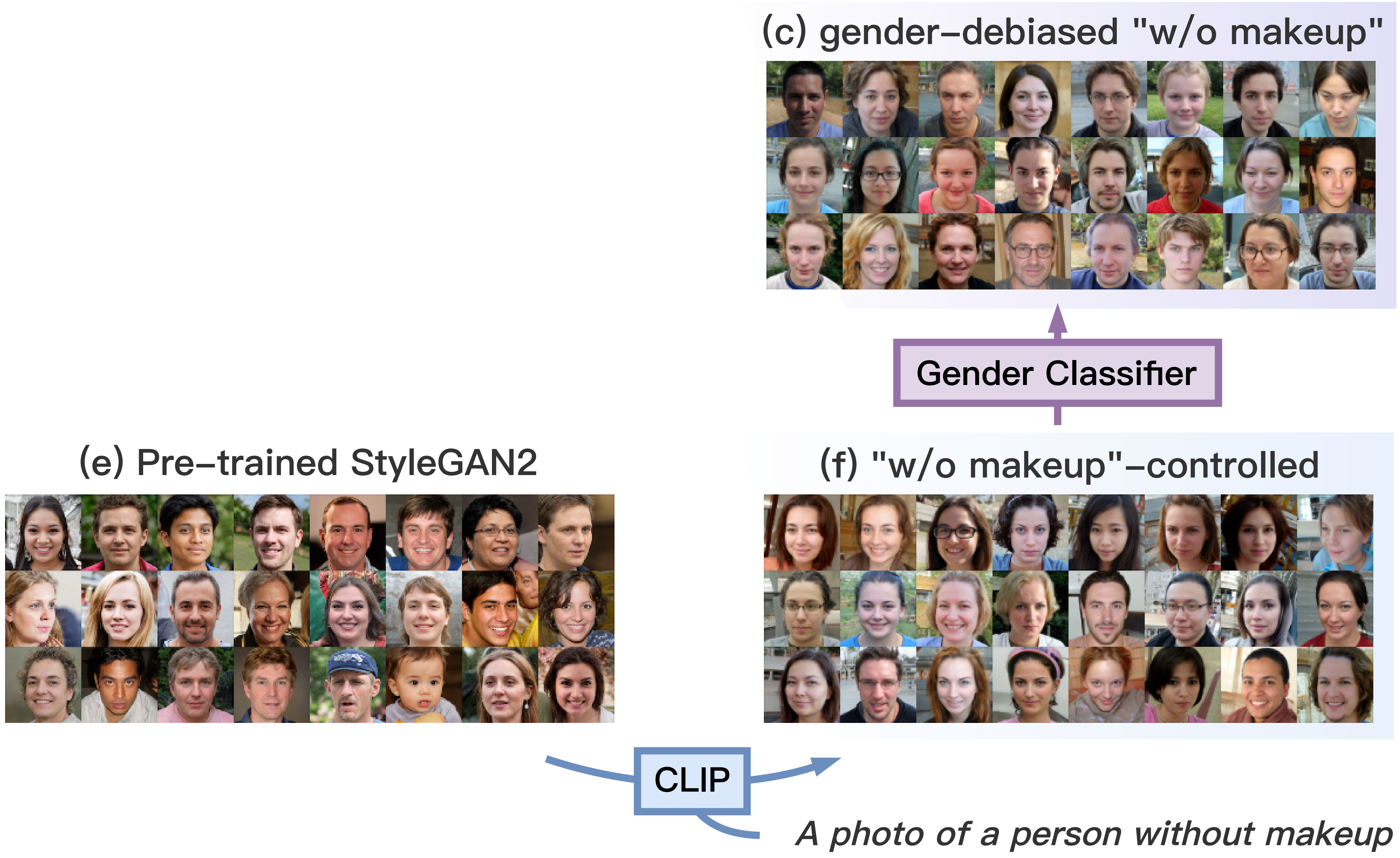
PromptGen is a remedy!

A **feed-forward** neural network to model desired distributions in latent space



PromptGen is a remedy!

A **feed-forward** neural network to model desired distributions in latent space



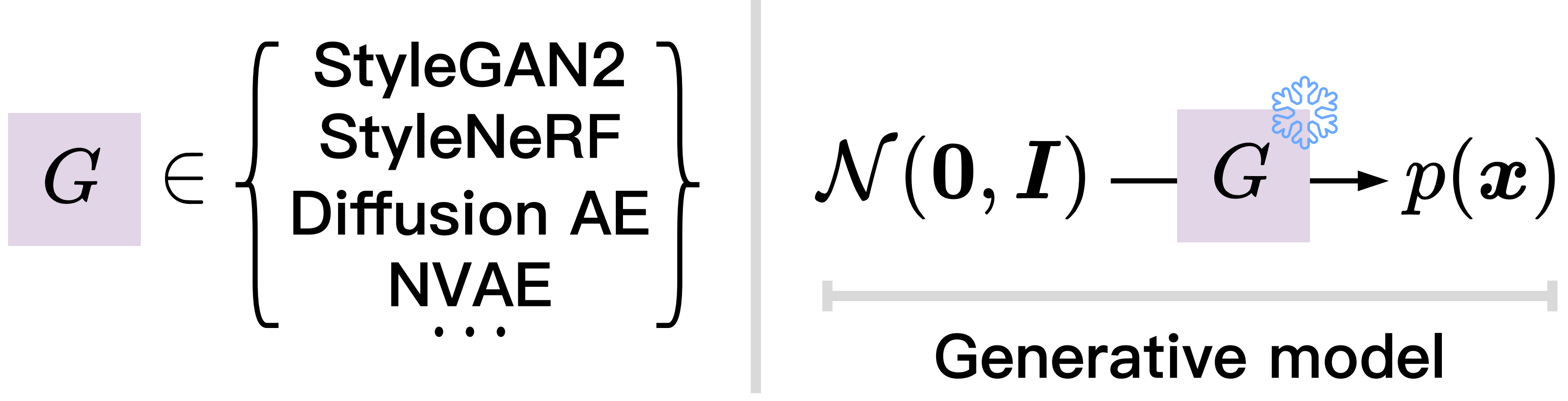
Framework overview

1. User specifies a generative model;

$$\boxed{G} \in \left\{ \begin{array}{c} \text{StyleGAN2} \\ \text{StyleNeRF} \\ \text{Diffusion AE} \\ \text{NVAE} \\ \cdots \end{array} \right\}$$

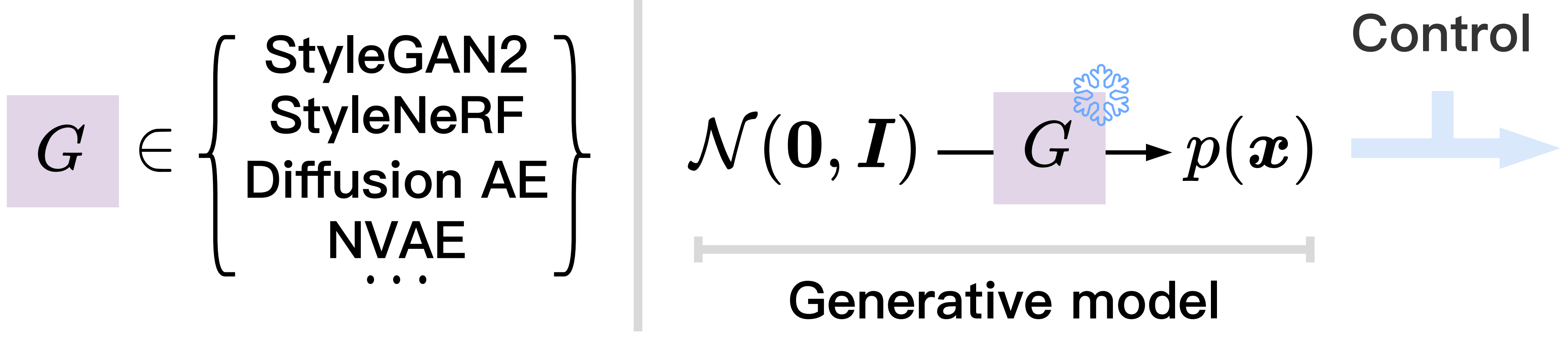
Framework overview

1. User specifies a generative model;



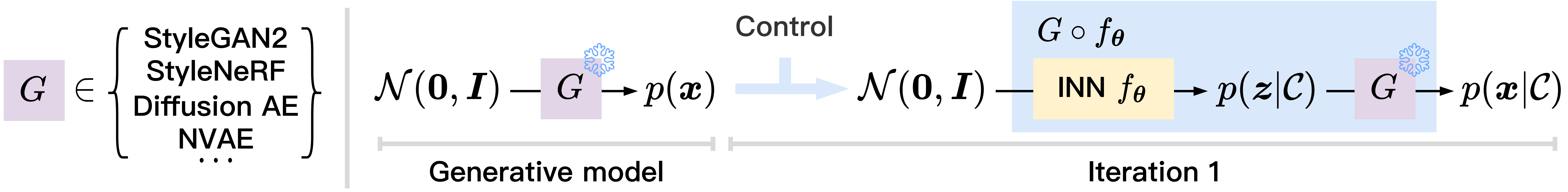
Framework overview

1. User specifies a generative model;
2. User specifies a control;



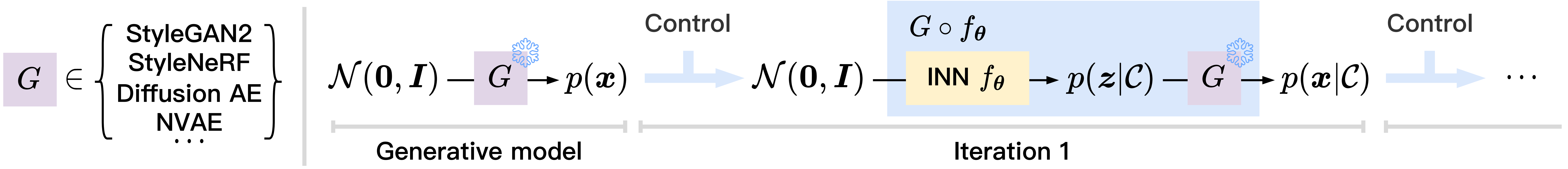
Framework overview

1. User specifies a generative model;
2. User specifies a control;
3. Approximates the control with an invertible neural network;



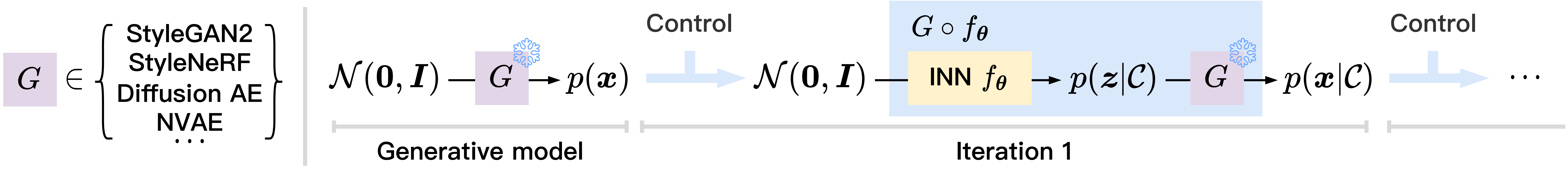
Framework overview

1. User specifies a generative model;
2. User specifies a control;
3. Approximates the control with an invertible neural network;
4. If iterative control is needed: functional composition and repeat 2 & 3.



Framework overview

1. User specifies a generative model;
2. User specifies a control;
3. Approximates the control with an invertible neural network;
4. If iterative control is needed: functional composition and repeat 2 & 3.



How to specify a control?

Image-space energy (lower is better)

Inverse graphics energy

$$E_{\text{inv-graphics}}(\boldsymbol{x}, \boldsymbol{\rho}) = d\langle f_{\mathcal{X}\to\mathcal{P}}(\boldsymbol{x}), \boldsymbol{\rho}\rangle^2$$

Classifier energy

$$E_{\text{classifier}}(\boldsymbol{x}, a) = -\log P(a|\boldsymbol{x})$$

CLIP energy

$$E_{\text{CLIP}}(\boldsymbol{x}, \boldsymbol{t}) = \frac{1}{L}\sum_{l=1}^{L}\left(1 - \cos\left\langle \text{CLIP}_{\text{img}}(\text{DiffAug}_l(\boldsymbol{x})), \text{CLIP}_{\text{text}}(\boldsymbol{t})\right\rangle\right)$$

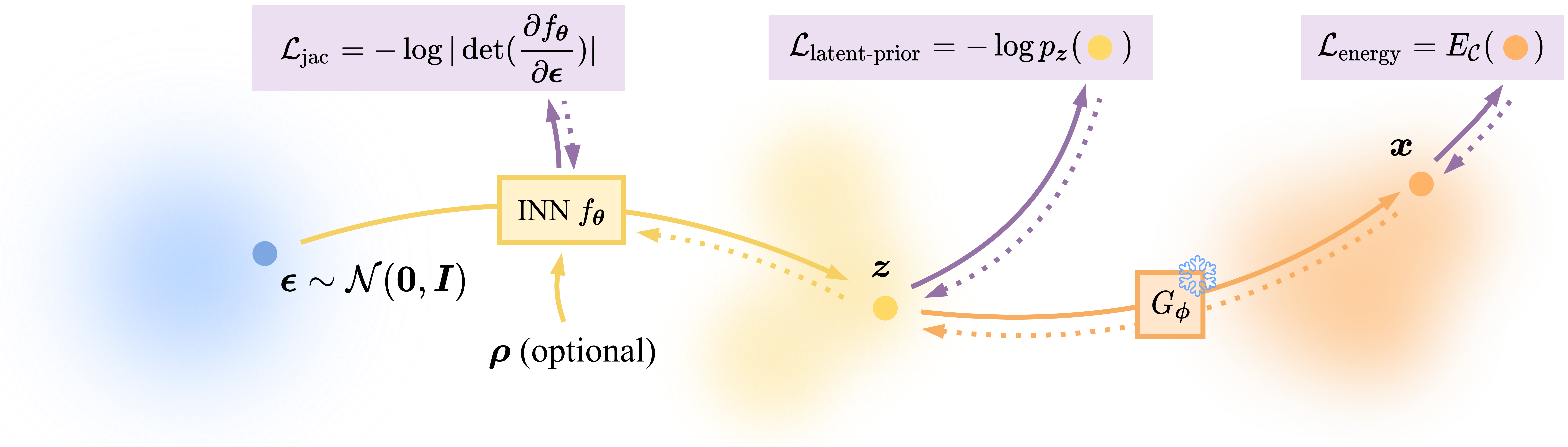
Moment constraint

$$p(\boldsymbol{x}|\mathcal{C}) = \underbrace{\arg\min_{p(\boldsymbol{x}|\mathcal{C})} \mathbb{D}_{\text{KL}}(p(\boldsymbol{x}|\mathcal{C})\|p_{\boldsymbol{x}}(\boldsymbol{x}))}_{\text{Deviation from the pre-trained distribution}}, \quad \text{s.t. } \underbrace{\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|\mathcal{C})}[\boldsymbol{\gamma}(\boldsymbol{x})] = \boldsymbol{\mu}}_{\text{Moment constraint}}$$

How to approximate the control?

Optimize $\mathbb{D}_{\mathsf{KL}}(p_{\boldsymbol{\theta}}(\boldsymbol{z}) \| p(\boldsymbol{z}|\mathcal{C}))$ with a normalizing flow in the latent space

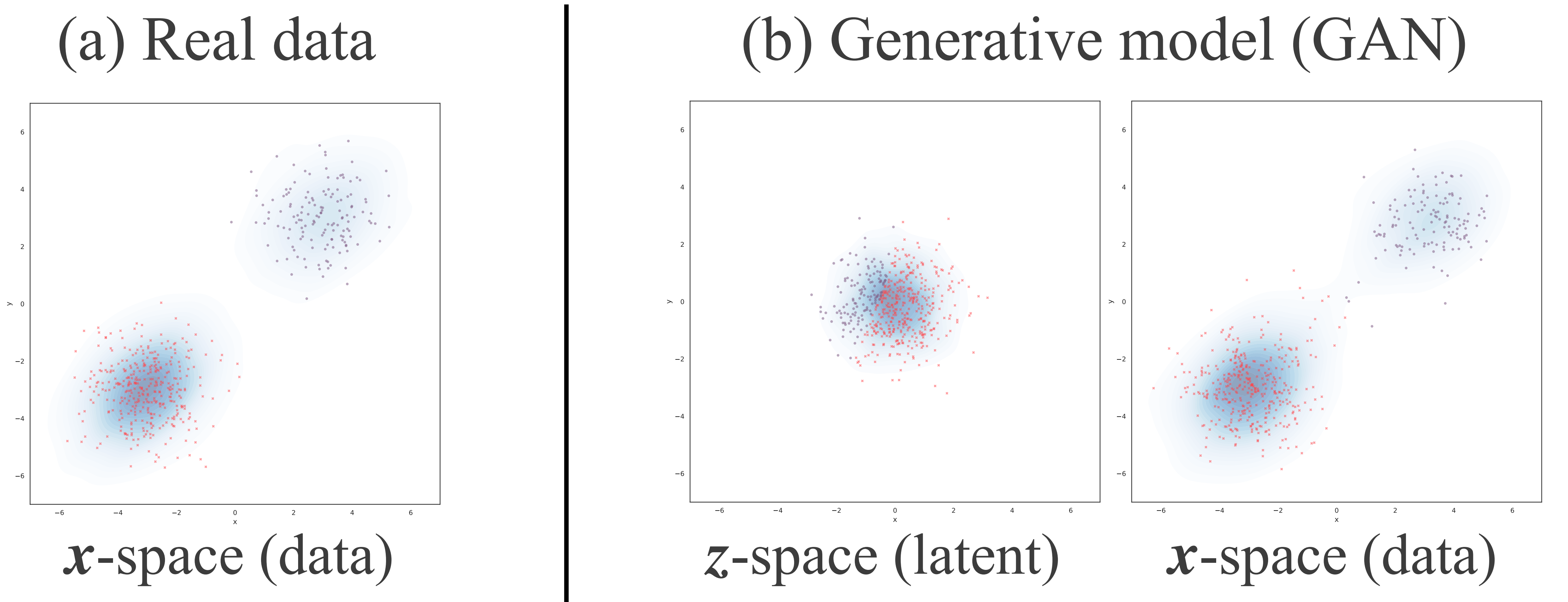
Training algorithm and objective:

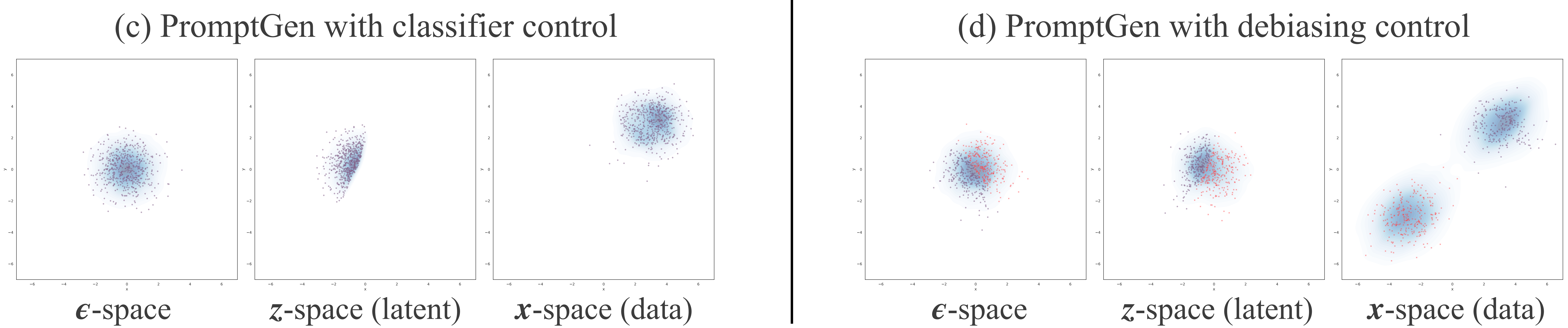


A synthetic 2D example

Real dist.
$$\begin{cases} p(\boldsymbol{x}) = 0.3 \cdot p(\boldsymbol{x} \mid a_1) + 0.7 \cdot p(\boldsymbol{x} \mid a_2) \\ p(\boldsymbol{x} \mid a_1) = N\left(\boldsymbol{x} \,\middle|\, (3,3)^\top, \boldsymbol{I}\right) \\ p(\boldsymbol{x} \mid a_2) = N\left(\boldsymbol{x} \,\middle|\, (-3,-3)^\top, \boldsymbol{I}\right) \end{cases}$$

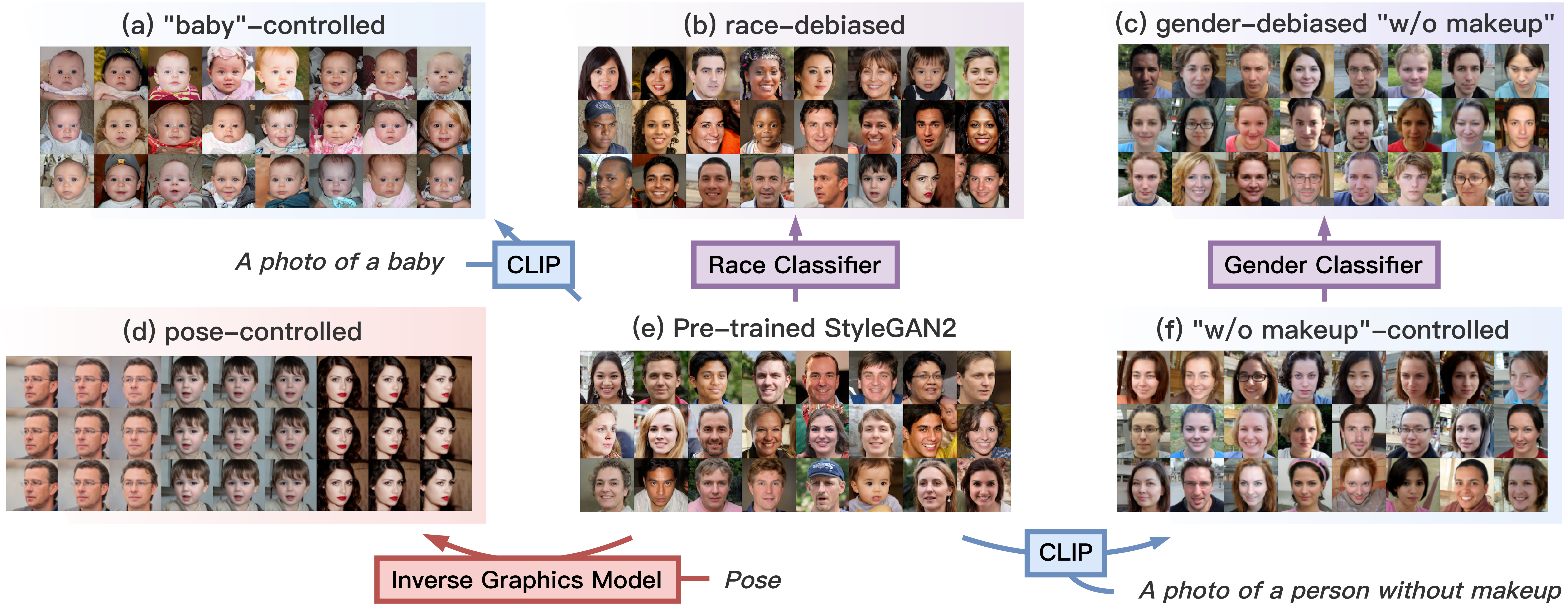
Fair classifier
$$p(a_i \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid a_i)}{p(\boldsymbol{x} \mid a_1) + p(\boldsymbol{x} \mid a_2)}$$





Real data experiments

Check our paper for details



Code available