Computational Models of Motion

Generative Motion Synthesis

Fatemeh Zargarbashi

08 May 2025

fatemeh.zargarbashi@inf.ethz.ch

Computational Robotics Lab

ETH zürich

Outline

- Recap
- Adversarial methods
- Diffusion-based methods
- Challenges in character animation

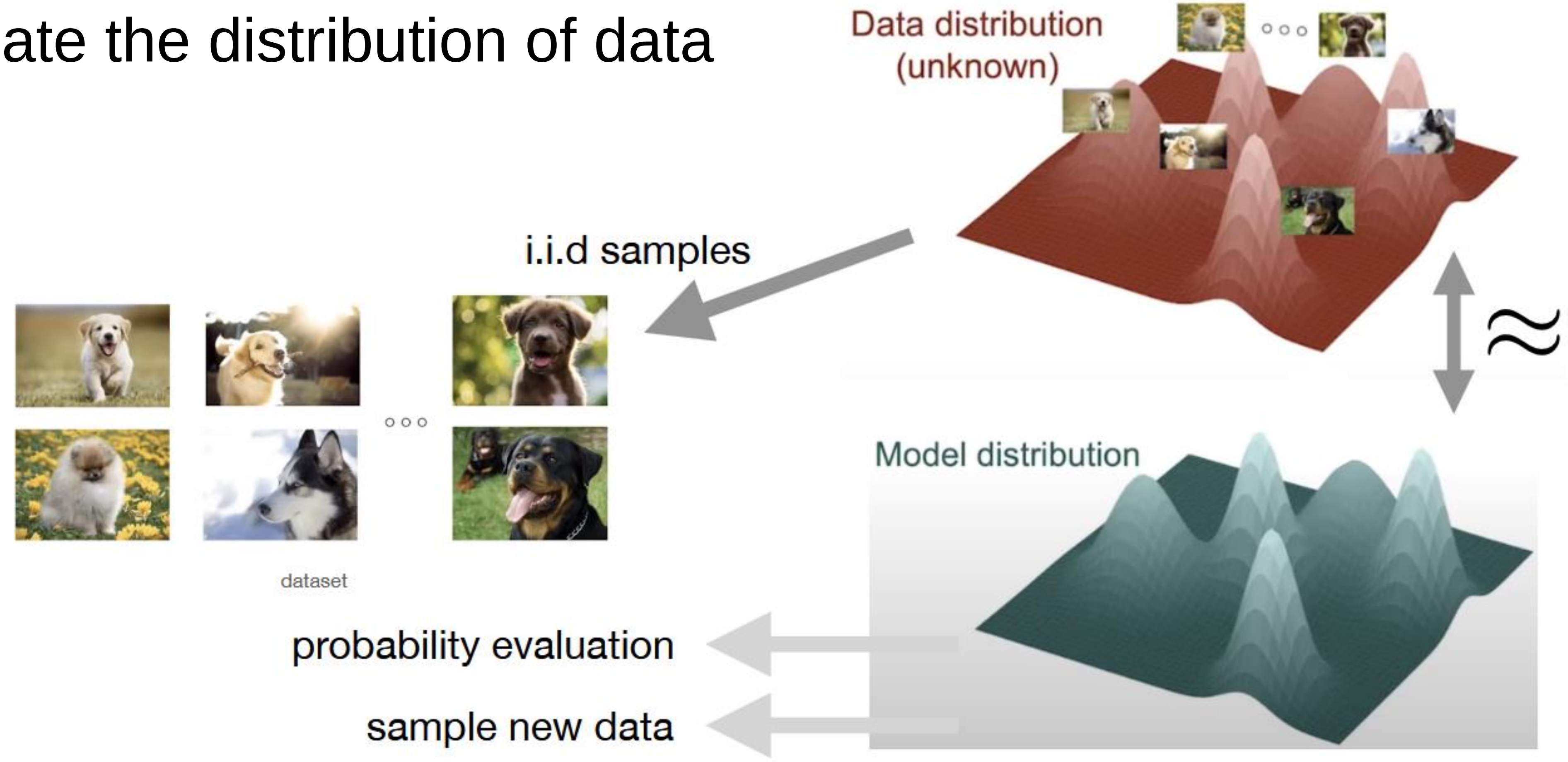
Recap

Given a motion dataset, how do we use it to generate natural and controllable motion?

- Planning based methods: behavior trees, motion graph, motion matching
- Phase-based methods: PFNN, MANN, DeepPhase
- Tracking-based methods for physics-based animation
- Generative models (VAEs) for motion generation

Recap – Generative models

- Estimate the distribution of data

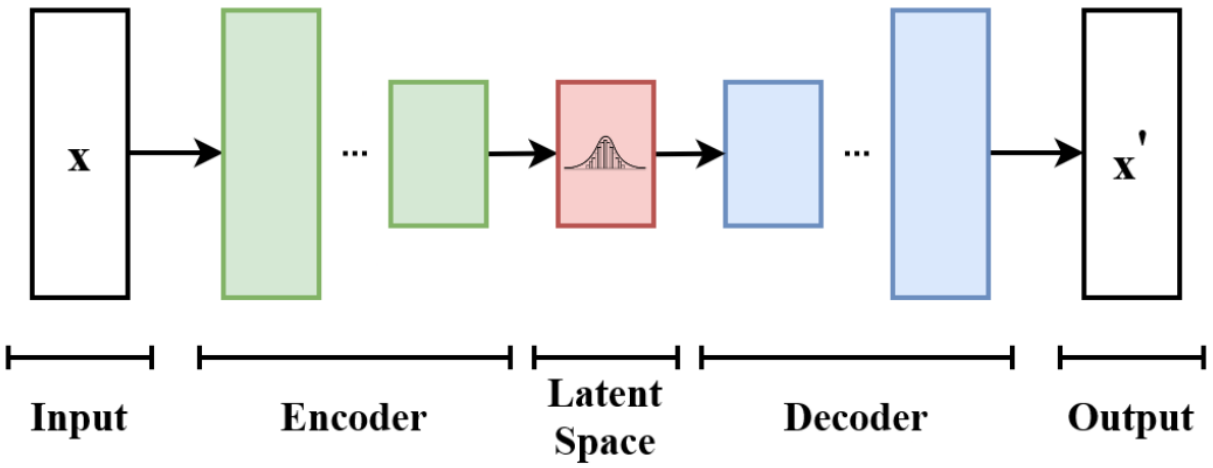


Recap – Generative models

- Auto-regressive models
- Variational Autoencoder (VAE)
- Generative Adversarial Network (GAN)
- Flow-based models
- Energy-based models
- Diffusion models (score-based models)

Recap – VAE

- Our data is generated from some latent variable with fixed distribution.
- Maximize (log-)likelihood of the data



VAEs offer smooth latent spaces but suffer from **blurry or mean-like outputs** (due to Gaussian likelihood and KL penalty).

Today's focus

- **Adversarial** methods
- **Diffusion** models

- How these are used in **motion generation** and **character control**

How can we generate highly realistic, diverse, and controllable character motion?

- Both **kinematic** and **physics-based** examples

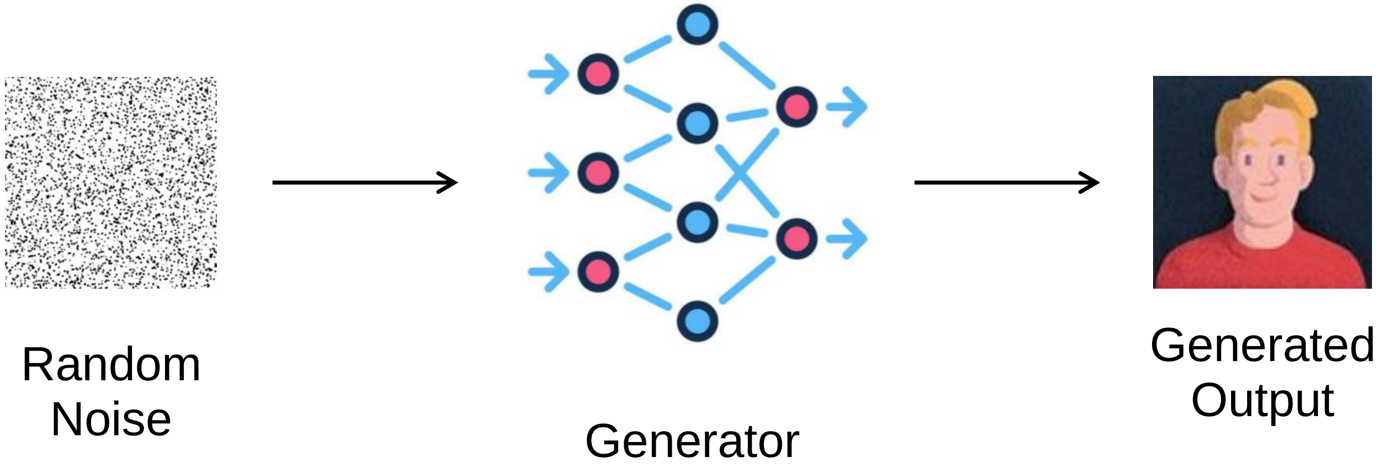
- Current Challenges in character animation

Outline

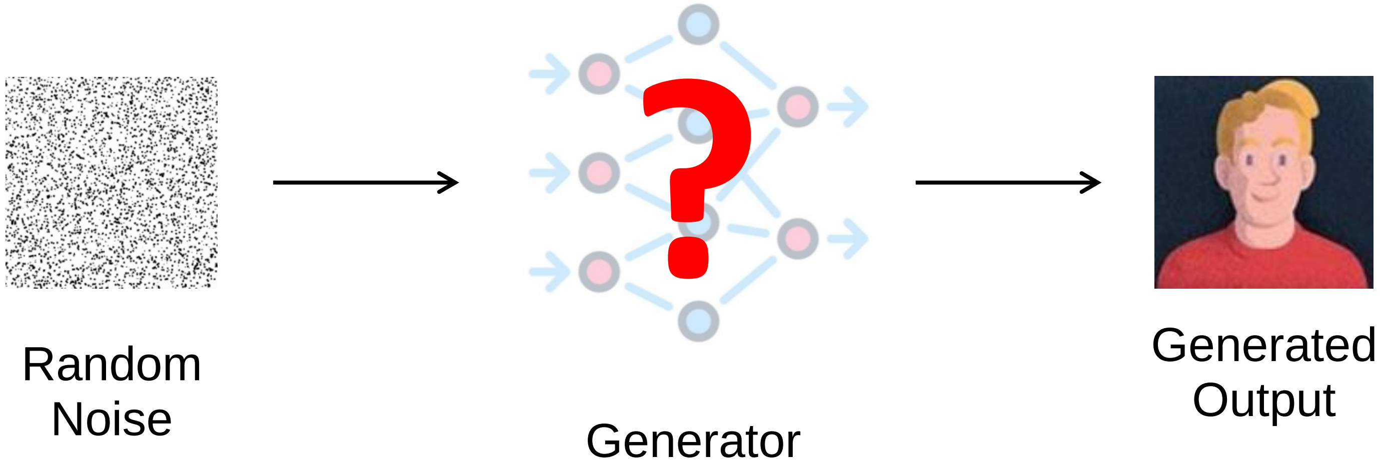
- Recap
- Adversarial methods
- Diffusion-based methods
- Challenges in character animation

Generative Adversarial Networks (GAN)

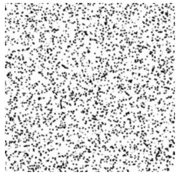
Generative Adversarial Networks (GAN)



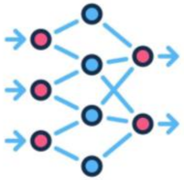
Generative Adversarial Networks (GAN)



Generative Adversarial Networks (GAN)



Random Noise



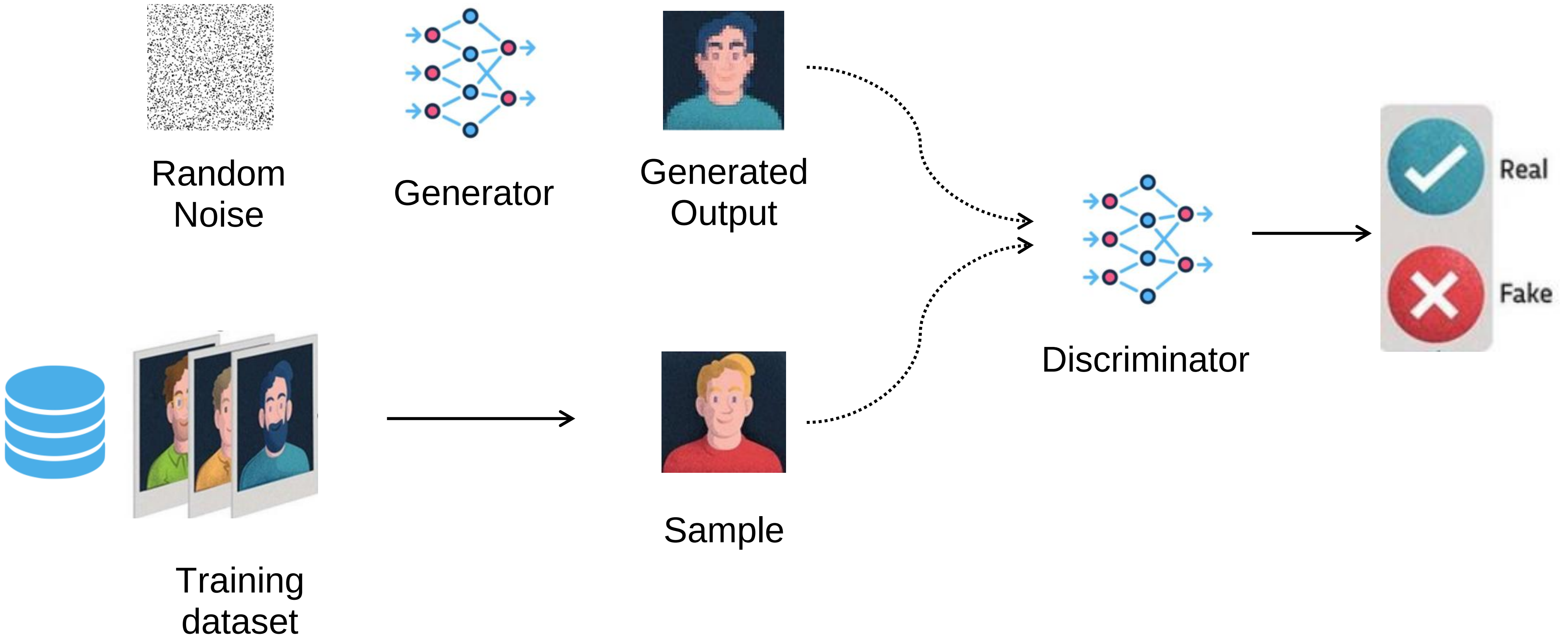
Generator

Generated Output

Training dataset

Sample

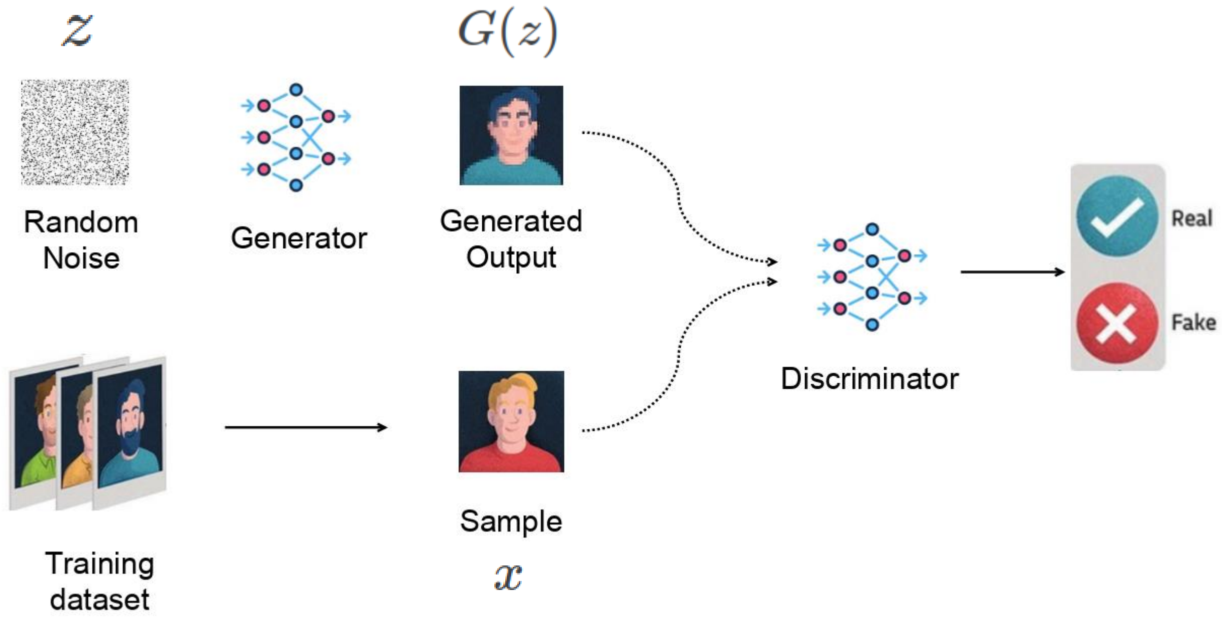
Generative Adversarial Networks (GAN)



Generative Adversarial Networks (GAN)

Discriminator: tries to classify real/fake images

Generator: tries to generate outputs that fool the discriminator into being real



$$L_D = \text{Error}(D(x), 1) + \text{Error}(D(G(z)), 0)$$

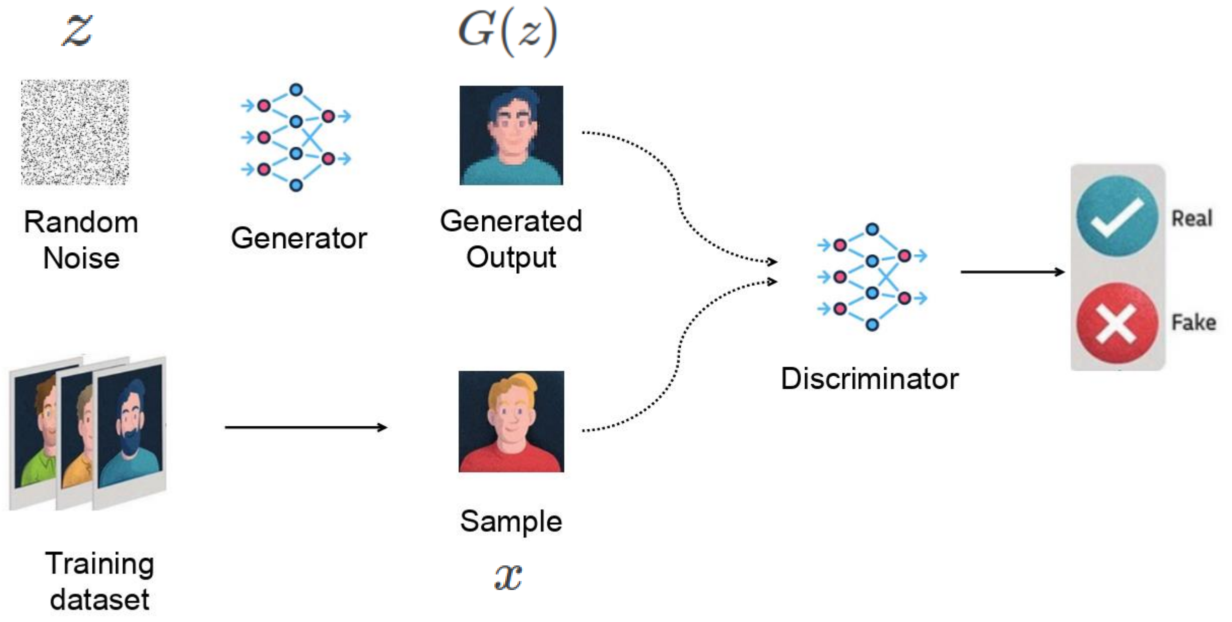
$$L_G = \text{Error}(D(G(z)), 1)$$

Goodfellow, Ian, et al. "Generative adversarial networks." *Communications of the ACM* 63.11 (2020): 139-144.
https://jaketae.github.io/study/gan-math/

Generative Adversarial Networks (GAN)

Discriminator: tries to classify real/fake images

Generator: tries to generate outputs that fool the discriminator into being real



$$L_D = \text{Error}(D(x), 1) + \text{Error}(D(G(z)), 0) = \max_D \{\log(D(x)) + \log(1 - D(G(z)))\}$$

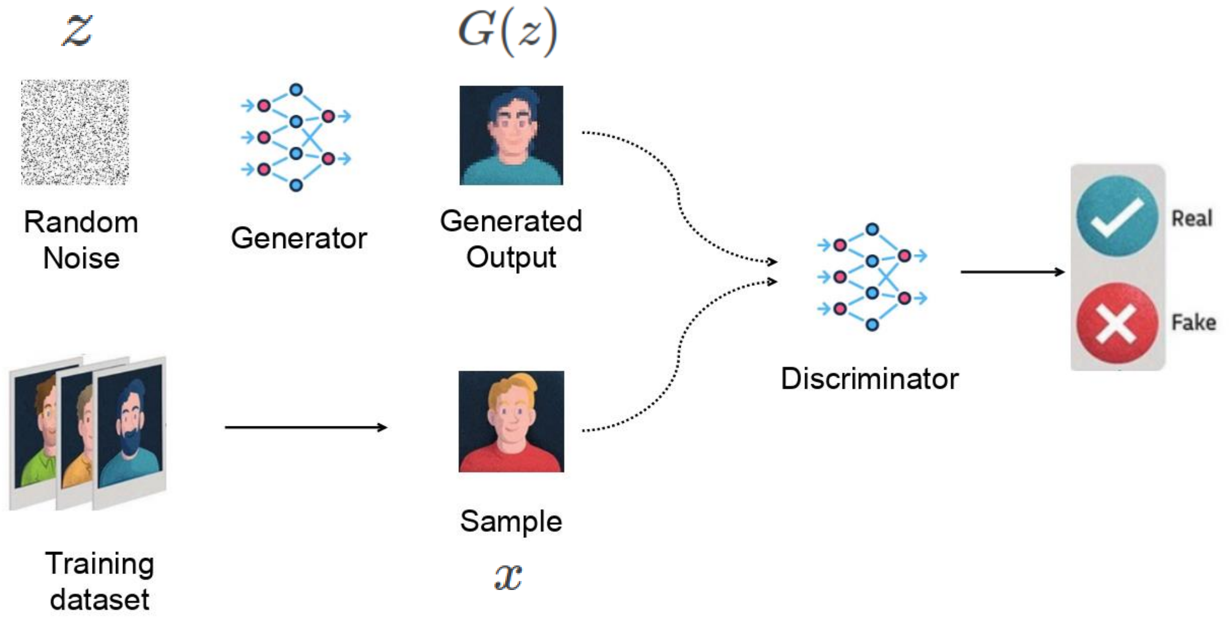
$$L_G = \text{Error}(D(G(z)), 1) = \min_G \{\log(1 - D(G(z)))\}$$

Goodfellow, Ian, et al. "Generative adversarial networks." *Communications of the ACM* 63.11 (2020): 139-144.
https://jaketae.github.io/study/gan-math/

Generative Adversarial Networks (GAN)

Discriminator: tries to classify real/fake images

Generator: tries to generate outputs that fool the discriminator into being real



$$L_D = \mathrm{Error}(D(x), 1) + \mathrm{Error}(D(G(z)), 0) = \max_D \{\log(D(x)) + \log(1 - D(G(z)))\}$$

$$L_G = \mathrm{Error}(D(G(z)), 1) = \min_G \{\log(1 - D(G(z)))\}$$

$$V(G, D) = \min_G \max_D \{\log(D(x)) + \log(1 - D(G(z)))\}$$

Goodfellow, Ian, et al. "Generative adversarial networks." *Communications of the ACM* 63.11 (2020): 139-144.
https://jaketae.github.io/study/gan-math/

Generative Adversarial Networks (GAN)

Training GANs requires **good balance** between generator and discriminator

$$V(G, D) = \min_G \max_D \{\log(D(x)) + \log(1 - D(G(z)))\}$$

- If discriminator is **strong**:

  generator gets no feedback for small improvements

- If discriminator is **weak**:

  can’t distinguish real/fake, so no informative feedback

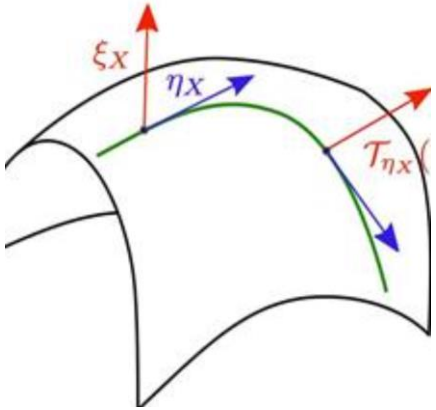
Mescheder, Lars, Andreas Geiger, and Sebastian Nowozin. "Which training methods for GANs do actually converge?." *International conference on machine learning*. PMLR, 2018.

Generative Adversarial Networks (GAN)

Issue: the discriminator may assign nonzero gradients on the manifold of real data samples

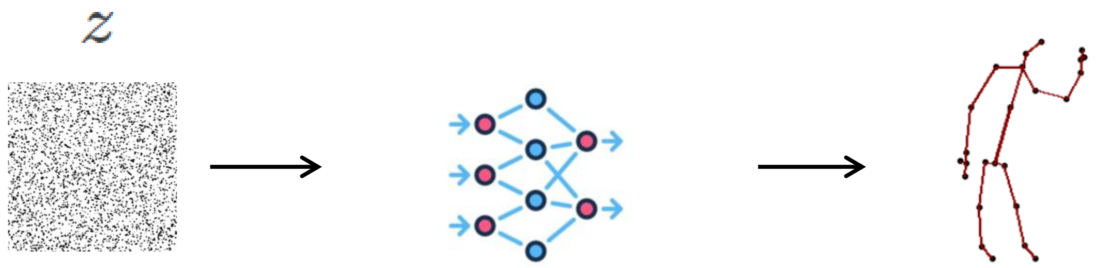
Solution: add gradient penalty to the discriminator loss

$$\mathrm{E}_{p_{\mathcal{D}}(x)}\left[\|\nabla D_{\psi}(x)\|^2\right]$$



Mescheder, Lars, Andreas Geiger, and Sebastian Nowozin. "Which training methods for GANs do actually converge?." *International conference on machine learning*. PMLR, 2018.

Motion synthesis as pose prediction

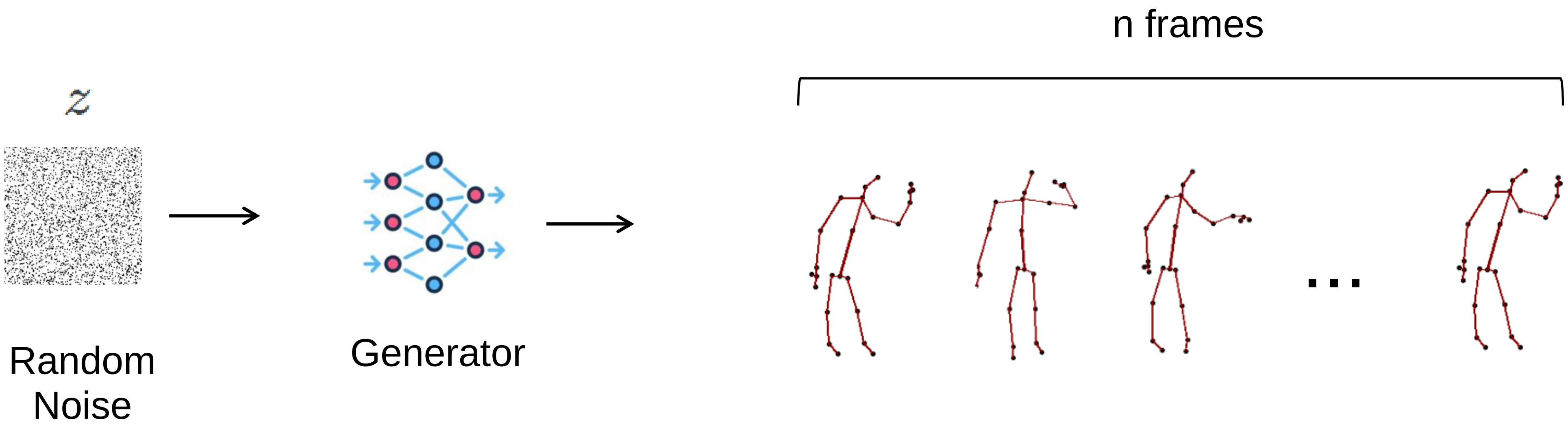


Random Noise

Generator

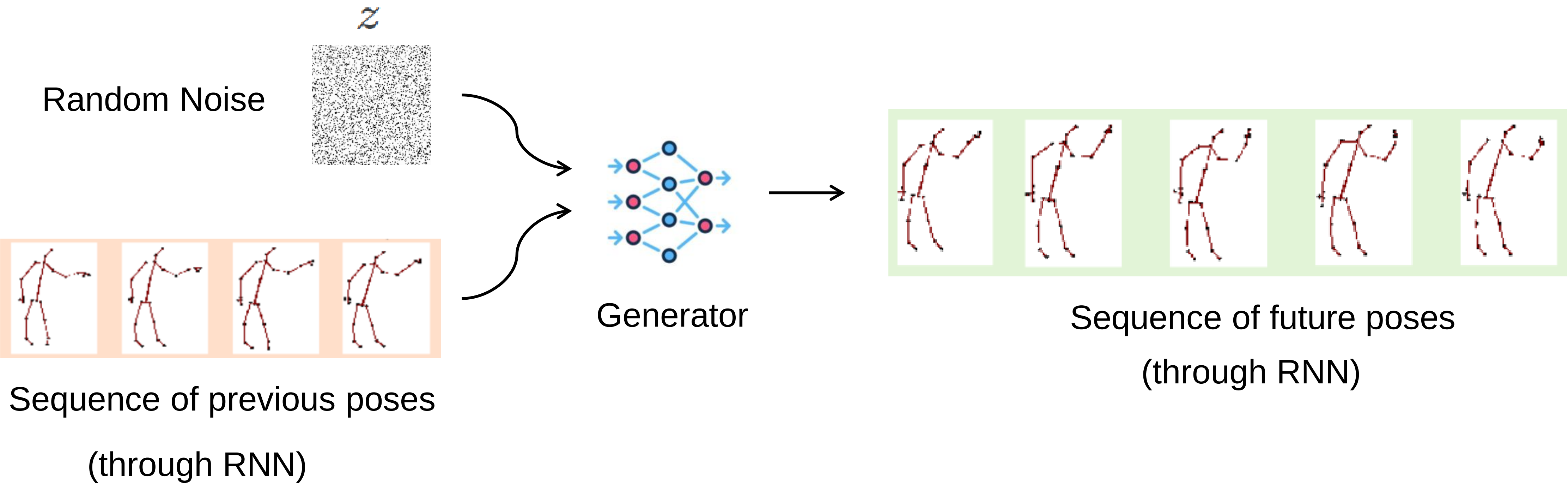
Generated Output (human pose)

Motion synthesis as pose prediction



But these poses are not temporally consistent!!

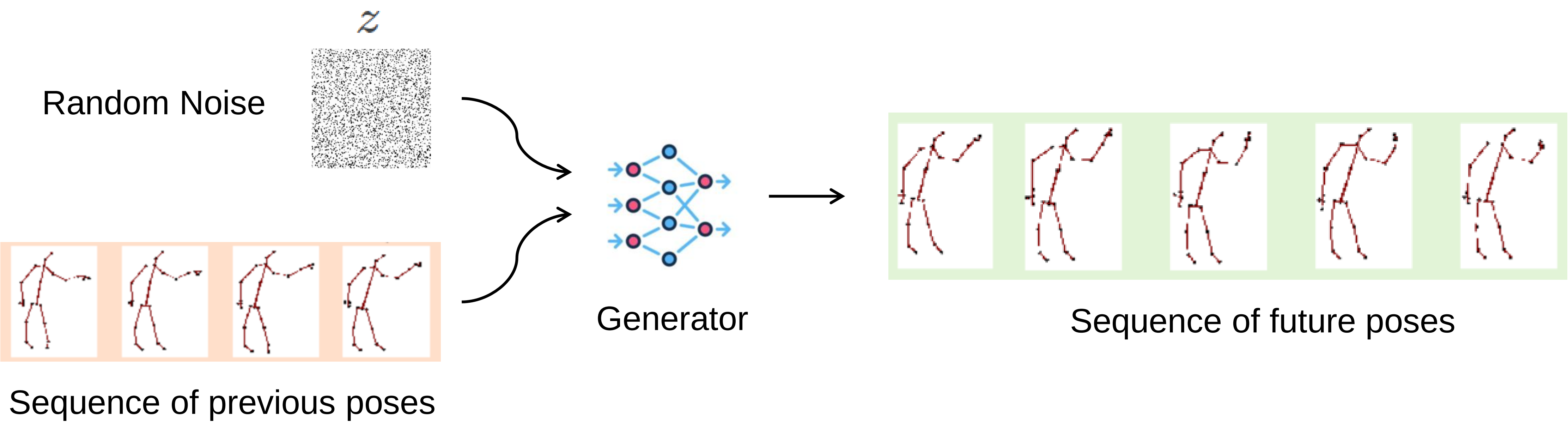
Motion synthesis as pose prediction



$G(z) \longrightarrow G(z|x)$ introducing a **prior**

Barsoum, Emad, John Kender, and Zicheng Liu. "Hp-gan: Probabilistic 3d human motion prediction via gan." *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018.

Motion synthesis as pose prediction



$$L_g = L_{adv} + \alpha L_{pg} + \beta L_b$$

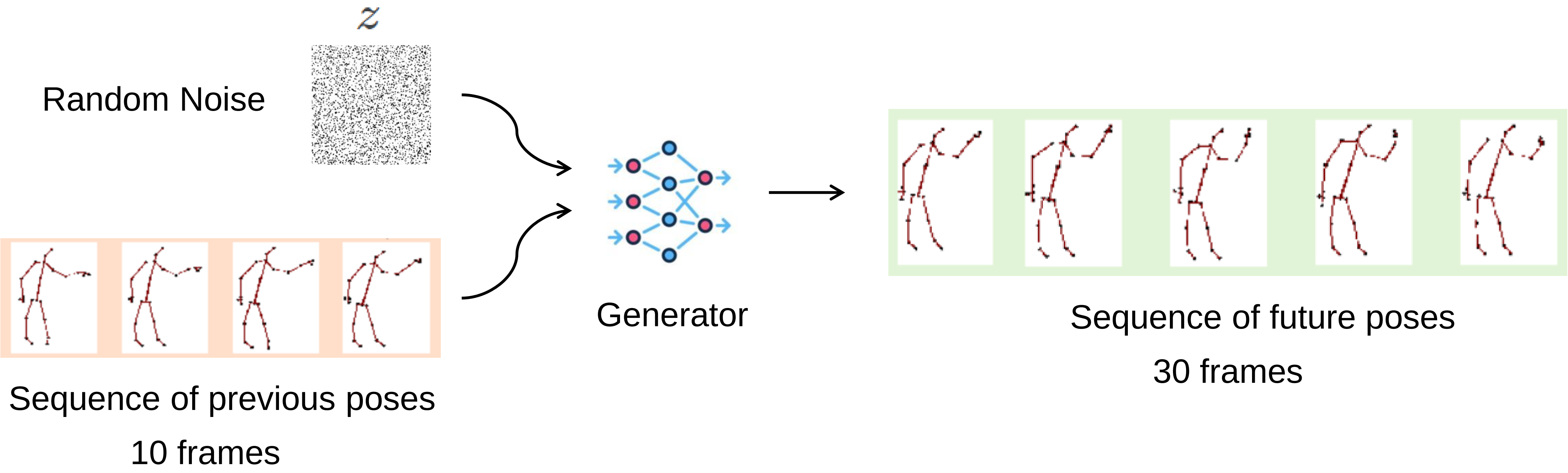
adversarial loss
pose gradient loss
bone length loss

$$L_c = L_{wgan} + \lambda L_{gp} + \alpha L_2$$

WGAN loss
gradient penalty loss
regularization loss

Barsoum, Emad, John Kender, and Zicheng Liu. "Hp-gan: Probabilistic 3d human motion prediction via gan." *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018.

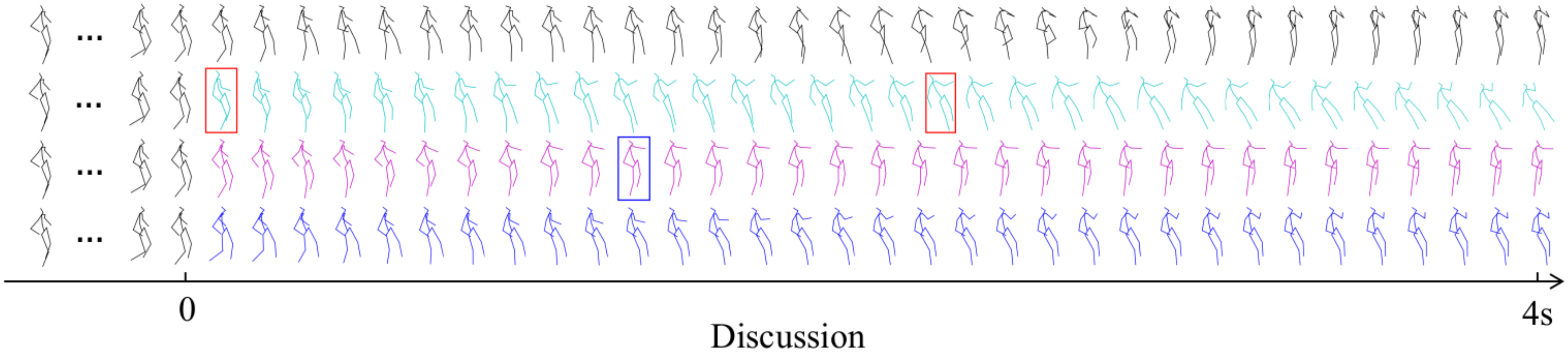
Motion synthesis as pose prediction



Probabilistic model: can get diverse outputs by sampling different z

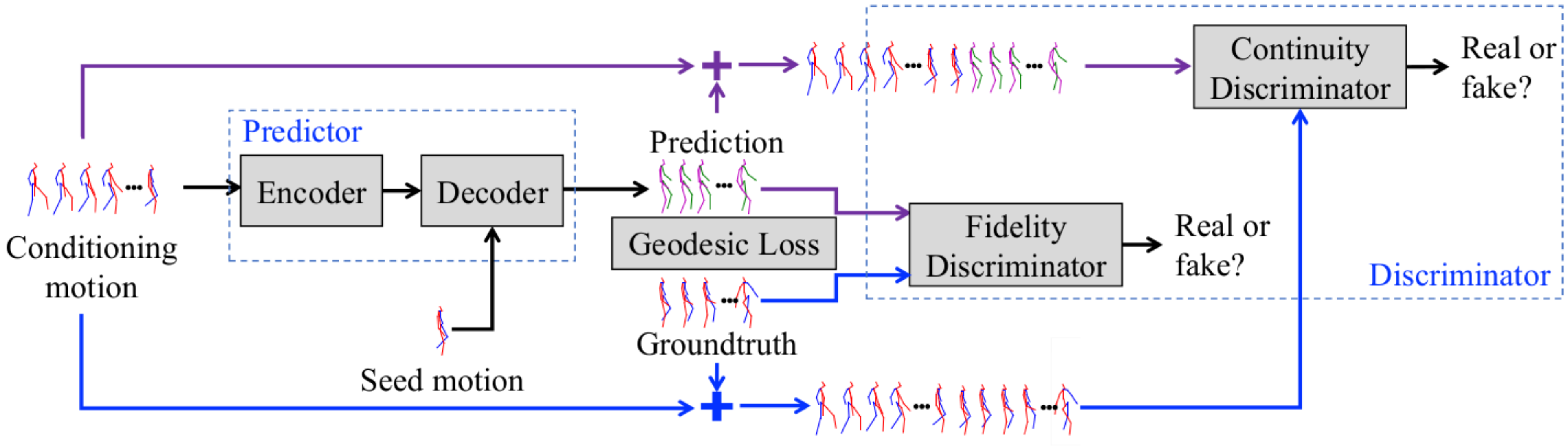
Barsoum, Emad, John Kender, and Zicheng Liu. "Hp-gan: Probabilistic 3d human motion prediction via gan." *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018.

Motion synthesis as pose prediction
Longer horizons



Gui, Liang-Yan, et al. "Adversarial geometry-aware human motion prediction." *Proceedings of the european conference on computer vision (ECCV)*. 2018.

Motion synthesis as pose prediction
Longer horizons



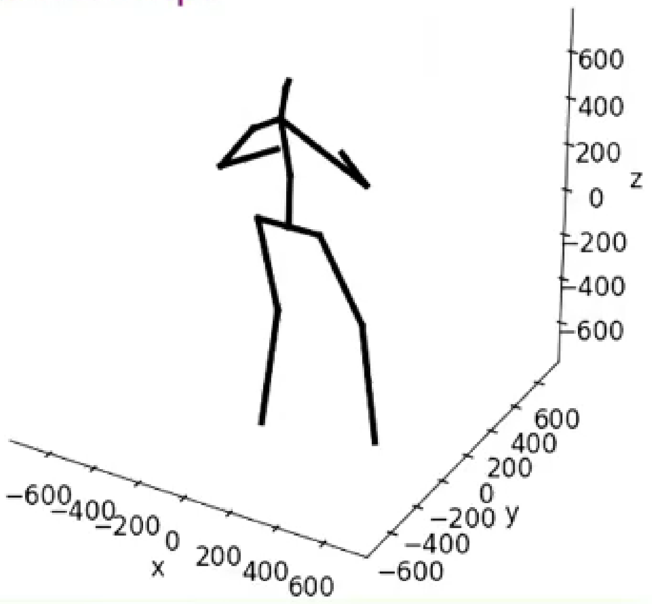
Geodesic distance in SO(3) instead of Euclidean distance
Dual discriminator

Gui, Liang-Yan, et al. "Adversarial geometry-aware human motion prediction." *Proceedings of the european conference on computer vision (ECCV)*. 2018.

Aperiodic Activity: Taking Photo

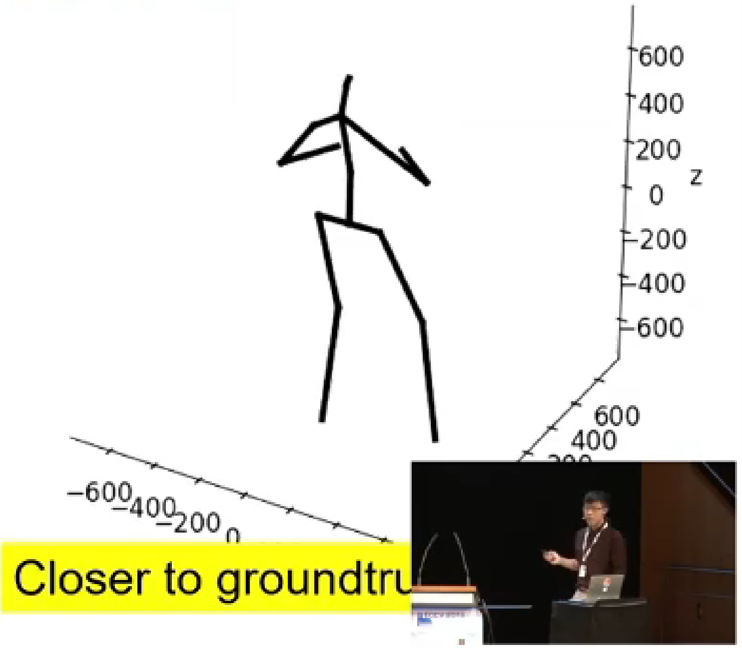
(in slow motion: 400ms displays in 10s)

Residual sup.

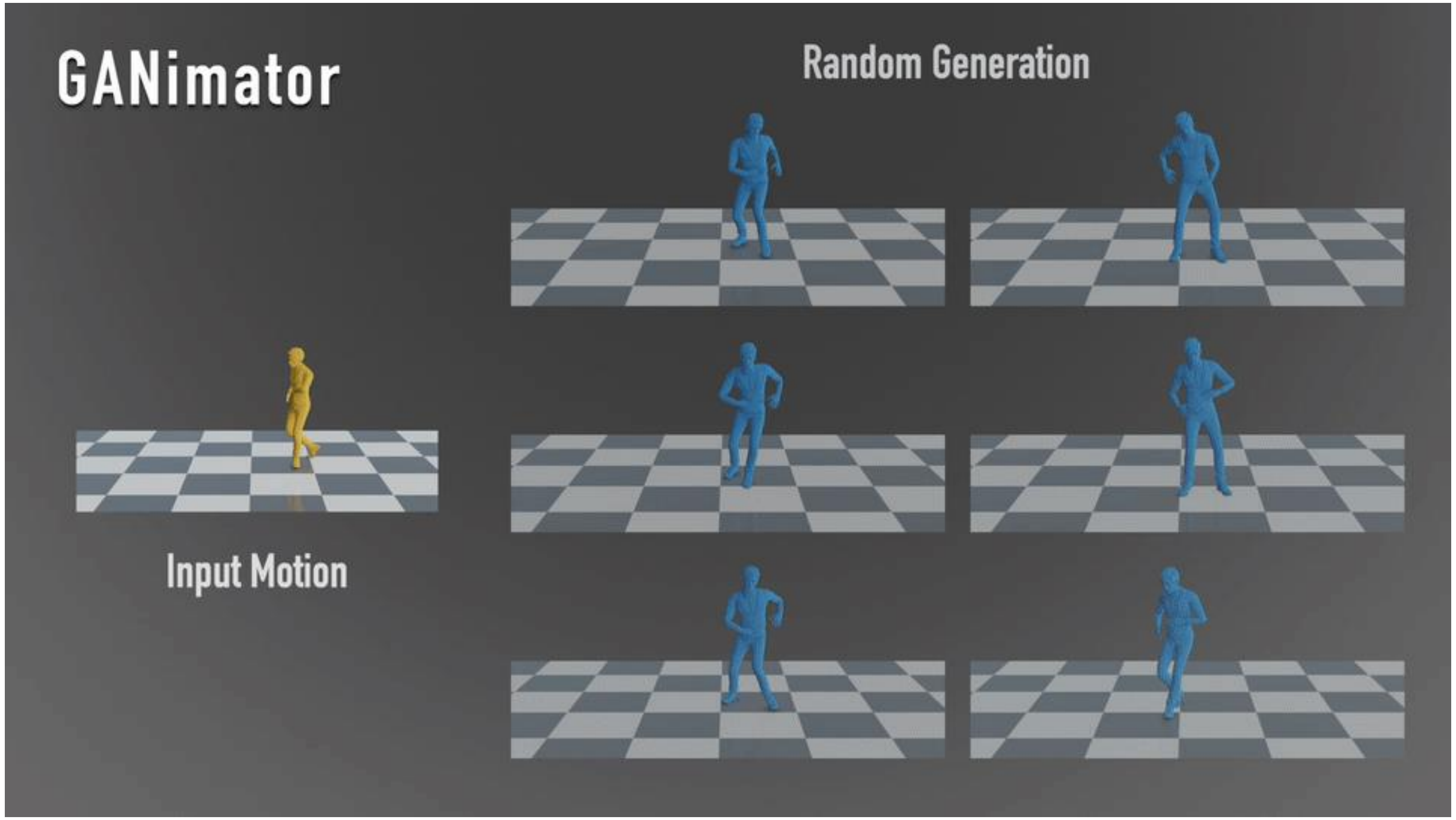


Martinez, Black, and Romero, 2017.

Our Model



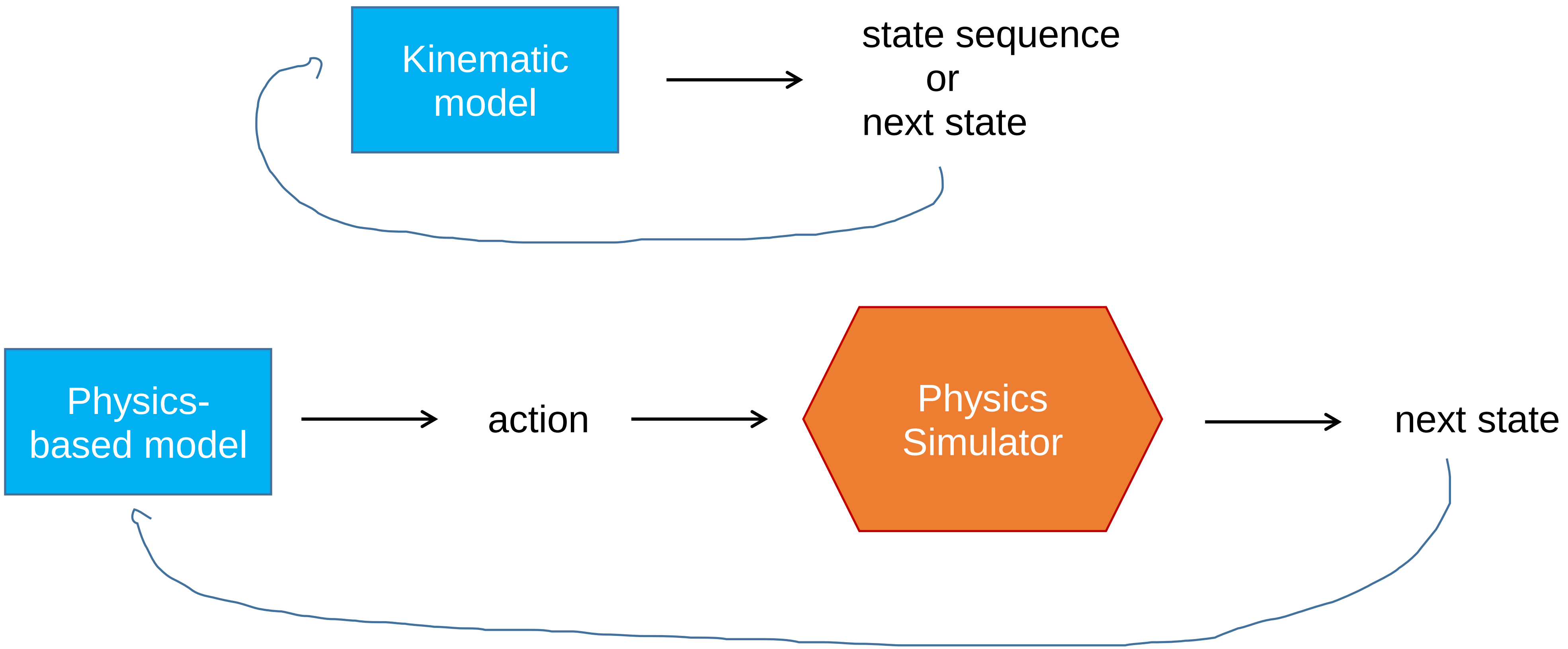
GANs can learn from low amount of data



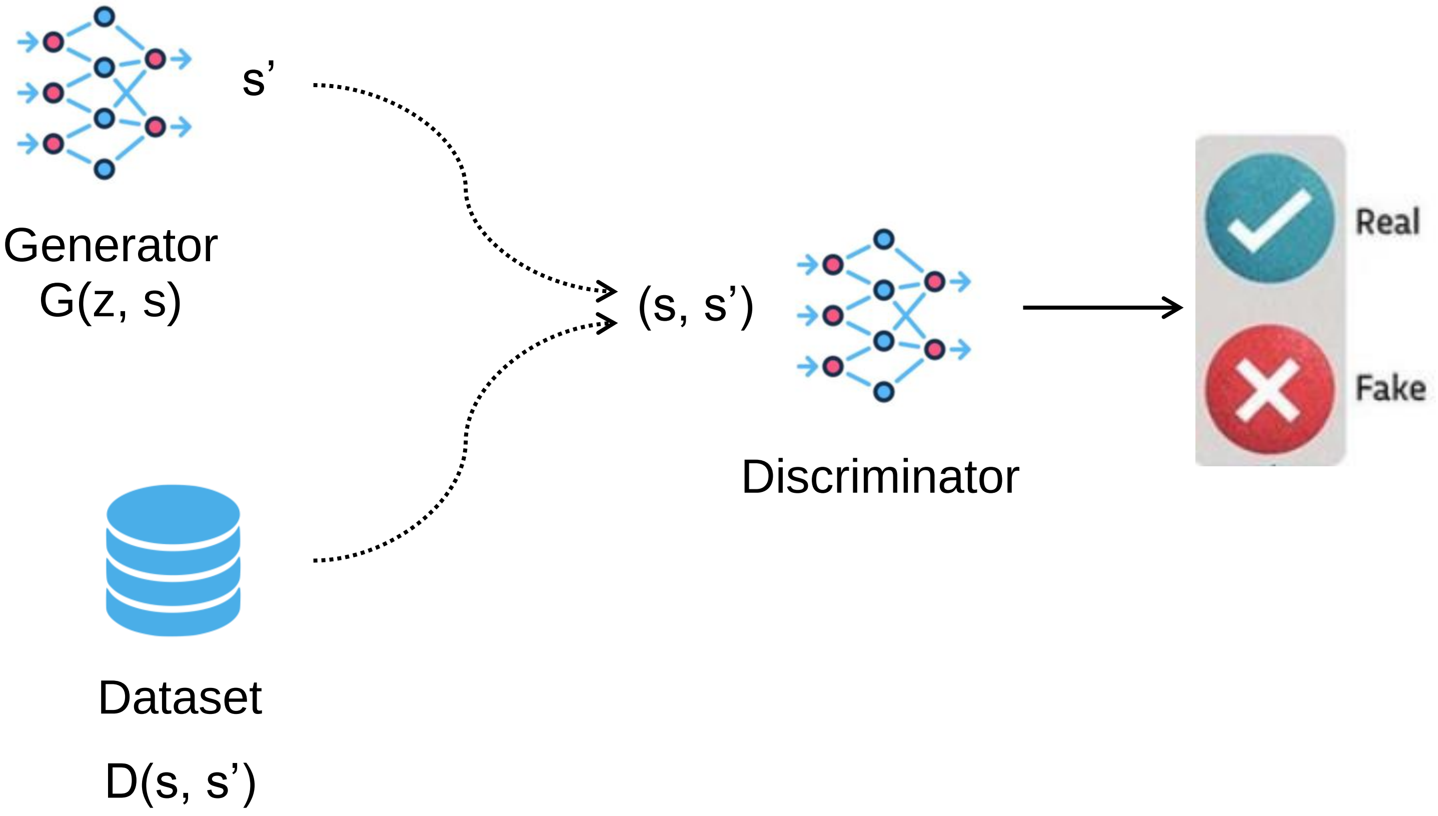
Li, Peizhuo, et al. "Ganimator: Neural motion synthesis from a single sequence." *ACM Transactions on Graphics (TOG)* 41.4 (2022): 1-12.

How about physics-based?

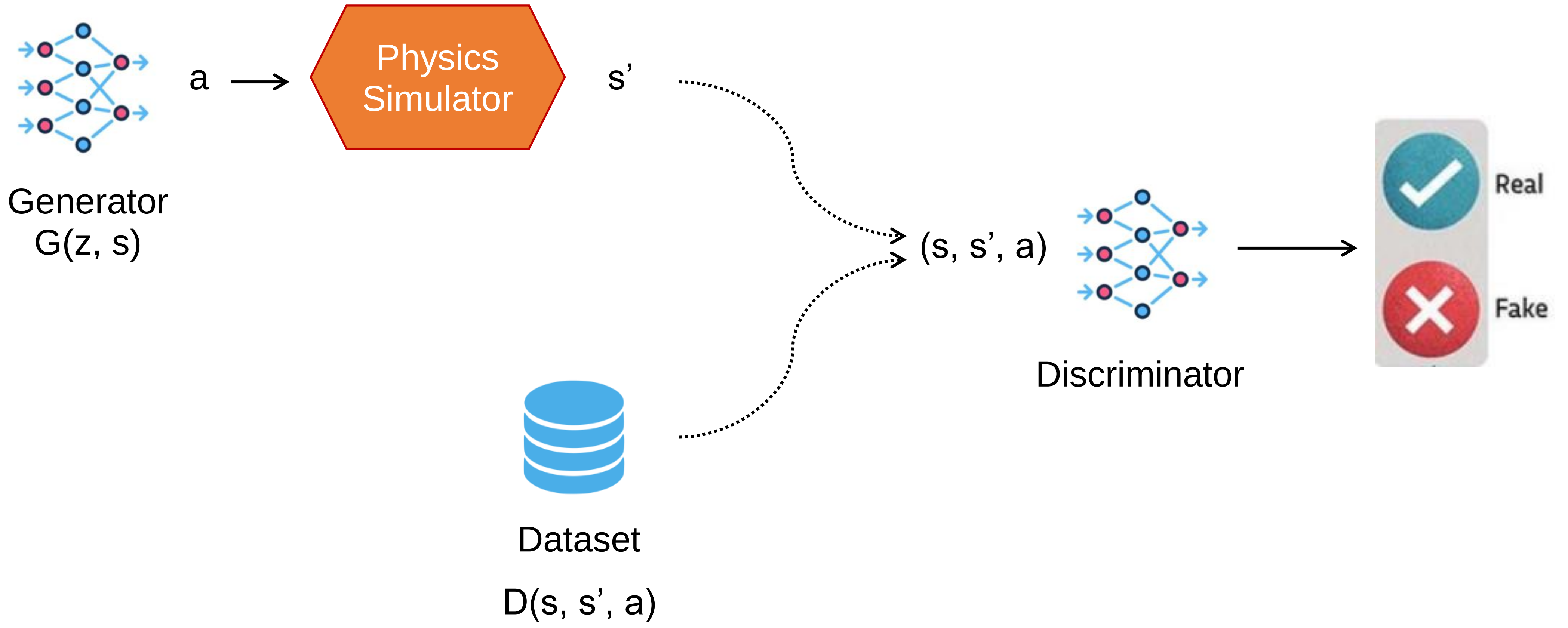
Need to predict actions



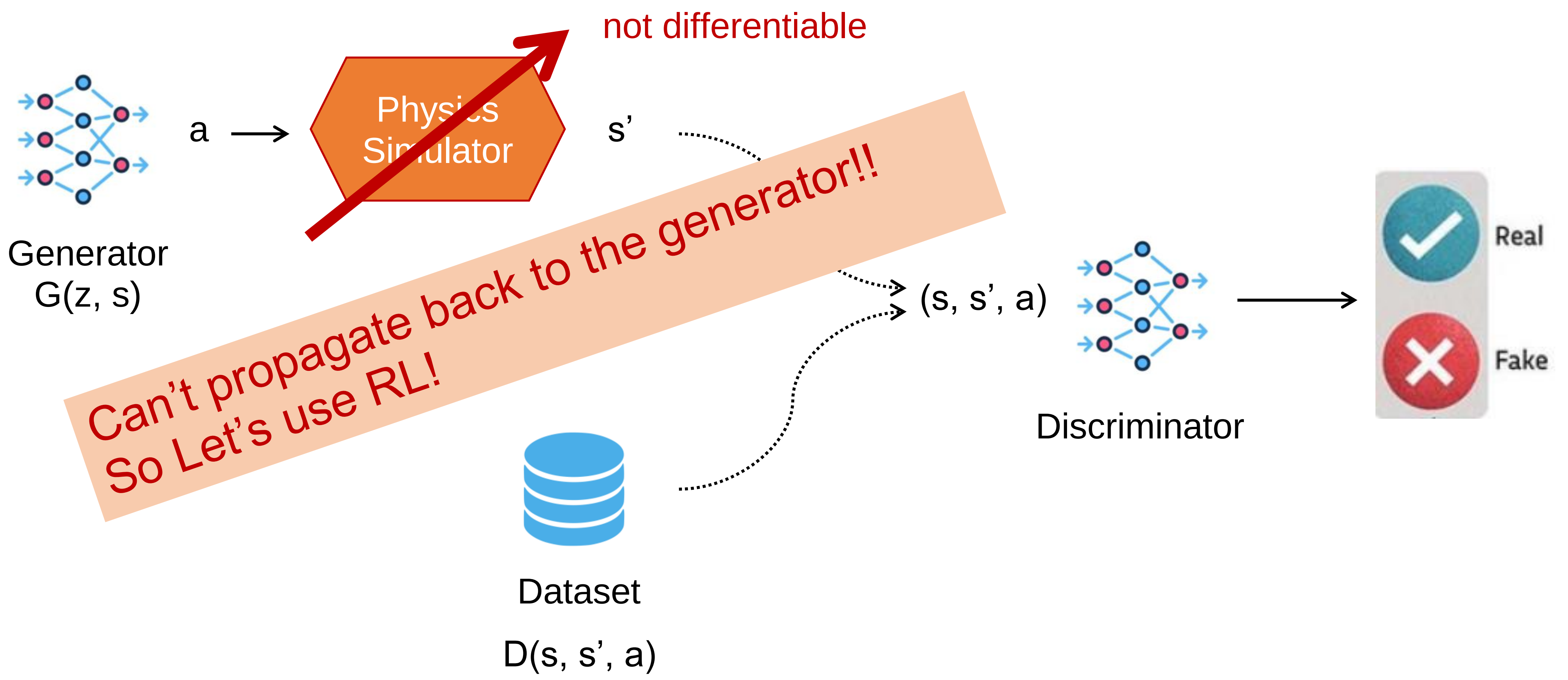
GAN (Kinematic)



GAN (Physics-based)

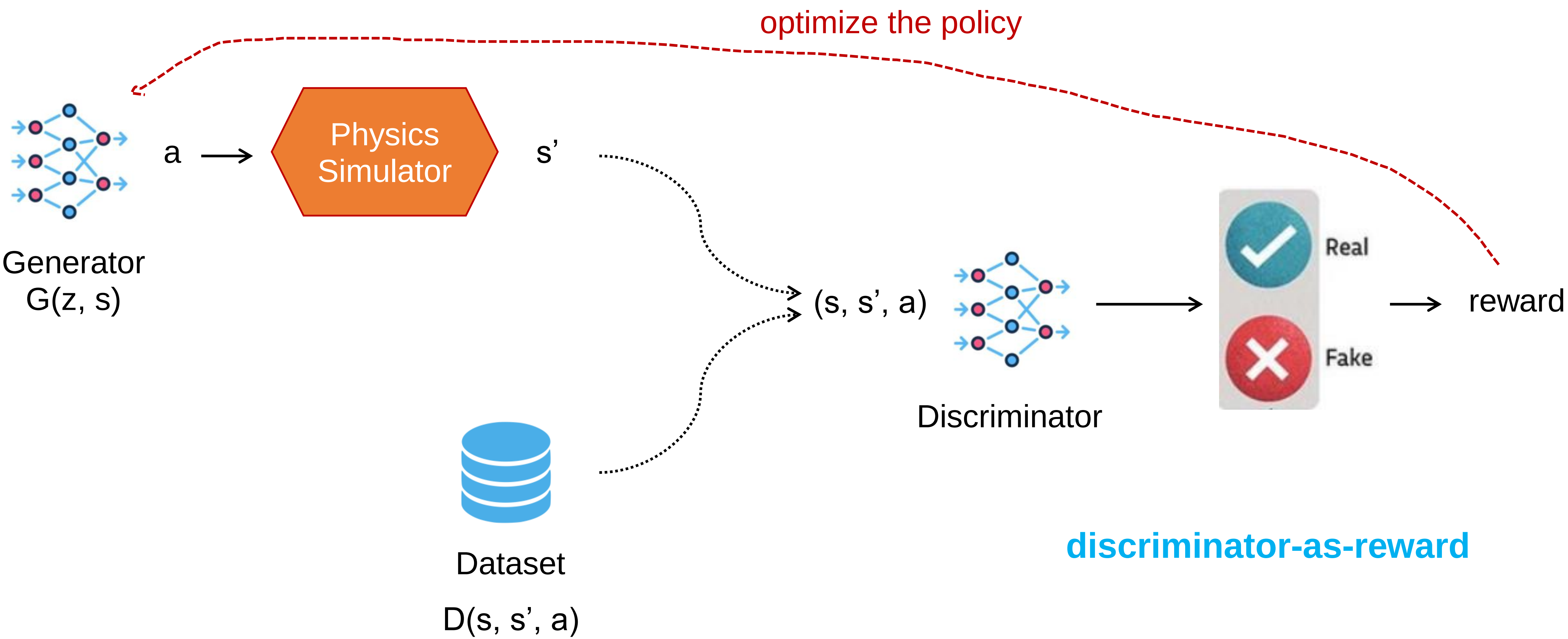


GAN (Physics-based)



Generative Adversarial Imitation Learning

Generative Adversarial Imitation Learning



discriminator-as-reward

Generative Adversarial Imitation Learning

$$\mathbb{E}_{\pi}[\log(D(s,a))] + \mathbb{E}_{\pi_E}[\log(1 - D(s,a))] - \lambda H(\pi)$$

policy expert policy

Algorithm 1 Generative adversarial imitation learning

1: **Input:** Expert trajectories $\tau_E \sim \pi_E$, initial policy and discriminator parameters θ_0, w_0
2: **for** $i = 0, 1, 2, \ldots$ **do**
3: Sample trajectories $\tau_i \sim \pi_{\theta_i}$
4: Update the discriminator parameters from w_i to w_{i+1} with the gradient

$$\hat{\mathbb{E}}_{\tau_i}[\nabla_w \log(D_w(s,a))] + \hat{\mathbb{E}}_{\tau_E}[\nabla_w \log(1 - D_w(s,a))] \quad (17)$$

5: Take a policy step from θ_i to θ_{i+1}, using the TRPO rule with cost function $\log(D_{w_{i+1}}(s,a))$. Specifically, take a KL-constrained natural gradient step with

$$\begin{aligned} &\hat{\mathbb{E}}_{\tau_i}[\nabla_\theta \log \pi_\theta(a|s) Q(s,a)] - \lambda \nabla_\theta H(\pi_\theta), \\ &\text{where } Q(\bar{s},\bar{a}) = \hat{\mathbb{E}}_{\tau_i}[\log(D_{w_{i+1}}(s,a)) \mid s_0 = \bar{s}, a_0 = \bar{a}] \end{aligned} \quad (18)$$

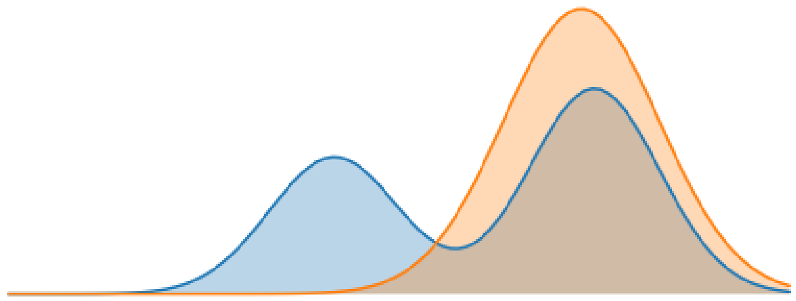
6: **end for**

Ho, Jonathan, and Stefano Ermon. "Generative adversarial imitation learning." *Advances in neural information processing systems* 29 (2016).

Generative Adversarial Imitation Learning

$r_t = -\log(1 - D((s_t, a_t)))$ or $r_t = \log(D((s_t, a_t)))$ **discriminator-as-reward**

minimizes Jensen-Shannon divergence between $d^{\mathcal{M}}(\mathrm{s}, \mathrm{a})$ and $d^{\pi}(\mathrm{s}, \mathrm{a})$



Ho, Jonathan, and Stefano Ermon. "Generative adversarial imitation learning." *Advances in neural information processing systems* 29 (2016).

But for motion, we have no action in the demonstration!!

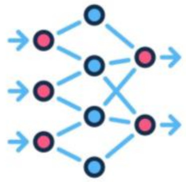
But for motion, we have no action in the demonstration!!

$$\underset{D}{\arg\min}\ -\mathbb{E}_{d^{\mathcal{M}}(\mathrm{s},\mathrm{a})}\left[\log\left(D(\mathrm{s},\mathrm{a})\right)\right] - \mathbb{E}_{d^{\pi}(\mathrm{s},\mathrm{a})}\left[\log\left(1 - D(\mathrm{s},\mathrm{a})\right)\right].$$

$$\underset{D}{\arg\min} - \mathbb{E}_{d^{\mathcal{M}}(\mathrm{s},\mathrm{s}')}\left[\log\left(D(\mathrm{s},\mathrm{s}')\right)\right] - \mathbb{E}_{d^{\pi}(\mathrm{s},\mathrm{s}')}\left[\log\left(1 - D(\mathrm{s},\mathrm{s}')\right)\right].$$

~~(s, a)~~

(s, s')



Discriminator

Faraz Torabi, Garrett Warnell, and Peter Stone. 2018. Generative Adversarial Imitation from Observation. CoRR abs/1807.06158 (2018). arXiv:1807.06158 http://arxiv.org/abs/1807.06158

Adversarial Motion Priors

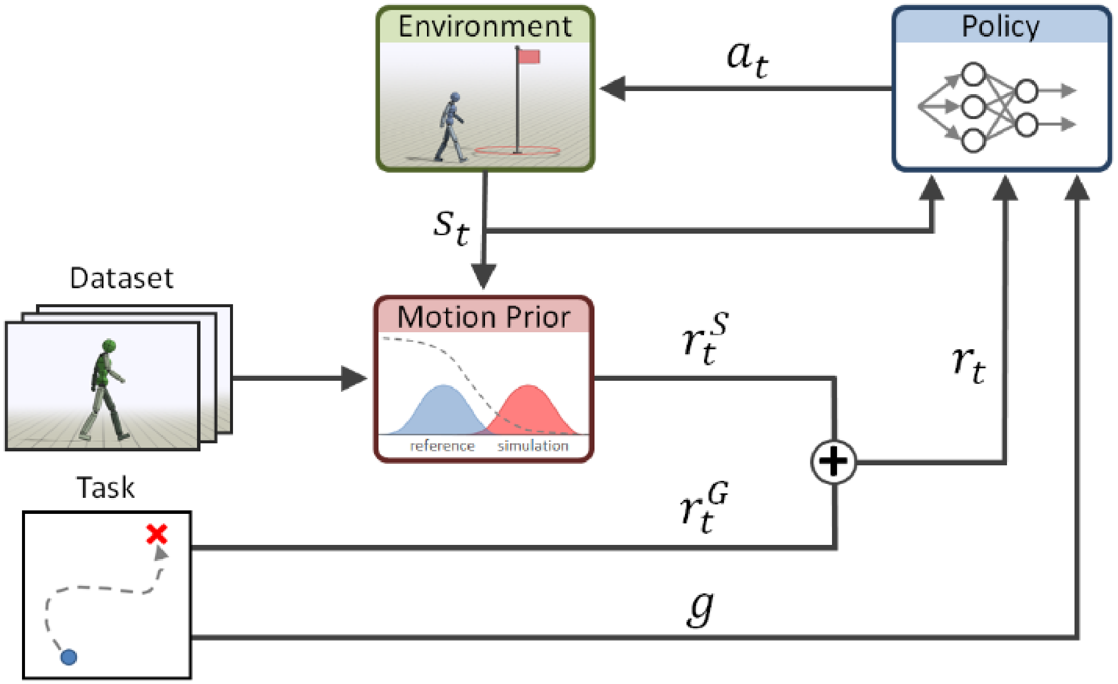
$$\arg\min_{D} \; \mathbb{E}_{d^{\mathcal{M}}(s,s')}\left[\left(D(s,s')-1\right)^2\right] + \mathbb{E}_{d^{\pi}(s,s')}\left[\left(D(s,s')+1\right)^2\right]$$

$$r(s_t, s_{t+1}) = \max\left[0,\;\; 1 - 0.25\left(D(s_t, s_{t+1}) - 1\right)^2\right].$$

Style reward

$$r(s_t, a_t, s_{t+1}, g) = w^G r^G(s_t, a_t, s_t, g) + w^S r^S(s_t, s_{t+1}).$$

Goal (Task) reward

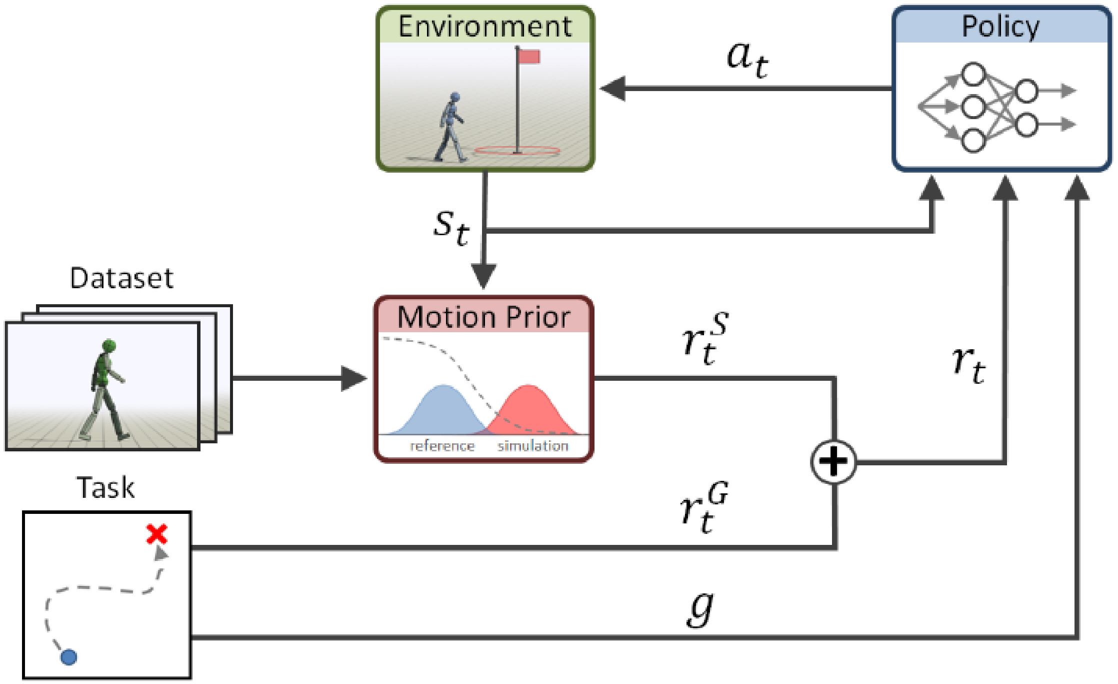


Peng, Xue Bin, et al. "Amp: Adversarial motion priors for stylized physics-based character control." *ACM Transactions on Graphics (ToG)* 40.4 (2021): 1-20.

Adversarial Motion Priors

gradient penalty

$$\arg\min_{D} \quad \mathbb{E}_{d^{\mathcal{M}}(\mathrm{s},\mathrm{s}')}\left[\left(D(\Phi(\mathrm{s}),\Phi(\mathrm{s}'))-1\right)^2\right] + \mathbb{E}_{d^{\pi}(\mathrm{s},\mathrm{s}')}\left[\left(D\left(\Phi(\mathrm{s}),\Phi(\mathrm{s}')\right)+1\right)^2\right] + \frac{w^{\mathrm{gp}}}{2}\,\mathbb{E}_{d^{\mathcal{M}}(\mathrm{s},\mathrm{s}')}\left[\left\|\nabla_{\phi}D(\phi)\big|_{\phi=(\Phi(\mathrm{s}),\Phi(\mathrm{s}'))}\right\|^2\right],$$



Peng, Xue Bin, et al. "Amp: Adversarial motion priors for stylized physics-based character control." *ACM Transactions on Graphics (ToG)* 40.4 (2021): 1-20.

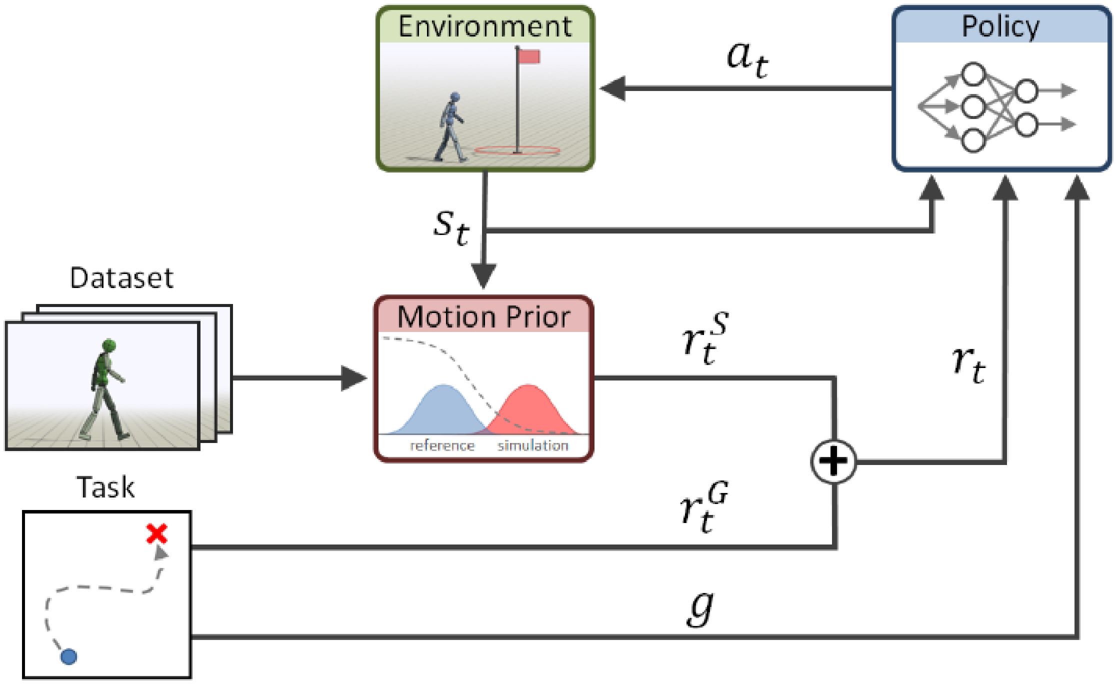
Adversarial Motion Priors

ALGORITHM 1: Training with AMP

while not done **do**
  for trajectory $i = 1, ..., m$ **do**
    $\tau^i \leftarrow \{(s_t, a_t, r_t^G)_{t=0}^{T-1}, s_T^G, g\}$ collect trajectory with π
    for time step $t = 0, ..., T-1$ **do**
      $d_t \leftarrow D(\Phi(s_t), \Phi(s_{t+1}))$
      $r_t^S \leftarrow$ calculate style reward according to Equation 7 using d_t
      $r_t \leftarrow w^G r_t^G + w^S r_t^S$
      record r_t in τ^i
    end for
    store τ^i in $\mathcal{B}$
  end for

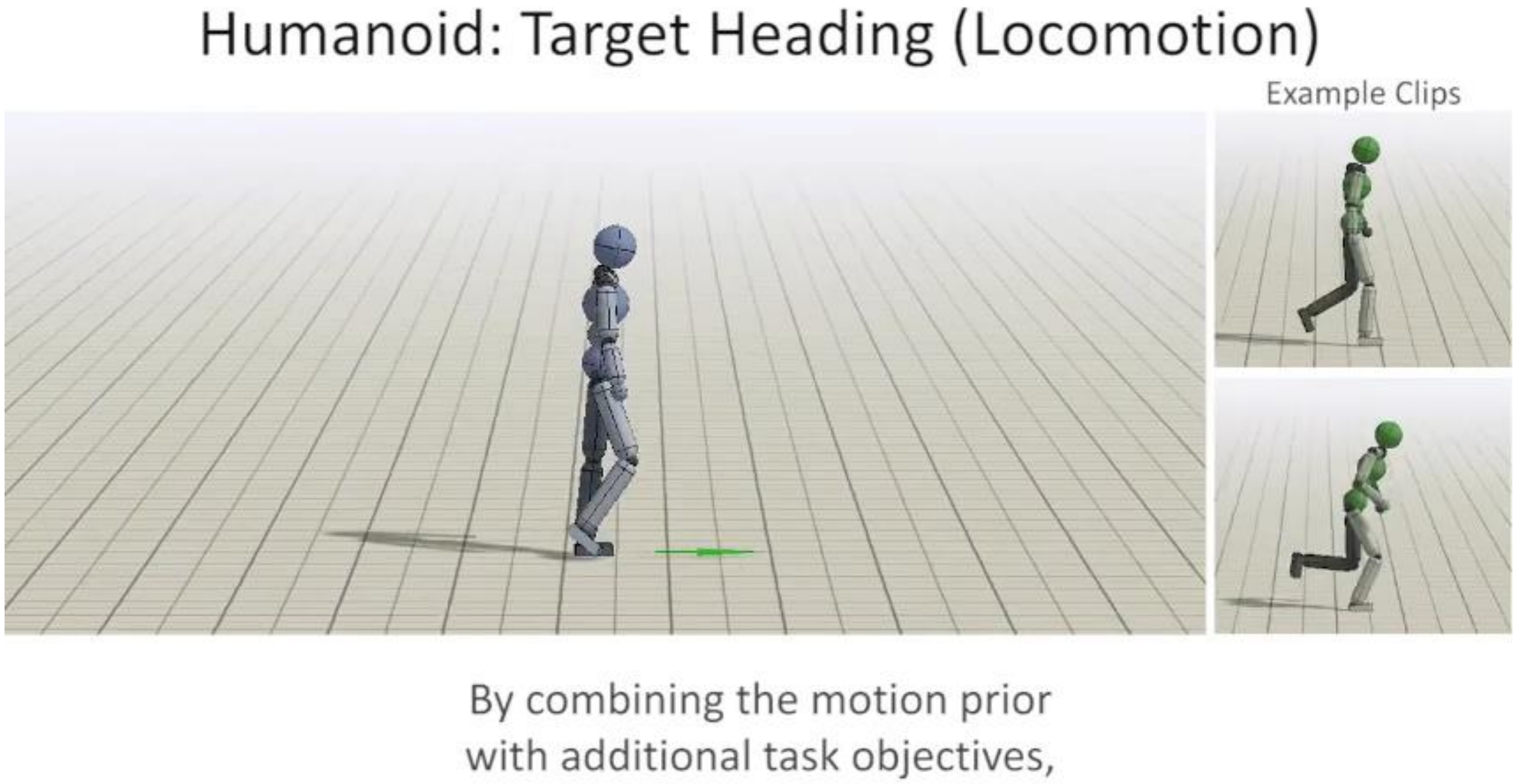
  for update step $= 1, ..., n$ **do**
    $b^{\mathcal{M}} \leftarrow$ sample batch of K transitions $\{(s_j, s'_j)\}_{j=1}^K$ from $\mathcal{M}$
    $b^{\pi} \leftarrow$ sample batch of K transitions $\{(s_j, s'_j)\}_{j=1}^K$ from $\mathcal{B}$
    update D according to Equation 8 using $b^{\mathcal{M}}$ and b^{π}
  end for

  update V and π using data from trajectories $\{\tau^i\}_{i=1}^m$
end while



Peng, Xue Bin, et al. "Amp: Adversarial motion priors for stylized physics-based character control." *ACM Transactions on Graphics (ToG)* 40.4 (2021): 1-20.

Adversarial Motion Priors



Peng, Xue Bin, et al. "Amp: Adversarial motion priors for stylized physics-based character control." *ACM Transactions on Graphics (ToG)* 40.4 (2021): 1-20.

Adversarial Motion Priors

Advantages:

- Can generate novel motions
- Can learn from low amount of data
- Flexible
- Learning is easier

$$||s_t^{\mathcal{M}} - s_t^{\pi}||$$

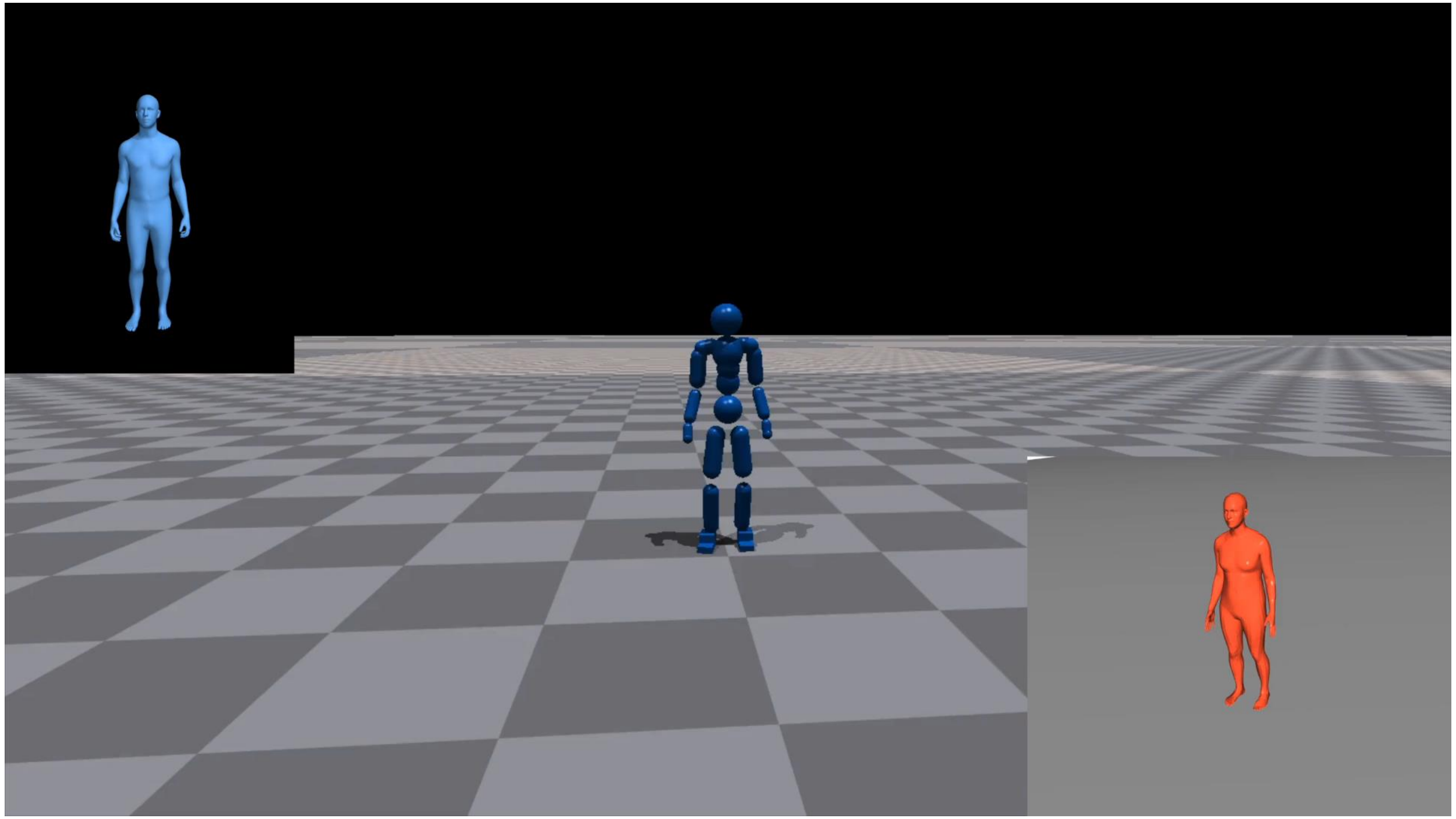
Motion Tracking

$$D_{JS}\{p(s, s')^{\mathcal{M}}, p(s, s')^{\pi}\}$$

AMP

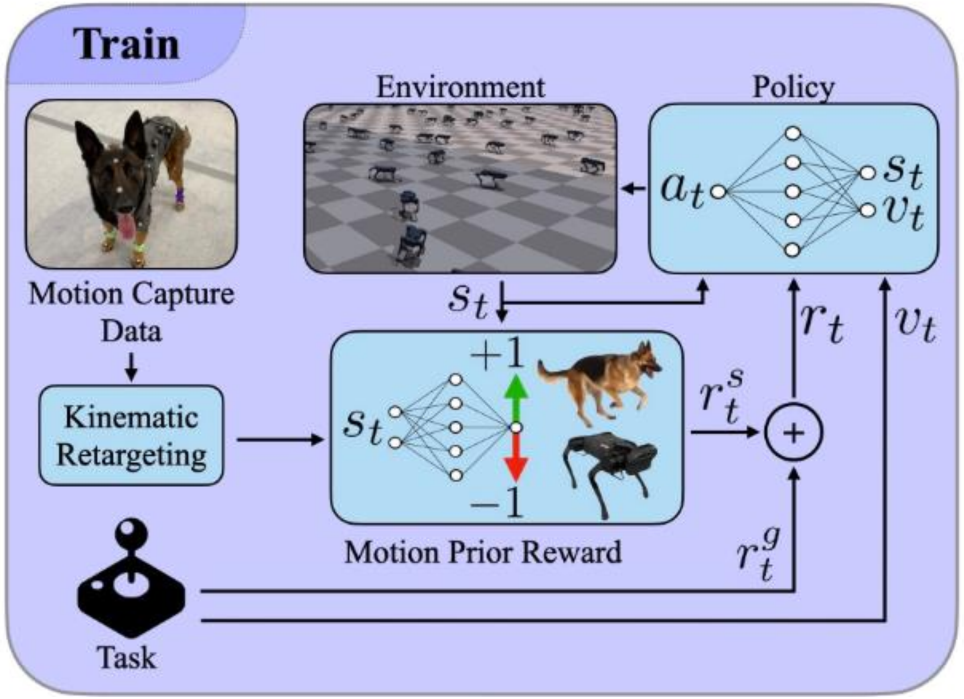
Peng, Xue Bin, et al. "Amp: Adversarial motion priors for stylized physics-based character control." *ACM Transactions on Graphics (ToG)* 40.4 (2021): 1-20.

AMP follow-ups



Luo, Zhengyi, et al. "Perpetual humanoid control for real-time simulated avatars." *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023.

AMP for robots

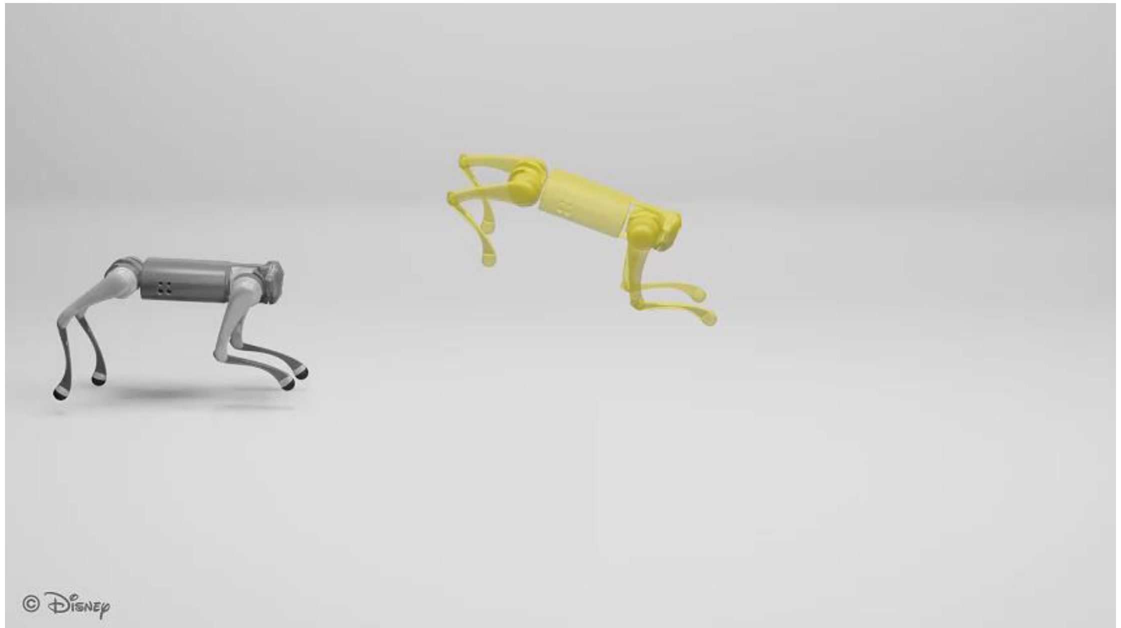


Escontrela, Alejandro, et al. "Adversarial motion priors make good substitutes for complex reward functions." *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022.

AMP for robots

Multiple skills

More control

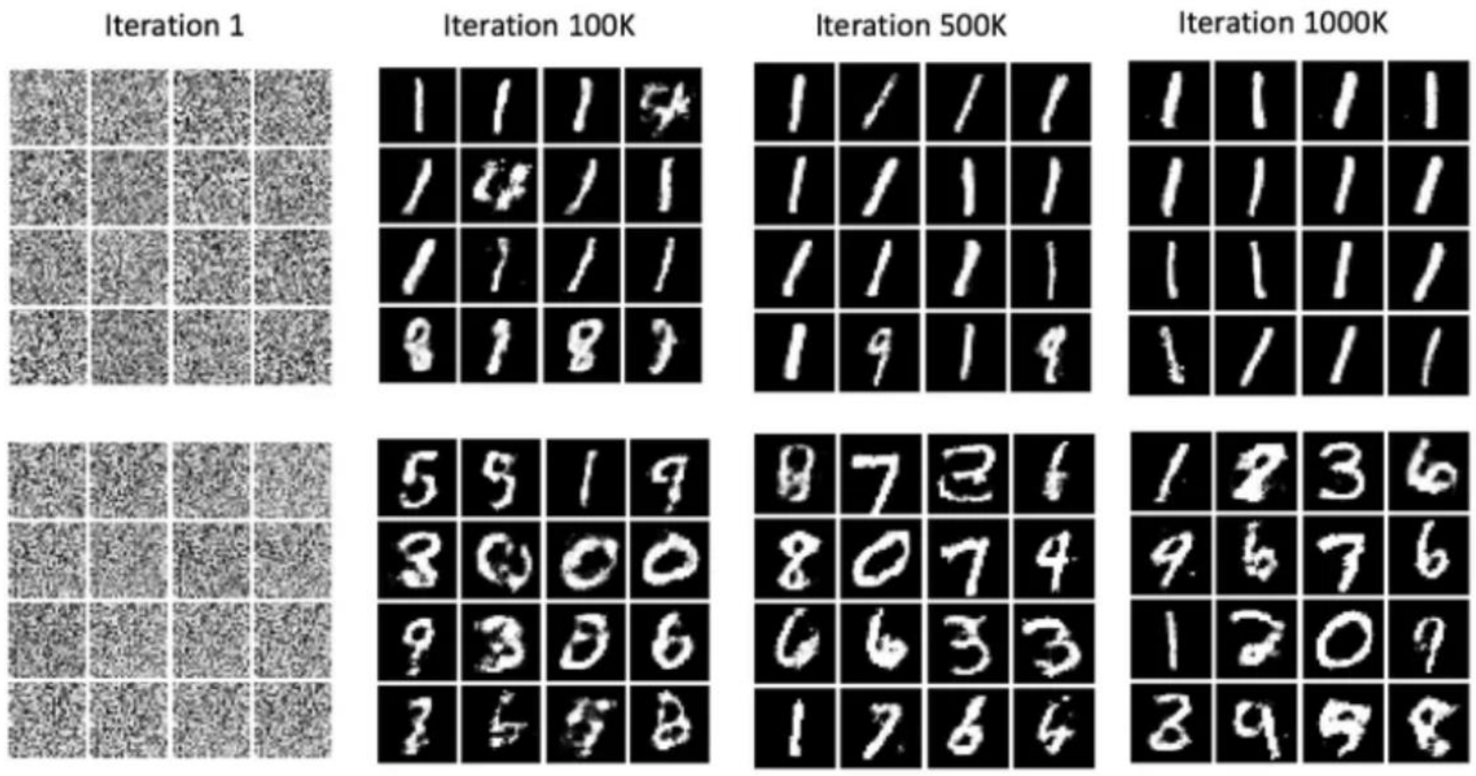


Vollenweider, Eric, et al. "Advanced skills through multiple adversarial motion priors in reinforcement learning." *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023.

Zargarbashi, Fatemeh, et al. “RobotKeyframing: Learning Locomotion with High-Level Objectives via Mixture of Dense and Sparse Rewards” in Proceedings of The 8th Conference on Robot Learning (CoRL 2024) 270 (PMLR, 2025),916.

Problem with GAN and GAIL

Mode collapse

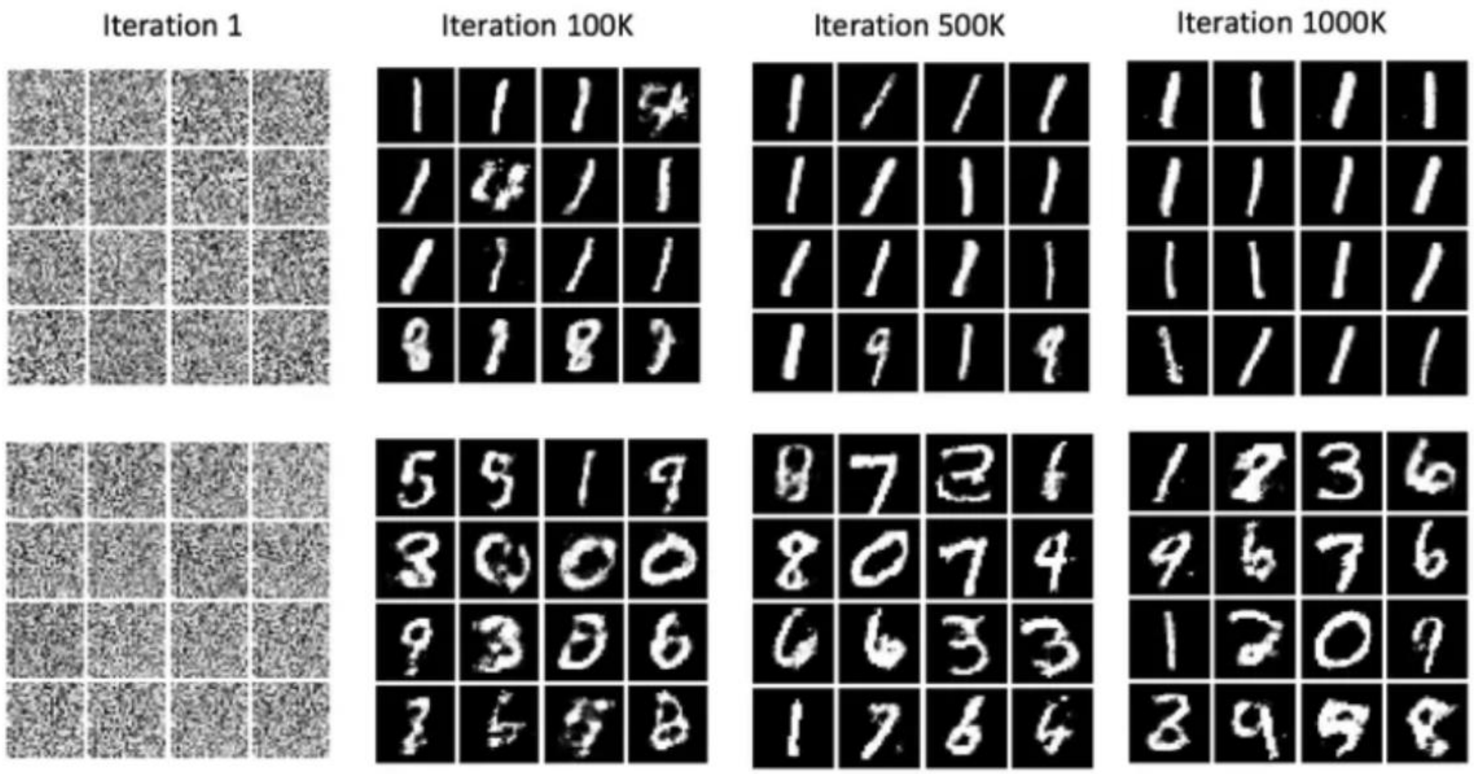


https://medium.com/@miraytopal/what-is-mode-collapse-in-gans-d3428a7bd9b8

Problem with GAN and GAIL

Mode collapse

As long as generator's output is similar to part of the data, the discriminator gives good score!



https://medium.com/@miraytopal/what-is-mode-collapse-in-gans-d3428a7bd9b8

How to avoid mode collapse?

Introduce a random/probabilistic prior

Adversarial Skill Embeddings

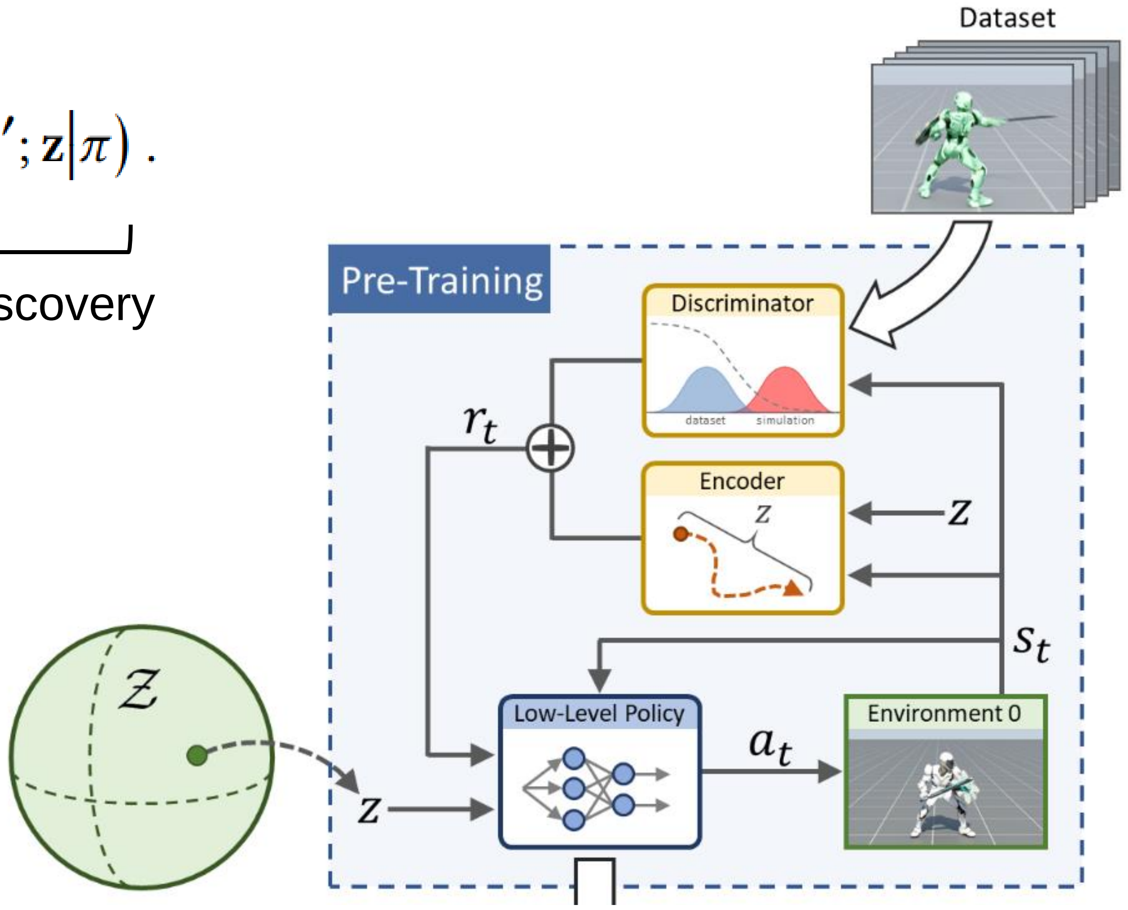
$$\max_{\pi} \quad -D_{\mathrm{JS}}\left(d^{\pi}(\mathbf{s},\mathbf{s}') \middle\| d^{\mathcal{M}}(\mathbf{s},\mathbf{s}')\right) + \beta\, I\left(\mathbf{s},\mathbf{s}';\mathbf{z}|\pi\right).$$

motion imitation — skill discovery

mutual information

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

$$I(\mathbf{s},\mathbf{s}';\mathbf{z}|\pi) = \mathcal{H}(\mathbf{s},\mathbf{s}'|\pi) - \mathcal{H}(\mathbf{s},\mathbf{s}'|\mathbf{z},\pi).$$



Peng, Xue Bin, et al. "Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters." *ACM Transactions On Graphics (TOG)* 41.4 (2022): 1-17.

Adversarial Skill Embeddings

$$I(\mathbf{s}, \mathbf{s}'; \mathbf{z}|\pi) = \mathcal{H}(\mathbf{s}, \mathbf{s}'|\pi) - \mathcal{H}(\mathbf{s}, \mathbf{s}'|\mathbf{z}, \pi).$$

but this is intractable to compute

$$\begin{aligned} I(\mathbf{s}, \mathbf{s}'; \mathbf{z}|\pi) &= I(\mathbf{z}; \mathbf{s}, \mathbf{s}'|\pi) \\ &= \mathcal{H}(\mathbf{z}) - \mathcal{H}(\mathbf{z}|\mathbf{s}, \mathbf{s}', \pi). \end{aligned}$$

constant

variational approximation $q(\mathbf{z}|\mathbf{s}, \mathbf{s}')$ encoder

$$\arg\max_{\pi} \quad \mathbb{E}_{p(\mathbf{z})}\mathbb{E}_{p(\tau|\pi,\mathbf{z})}\left[\sum_{t=0}^{T-1} \gamma^t \left(-\log\left(1 - D\left(\mathbf{s}_t, \mathbf{s}_{t+1}\right)\right) + \beta \log q\left(\mathbf{z}|\mathbf{s}_t, \mathbf{s}_{t+1}\right) \right)\right].$$

$$r_t = -\log\left(1 - D(\mathbf{s}_t, \mathbf{s}_{t+1})\right) + \beta \log q\left(\mathbf{z}_t|\mathbf{s}_t, \mathbf{s}_{t+1}\right).$$

Peng, Xue Bin, et al. "Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters." *ACM Transactions On Graphics (TOG)* 41.4 (2022): 1-17.

Adversarial Skill Embeddings

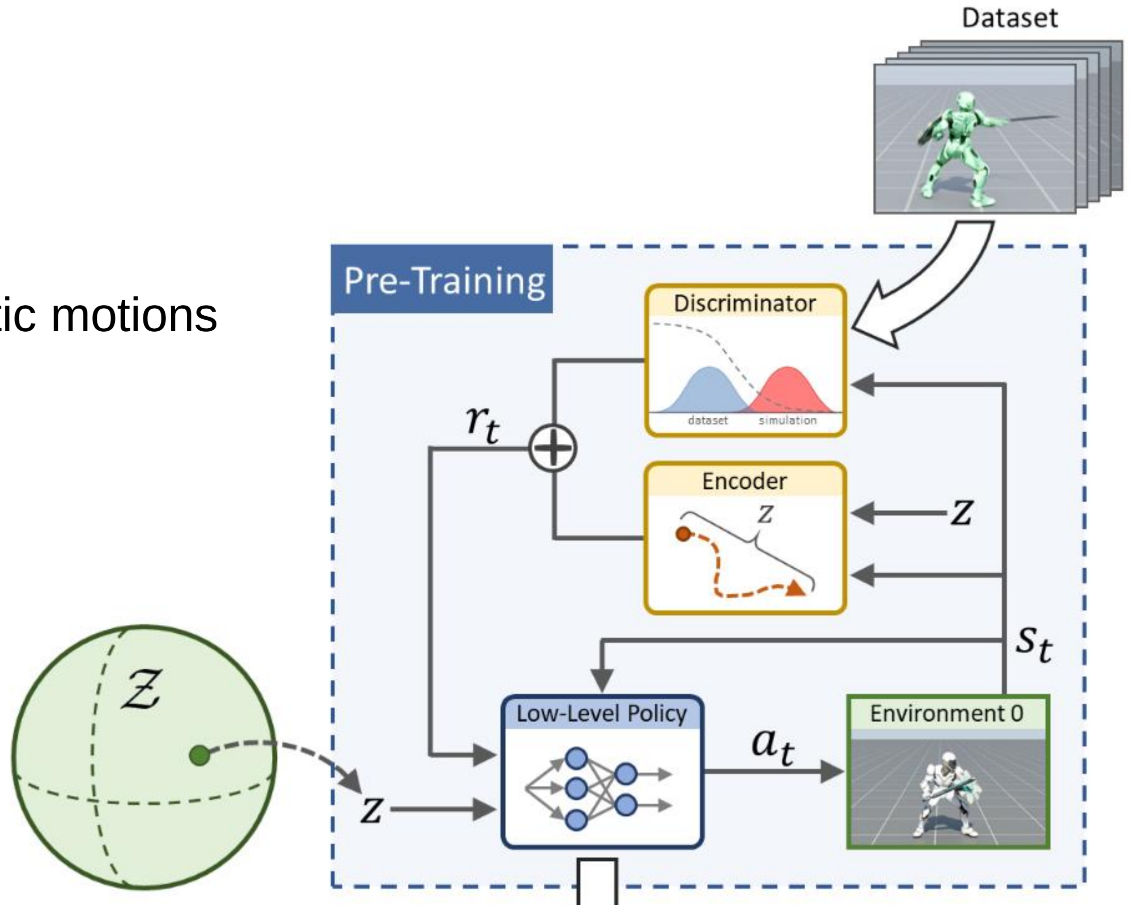
What prior distribution?

$$p(\mathbf{z}) = \mathcal{N}(0, I)$$

Gaussian distribution is unbounded ⟶ unrealistic motions

a uniform distribution on the surface of the sphere

$$\bar{\mathbf{z}} \sim \mathcal{N}(0, I), \qquad \mathbf{z} = \bar{\mathbf{z}} / ||\bar{\mathbf{z}}||.$$



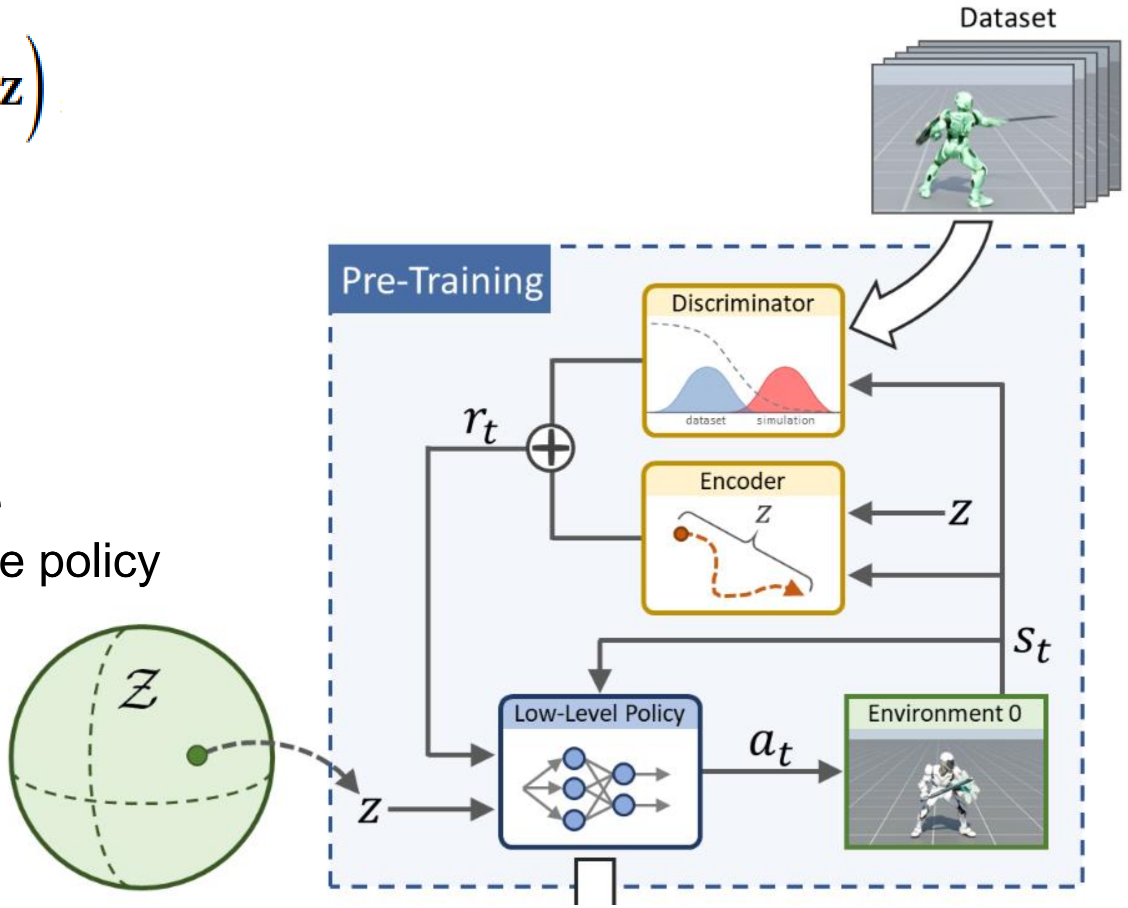
Peng, Xue Bin, et al. "Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters." *ACM Transactions On Graphics (TOG)* 41.4 (2022): 1-17.

Adversarial Skill Embeddings

skill encoder $q(\mathbf{z}|\mathbf{s}, \mathbf{s}') = \frac{1}{Z} \exp\left(\kappa \, \mu_q(\mathbf{s}, \mathbf{s}')^T \mathbf{z}\right)$

$$\max_q \quad \mathbb{E}_{p(\mathbf{z})} \mathbb{E}_{d^\pi(\mathbf{s}, \mathbf{s}'|\mathbf{z})} \left[\kappa \, \mu_q(\mathbf{s}, \mathbf{s}')^T z\right]$$

the encoder can then be trained by maximizing the log-likelihood of samples (z, s, s′) collected from the policy

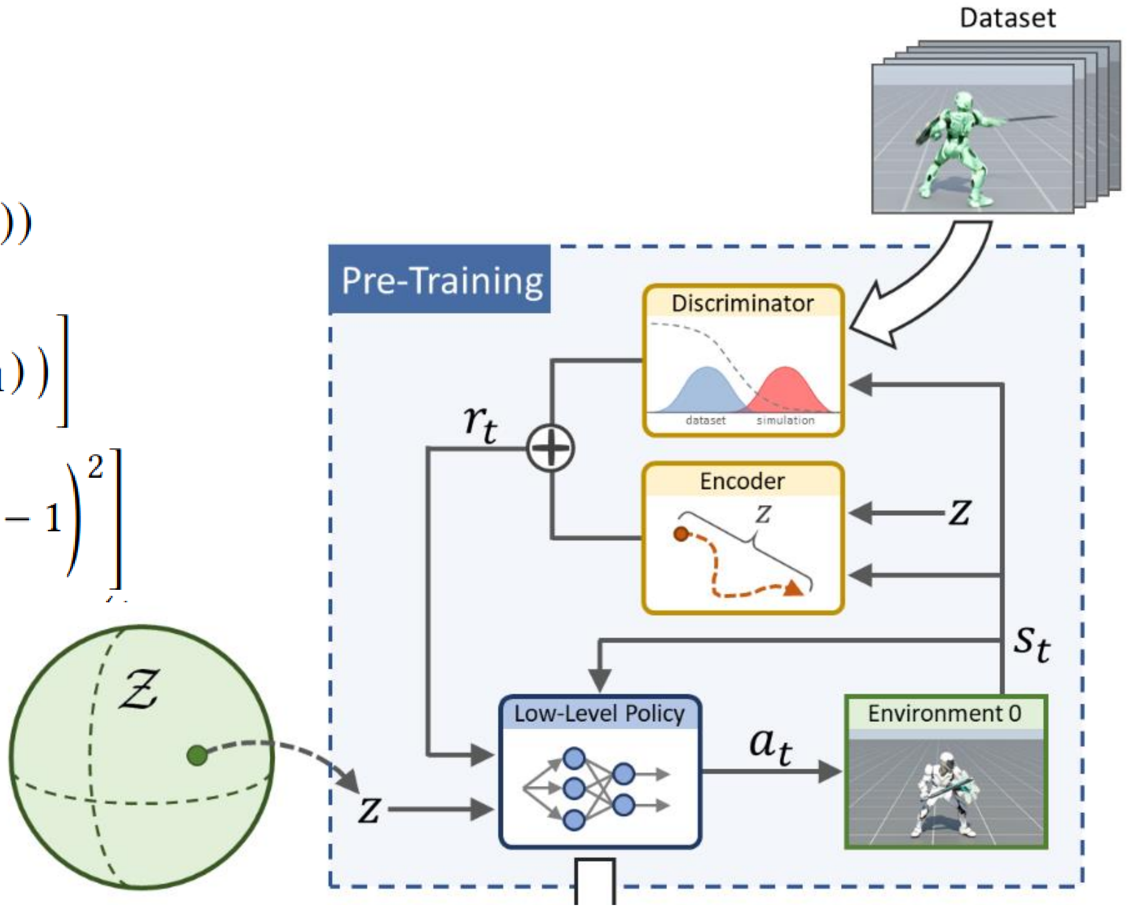


Peng, Xue Bin, et al. "Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters." *ACM Transactions On Graphics (TOG)* 41.4 (2022): 1-17.

Adversarial Skill Embeddings

diversity objective

$$\arg\max_{\pi} \quad \mathbb{E}_{p(\mathbf{Z})}\mathbb{E}_{p(\tau|\pi,\mathbf{Z})}\left[\sum_{t=0}^{T-1}\gamma^{t}\left(-\log\left(1-D\left(\mathbf{s}_t,\mathbf{s}_{t+1}\right)\right)\right.\right.$$
$$\left.\left.+\,\beta\log q\left(\mathbf{z}_t|\mathbf{s}_t,\mathbf{s}_{t+1}\right)\right)\right]$$
$$-\,w_{\text{div}}\,\mathbb{E}_{d^{\pi}(\mathbf{s})}\mathbb{E}_{\mathbf{z}_1,\mathbf{z}_2\sim p(\mathbf{z})}\left[\left(\frac{D_{\text{KL}}\left(\pi(\cdot|\mathbf{s},\mathbf{z}_1),\pi(\cdot|\mathbf{s},\mathbf{z}_2)\right)}{D_{\mathbf{z}}\left(\mathbf{z}_1,\mathbf{z}_2\right)}-1\right)^2\right]$$

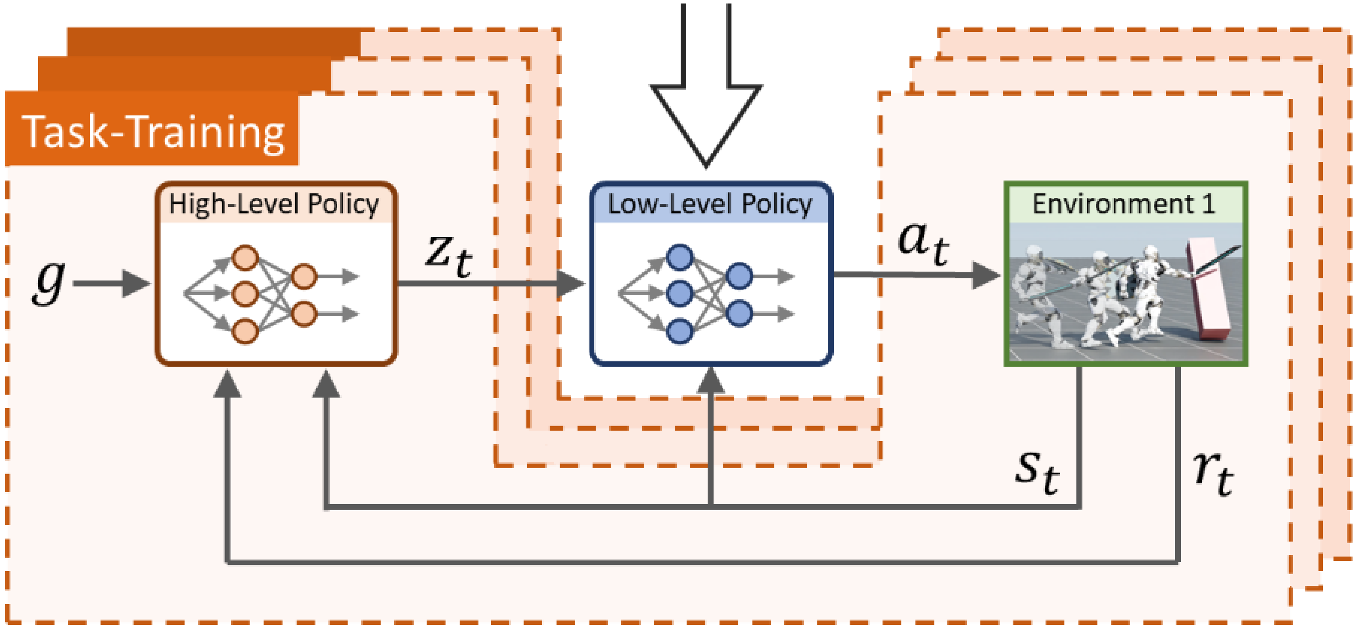


Peng, Xue Bin, et al. "Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters." *ACM Transactions On Graphics (TOG)* 41.4 (2022): 1-17.

Adversarial Skill Embeddings

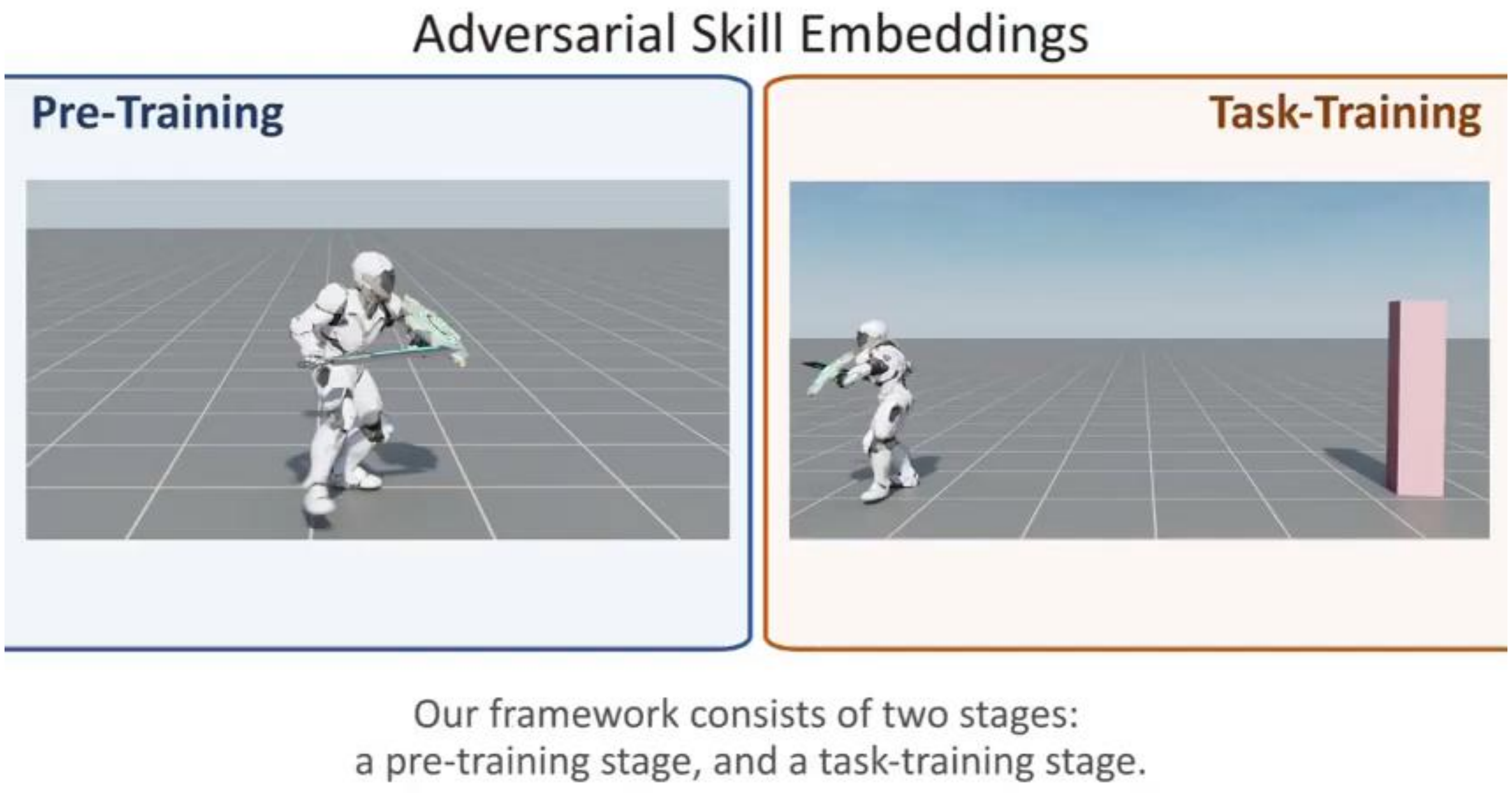
$$r_t = w_G \, r^G(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}, \mathbf{g}) - w_S \log\left(1 - D(\mathbf{s}_t, \mathbf{s}_{t+1})\right)$$

Hierarchical method



Peng, Xue Bin, et al. "Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters." *ACM Transactions On Graphics (TOG)* 41.4 (2022): 1-17.

Adversarial Skill Embeddings

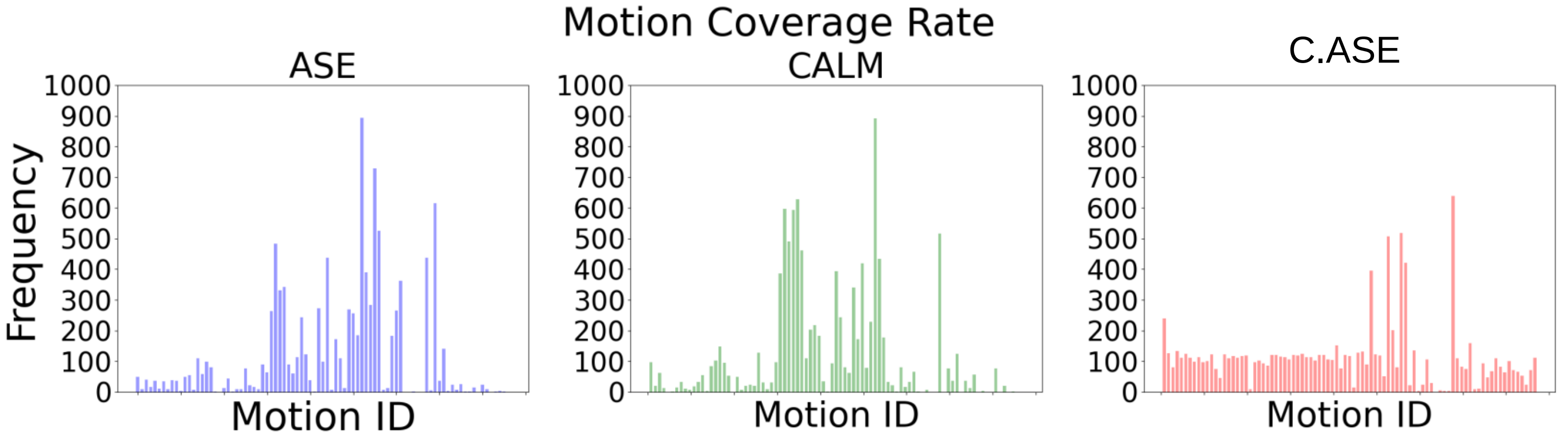


Peng, Xue Bin, et al. "Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters." *ACM Transactions On Graphics (TOG)* 41.4 (2022): 1-17.

ASE follow-ups

CALM / C.ASE

Conditioning – Tackling mode collapse



Tessler, Chen, et al. "Calm: Conditional adversarial latent models for directable virtual characters." *ACM SIGGRAPH 2023 Conference Proceedings*. 2023.

Dou, Zhiyang, et al. "C· ase: Learning conditional adversarial skill embeddings for physics-based characters." *SIGGRAPH Asia 2023 Conference Papers*. 2023.

Outline

- Recap
- Adversarial methods
- Diffusion-based methods
- Challenges in character animation

GAN

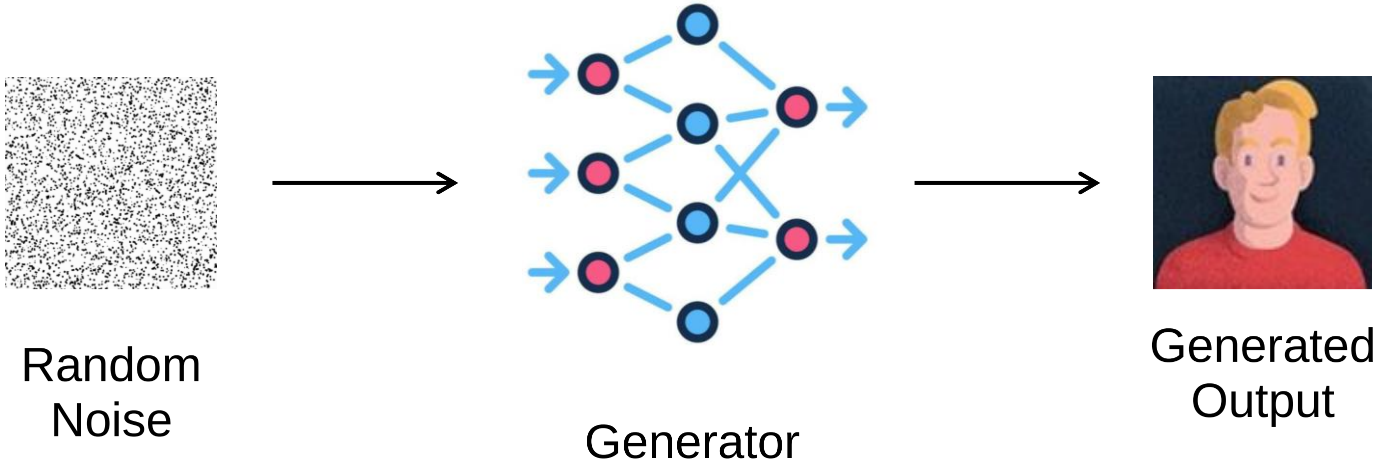
Diffusion

Dhariwal, Prafulla, and Alexander Nichol. "Diffusion models beat gans on image synthesis." *Advances in neural information processing systems* 34 (2021): 8780-8794.

Diffusion

Various images generated by DALL-E 2

Diffusion



Diffusion

Add noise, learn to denoise

Forward process (data → noise)

Reverse process (noise → data)

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." *Advances in neural information processing systems* 33 (2020): 6840-6851.

Song, Yang, et al. "Score-based generative modeling through stochastic differential equations." *arXiv preprint arXiv:2011.13456* (2020).

Diffusion

In diffusion, the forward process is modeled by a **Markov chain**: probability of each event only depends on the previous event

The transitions are **Gaussian**

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \prod_{t=1}^{T} \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

Forward process (data → noise)

$\mathbf{x}_0$ $\mathbf{x}_T$

Reverse process (noise → data)

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." *Advances in neural information processing systems* 33 (2020): 6840-6851.

Song, Yang, et al. "Score-based generative modeling through stochastic differential equations." *arXiv preprint arXiv:2011.13456* (2020).

Diffusion

Learn the reverse process

dependent on step

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \qquad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

parameterized by NN

Forward process (data → noise)

$\mathbf{x}_0$ $\mathbf{x}_T$

Reverse process (noise → data)

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." *Advances in neural information processing systems* 33 (2020): 6840-6851.

Song, Yang, et al. "Score-based generative modeling through stochastic differential equations." *arXiv preprint arXiv:2011.13456* (2020).

Diffusion

Learn the reverse process:
Find the reverse Markov transitions that maximize the **likelihood of the training data** $p(x_0)$

$$\mathbb{E}\left[-\log p_\theta(\mathbf{x}_0)\right] \leq \mathbb{E}_q\left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right] = \mathbb{E}_q\left[-\log p(\mathbf{x}_T) - \sum_{t\geq 1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})}\right] =: L$$

variational upper bound

Forward process (data → noise)

$\mathbf{x}_0$ $\mathbf{x}_T$

Reverse process (noise → data)

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." *Advances in neural information processing systems* 33 (2020): 6840-6851.

Song, Yang, et al. "Score-based generative modeling through stochastic differential equations." *arXiv preprint arXiv:2011.13456* (2020).

Diffusion

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)) \qquad \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t) = \sigma_t^2\mathbf{I}$$

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$$

reparameterization $\mathbf{x}_t(\mathbf{x}_0, \boldsymbol{\epsilon}) = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon} \text{ for } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right)$$

Loss can be simplified to:

$$L_{\text{simple}}(\theta) := \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\right\|^2\right]$$

Denoising Diffusion Probabilistic Models (DDPM)

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." *Advances in neural information processing systems* 33 (2020): 6840-6851.

Song, Yang, et al. "Score-based generative modeling through stochastic differential equations." *arXiv preprint arXiv:2011.13456* (2020).

DDPM

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \mathrm{Uniform}(\{1, \ldots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
6: **until** converged

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." *Advances in neural information processing systems* 33 (2020): 6840-6851.

How to control the output?

Need to introduce conditions $\longrightarrow$ maximize likelihood of conditional training data $p(x_0|y)$

$$\nabla_{x_t} \log p(x_t) \rightarrow \nabla_{x_t} \log p(x_t|y)$$

$$\nabla \log p(x_t|y) = \nabla \log \left(\frac{p(x_t)p(y|x_t)}{p(y)} \right) = \nabla \log p(x_t) + \underline{\nabla \log p(y|x_t)}$$

Train a classifier on noisy data $f_\phi(y|x_t)$

Then use the gradient $\nabla_{x_t} \log f_\phi(y|x_t)$ to guide diffusion **sampling** process: **Classifier Guidance**

Only at inference time

Algorithm 1 Classifier guided diffusion sampling, given a diffusion model $(\mu_\theta(x_t), \Sigma_\theta(x_t))$, classifier $f_\phi(y|x_t)$, and gradient scale s.

Input: class label y, gradient scale s
$x_T \leftarrow$ sample from $\mathcal{N}(0, \mathbf{I})$
for all t from T to 1 **do**
  $\mu, \Sigma \leftarrow \mu_\theta(x_t), \Sigma_\theta(x_t)$
  $x_{t-1} \leftarrow$ sample from $\mathcal{N}(\mu + s\Sigma \nabla_{x_t} \log f_\phi(y|x_t), \Sigma)$
end for
return x_0

Dhariwal, Prafulla, and Alexander Nichol. "Diffusion models beat gans on image synthesis." *Advances in neural information processing systems* 34 (2021): 8780-8794.

How to control the output?

Classifier-free guidance

Classifier doesn't need to be explicitly learned

$p_\theta(\mathbf{x})$ parameterized through $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$

$$
\begin{aligned}
\nabla_{\mathbf{x}_t} \log p(y|\mathbf{x}_t) &= \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|y) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) \\
&= -\frac{1}{\sqrt{1-\bar{\alpha}_t}}\Big(\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, y) - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\Big) \\
\bar{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, t, y) &= \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, y) - \sqrt{1-\bar{\alpha}_t}\; w \nabla_{\mathbf{x}_t} \log p(y|\mathbf{x}_t) \\
&= \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, y) + w\big(\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, y) - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\big) \\
&= (w+1)\underbrace{\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, y)}_{\text{conditional}} - w\underbrace{\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)}_{\text{unconditional}}
\end{aligned}
$$

$$\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) = \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, y=\varnothing).$$

Unlike classifier guidance, it requires specific training.

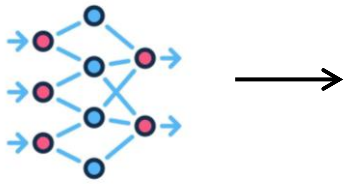
Ho, Jonathan, and Tim Salimans. "Classifier-free diffusion guidance." *arXiv preprint arXiv:2207.12598* (2022).
https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

Human Motion Diffusion Model (MDM)

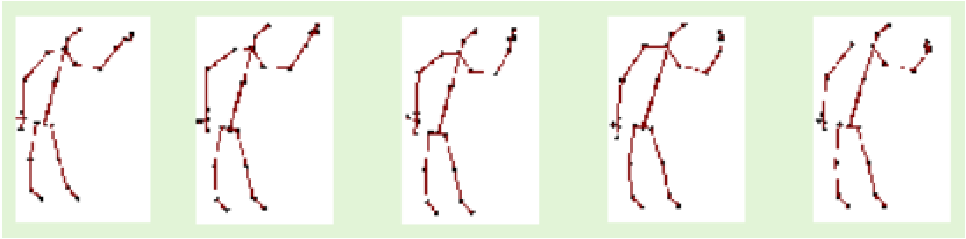
Tevet, Guy, et al. "Human motion diffusion model." *arXiv preprint arXiv:2209.14916* (2022).

Human Motion Diffusion Model (MDM)

Generate sequence of motion: Transformer



Generator

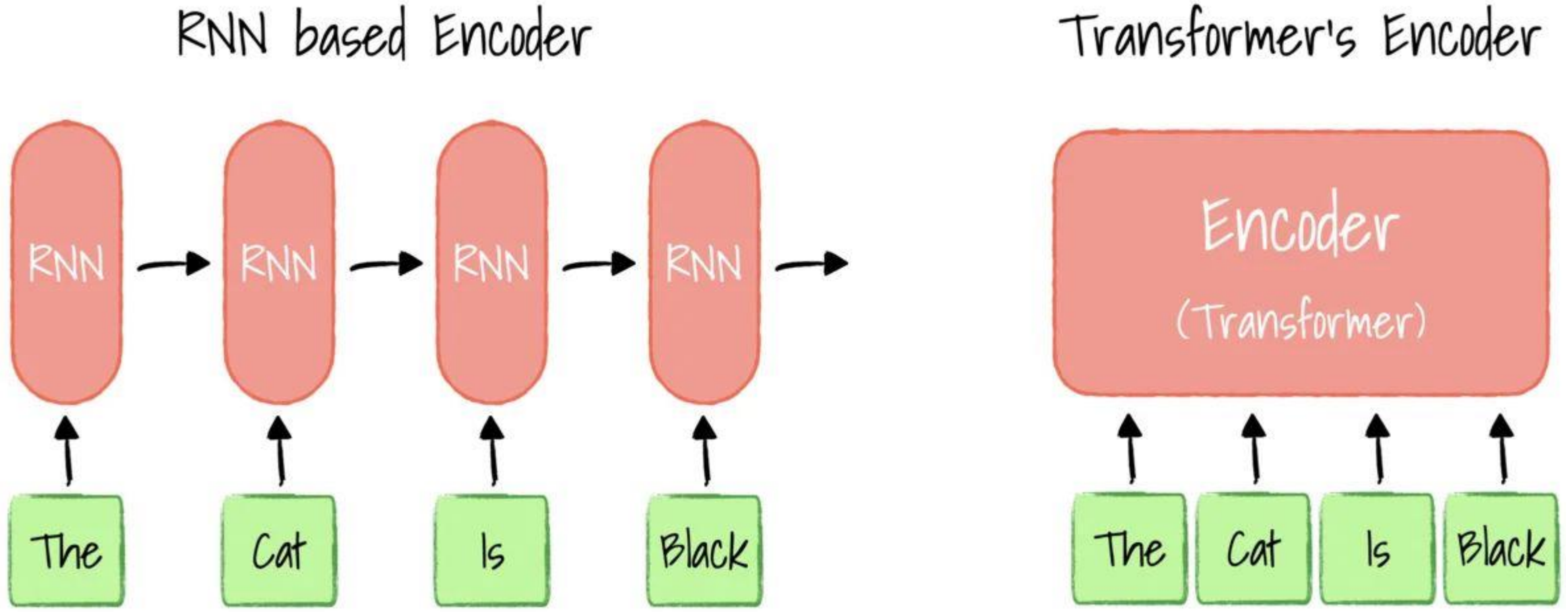


Motion sequence

Tevet, Guy, et al. "Human motion diffusion model." *arXiv preprint arXiv:2209.14916* (2022).

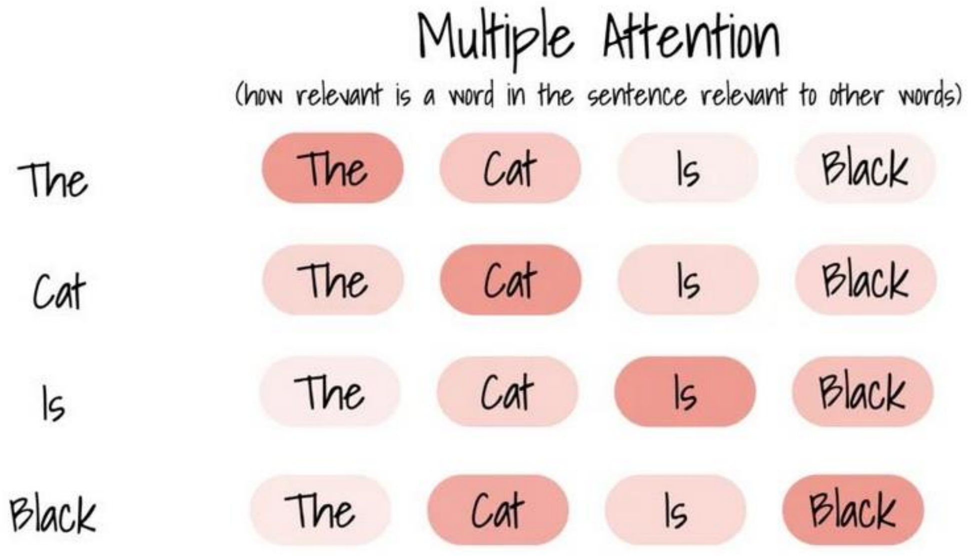
Transformers

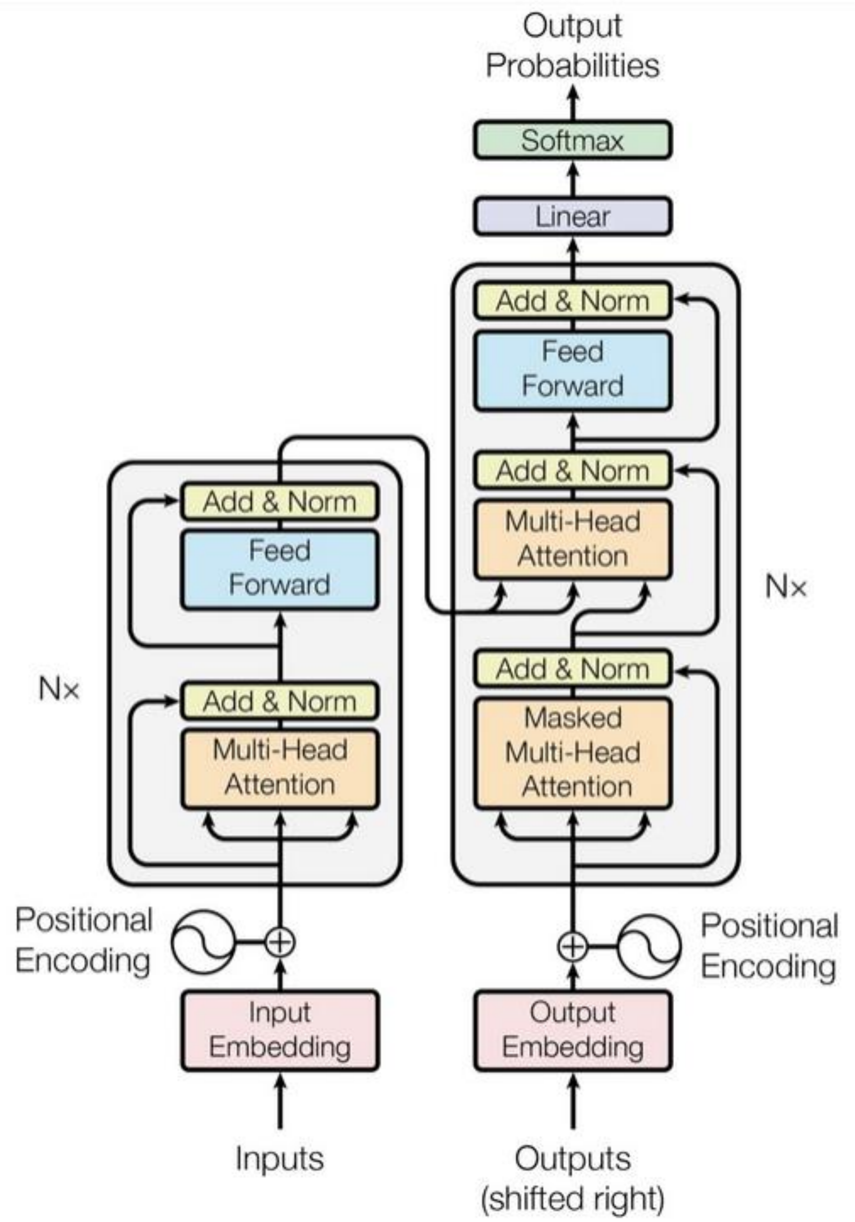
Another way to represent sequence data



https://jinglescode.github.io/2020/05/27/illustrated-guide-transformer/

Transformers





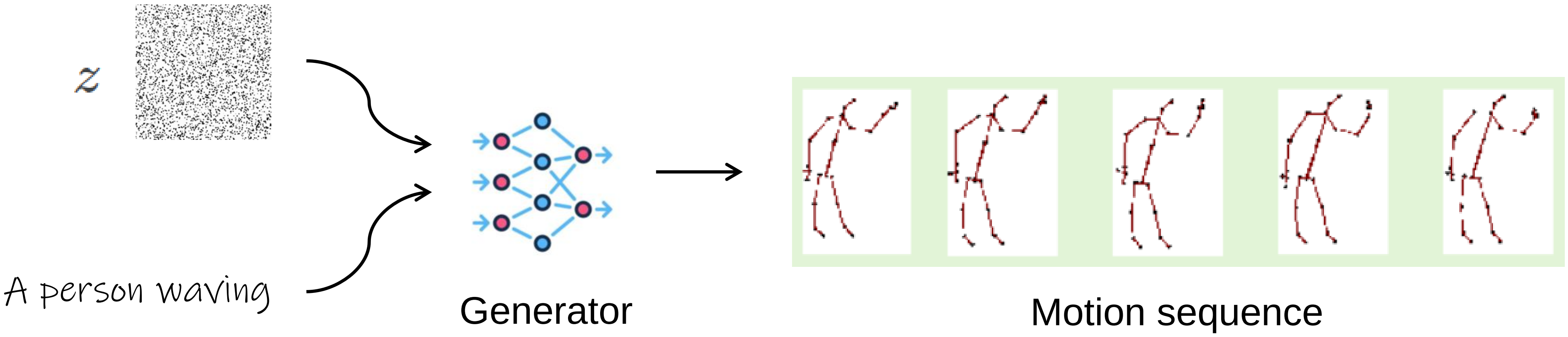
Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

https://jinglescode.github.io/2020/05/27/illustrated-guide-transformer/

Human Motion Diffusion Model (MDM)

Generate sequence of motion: Transformer

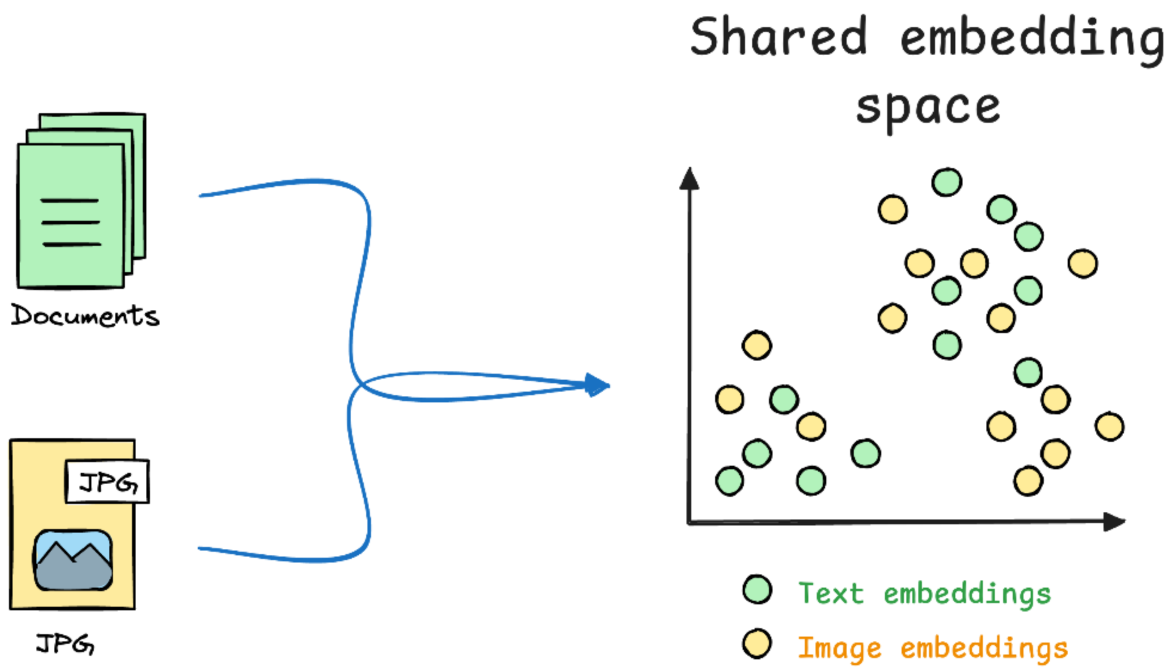
Conditioned on text prompt: CLIP embedding



Tevet, Guy, et al. "Human motion diffusion model." *arXiv preprint arXiv:2209.14916* (2022).

CLIP Embedding

A multi-modal representation that maps images and text into a shared space



Human Motion Diffusion Model (MDM)

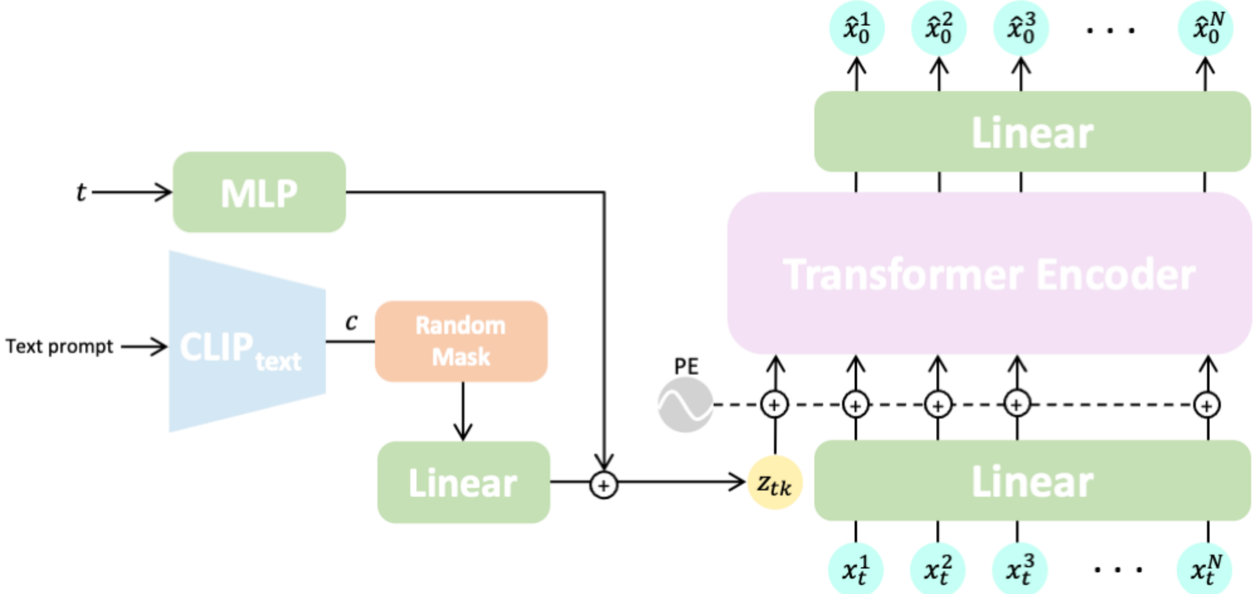
Generate sequence of motion: Transformer

Conditioned on text prompt: CLIP embedding

Classifier free training

$$\mathcal{L}_{\text{simple}} = E_{x_0 \sim q(x_0|c), t \sim [1,T]}[\|x_0 - G(x_t, t, c)\|_2^2]$$

predict x_0 directly, instead of predicting the noise



Tevet, Guy, et al. "Human motion diffusion model." *arXiv preprint arXiv:2209.14916* (2022).

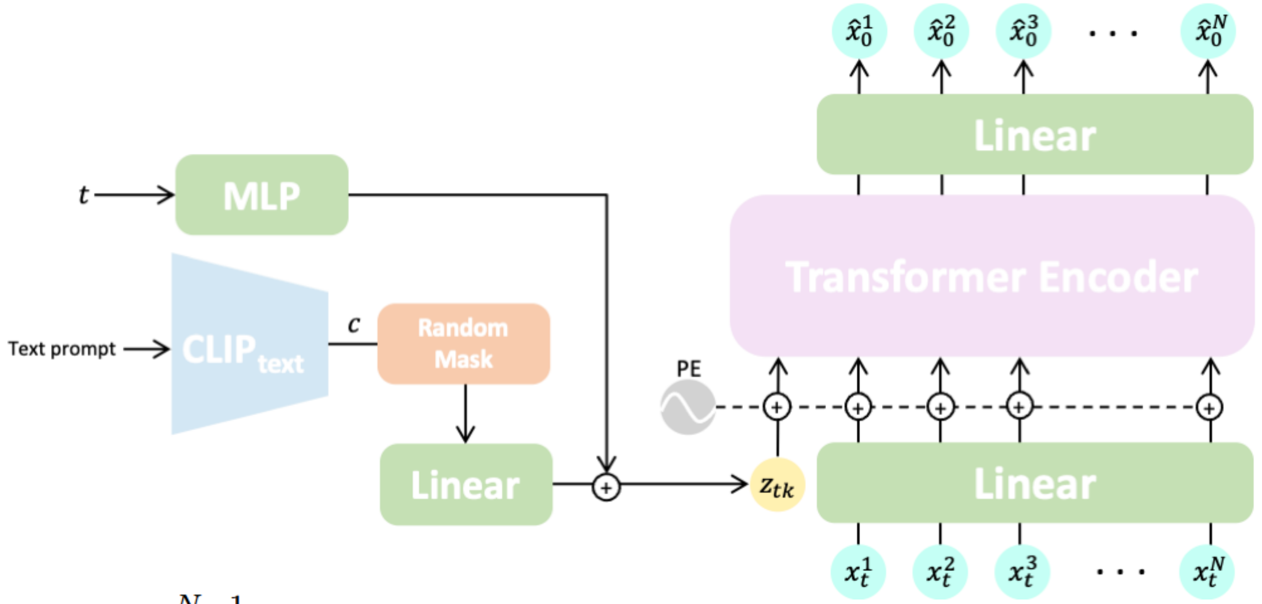
Human Motion Diffusion Model (MDM)

Generate sequence of motion: Transformer

Conditioned on text prompt: CLIP embedding

Classifier free training

$$\mathcal{L}_{\text{simple}} = E_{x_0 \sim q(x_0|c), t \sim [1,T]}[\|x_0 - G(x_t, t, c)\|_2^2]$$



Geometric losses

$$\mathcal{L}_{\text{pos}} = \frac{1}{N}\sum_{i=1}^{N}\|FK(x_0^i) - FK(\hat{x}_0^i)\|_2^2,$$

$$\mathcal{L}_{\text{foot}} = \frac{1}{N-1}\sum_{i=1}^{N-1}\|(FK(\hat{x}_0^{i+1}) - FK(\hat{x}_0^i)) \cdot f_i\|_2^2,$$

$$\mathcal{L}_{\text{vel}} = \frac{1}{N-1}\sum_{i=1}^{N-1}\|(x_0^{i+1} - x_0^i) - (\hat{x}_0^{i+1} - \hat{x}_0^i)\|_2^2$$

Tevet, Guy, et al. "Human motion diffusion model." *arXiv preprint arXiv:2209.14916* (2022).

Human Motion Diffusion Model

Guy Tevet Sigal Raab Brian Gordon Yonatan Shafir
Daniel Cohen-Or Amit H. Bermano

Tel Aviv University, Israel

https://github.com/andreas128/RePaint

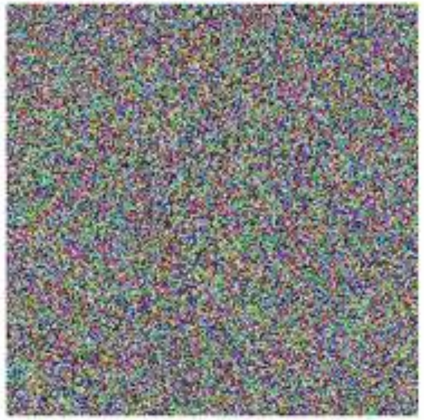
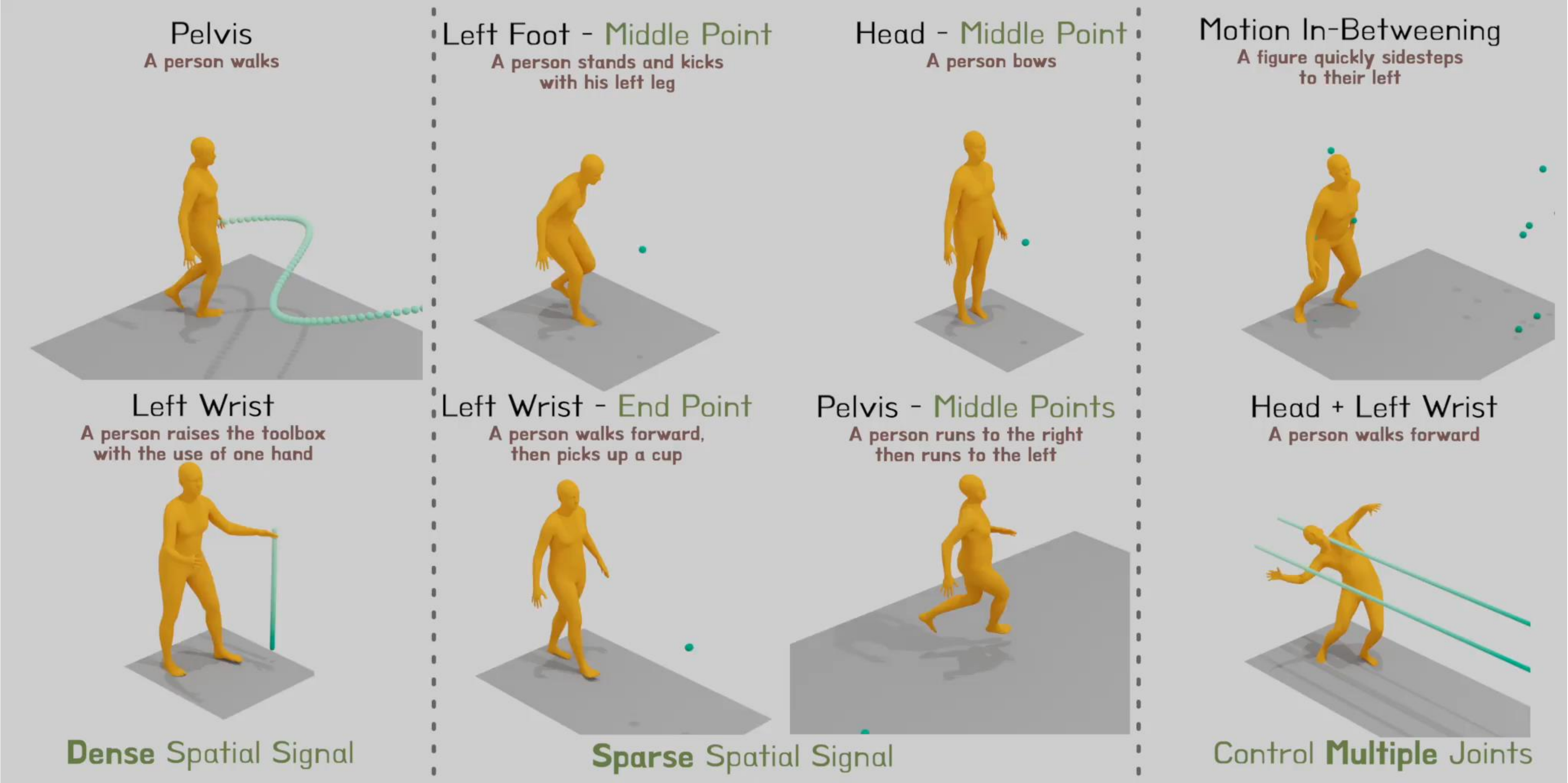


Image Diffusion

Motion Diffusion

Tevet, Guy, et al. "Human motion diffusion model." *arXiv preprint arXiv:2209.14916* (2022).

Omni Control

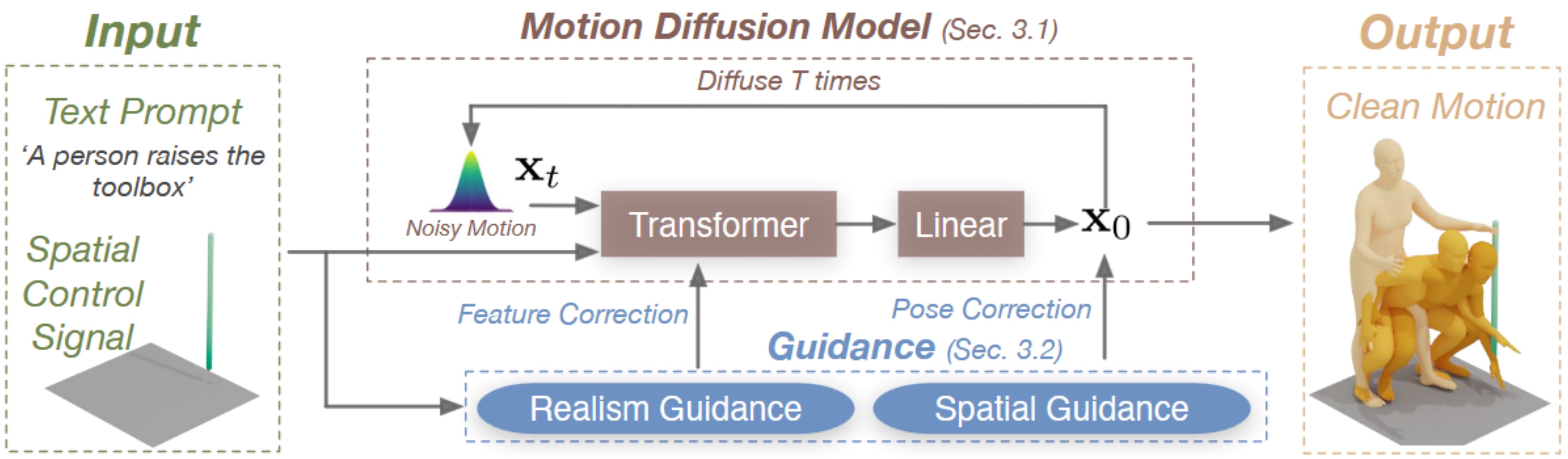


Xie, Yiming, et al. "Omnicontrol: Control any joint at any time for human motion generation." *arXiv preprint arXiv:2310.08580* (2023).

Omni Control

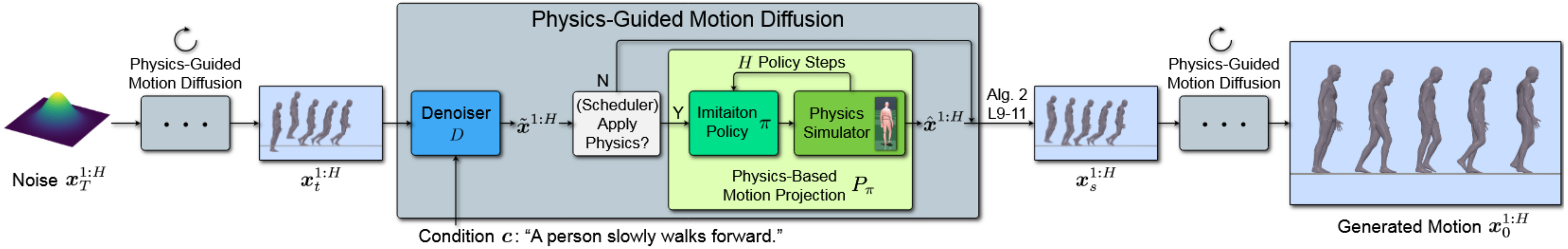
Analytic function for spatial guidance $$\boldsymbol{\mu}_t = \boldsymbol{\mu}_t - \tau \nabla_{\boldsymbol{\mu}_t} G(\boldsymbol{\mu}_t, \mathbf{c})$$

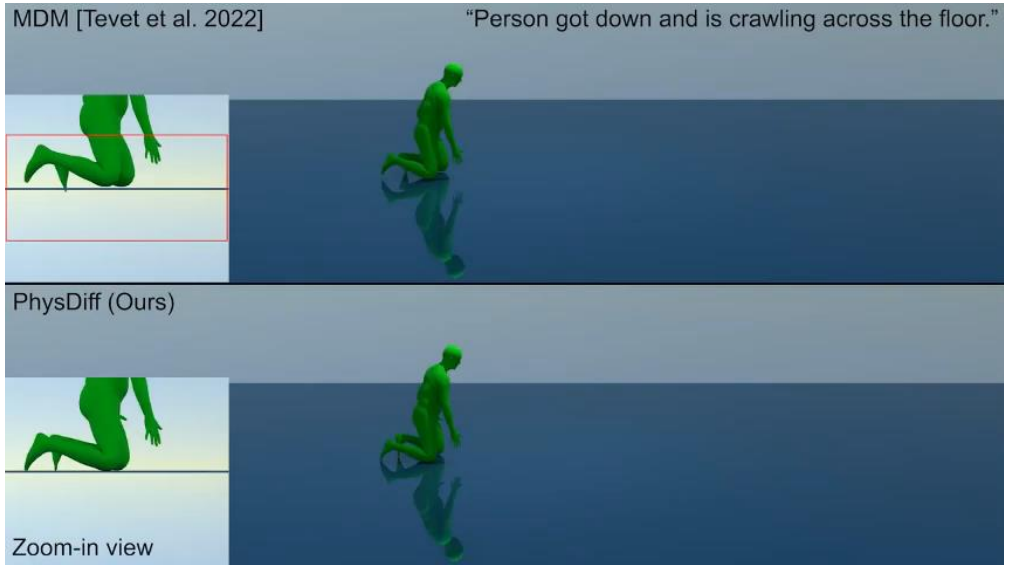
Realism guidance: a trainable copy of transformer encoder to learn to enforce the spatial constraint



Xie, Yiming, et al. "Omnicontrol: Control any joint at any time for human motion generation." *arXiv preprint arXiv:2310.08580* (2023).

PhysDiff: Physics-Guided Human Motion Diffusion Model





Algorithm 2 PhysDiff sampling algorithm for motion.

1: **Input**: Denoiser D, sample $\boldsymbol{x}_t^{1:H}$ at time t, condition $\boldsymbol{c}$, target time s, physics-based projection $\mathcal{P}_\pi$, $\eta \in [0, 1]$.
2: Compute the denoised motion $\tilde{\boldsymbol{x}}^{1:H} := D(\boldsymbol{x}_t^{1:H}, t, \boldsymbol{c})$.
3: **if** projection is performed at time t **then**
4: $\quad \hat{\boldsymbol{x}}^{1:H} := \mathcal{P}_\pi(\tilde{\boldsymbol{x}}^{1:H})$ # Physics-Based Projection
5: **else**
6: $\quad \hat{\boldsymbol{x}}^{1:H} := \tilde{\boldsymbol{x}}^{1:H}$
7: **end if**
8: # The remaining part is similar to DDIM
9: Obtain variance v_s as a scalar that depends on η.
10: Obtain mean μ_s:

$$\mu_s := \hat{\boldsymbol{x}}^{1:H} + \frac{\sqrt{\sigma_s^2 - v_s}}{\sigma_t}(\boldsymbol{x}_t^{1:H} - \hat{\boldsymbol{x}}^{1:H})$$

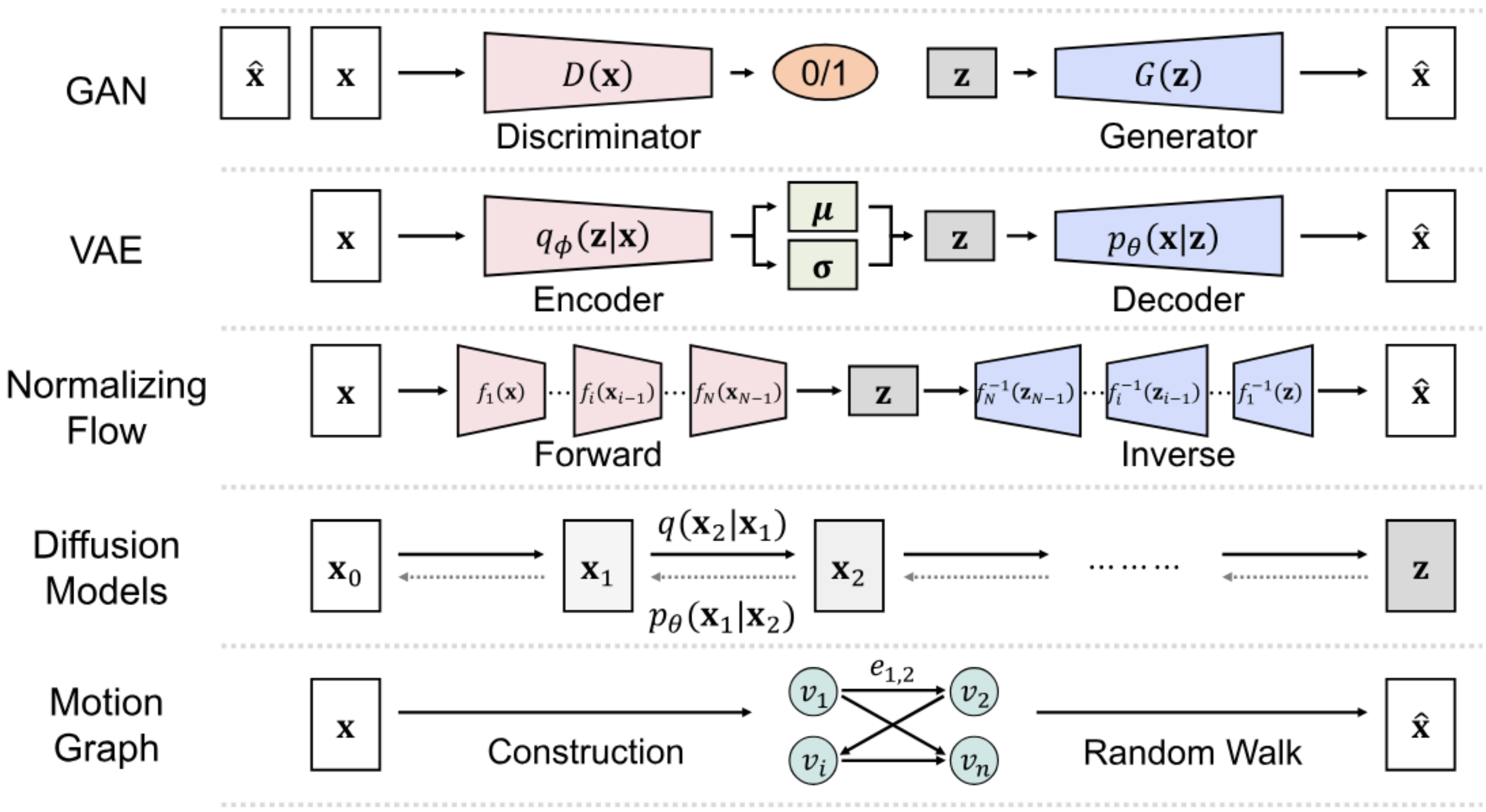
11: Draw sample $\boldsymbol{x}_s^{1:H} \sim \mathcal{N}(\mu_s, v_s \boldsymbol{I})$.

Yuan, Ye, et al. "Physdiff: Physics-guided human motion diffusion model." *Proceedings of the IEEE/CVF international conference on computer vision.* 2023.

Problem with diffusion

- Still computationally heavy at training time
- Usually needs large datasets to capture motion diversity

Overview of different generative models



Zhu, Wentao, et al. "Human motion generation: A survey." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46.4 (2023): 2430-2449.

Overview of different generative models

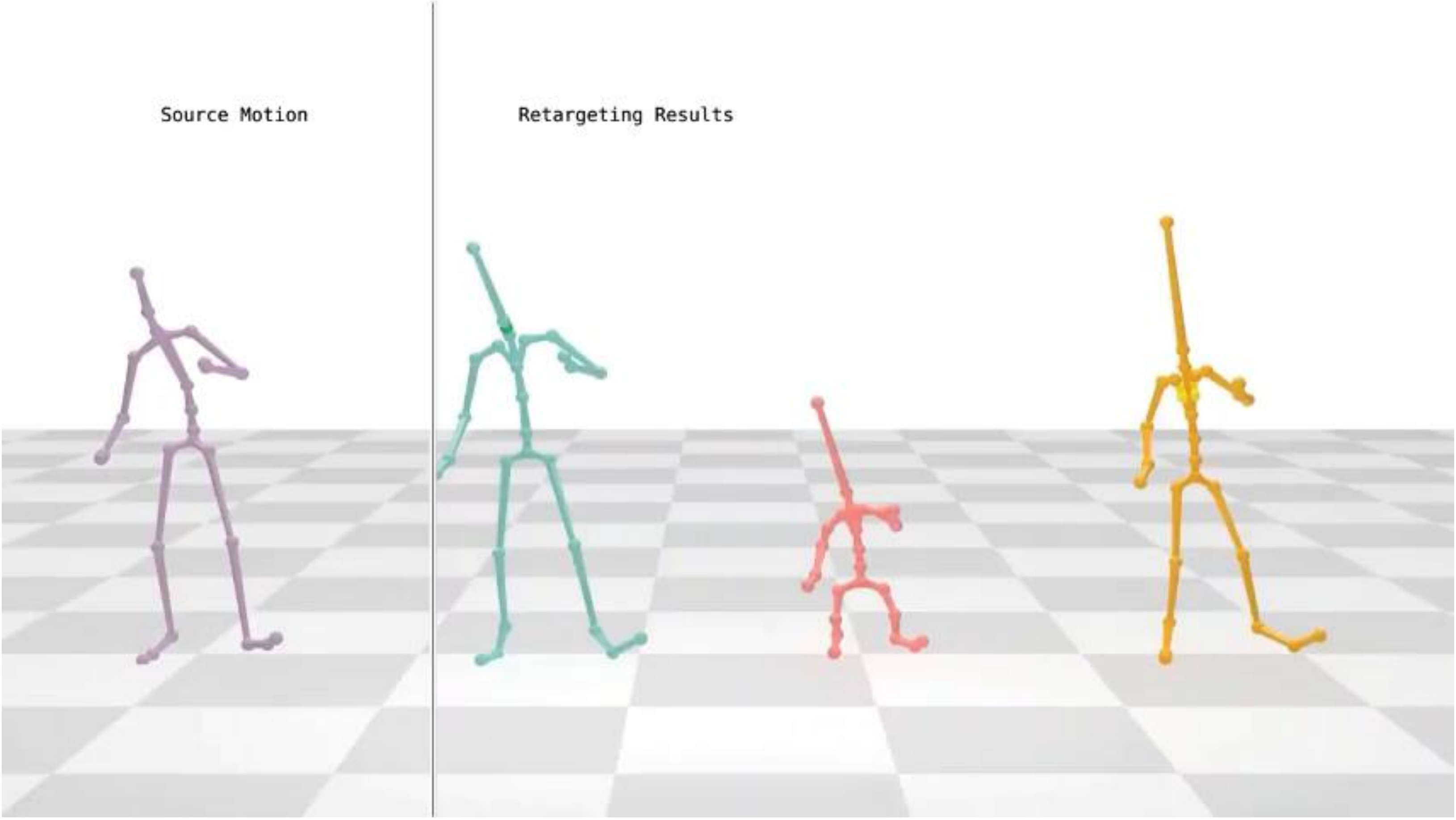
Property	VAE	GAN	Diffusion
Training Stability	✅ Stable	❌ Unstable	✅ Stable
Sample Diversity	❗ Moderate	❗ Moderate	✅ High (multi-modal)
Output Sharpness	❌ Blurry	✅ Sharp	✅ Sharp
Control Conditioning	❗ Manual	❗ Complex	✅ Flexible (guidance)
Inference Speed	✅ Fast	✅ Fast	❌ Slow
Best Use Case	Latent-space interpolation	Stylized generation	Multi-modal and controllable generation

Outline

- Recap
- Adversarial methods
- Diffusion-based methods
- Challenges in character animation

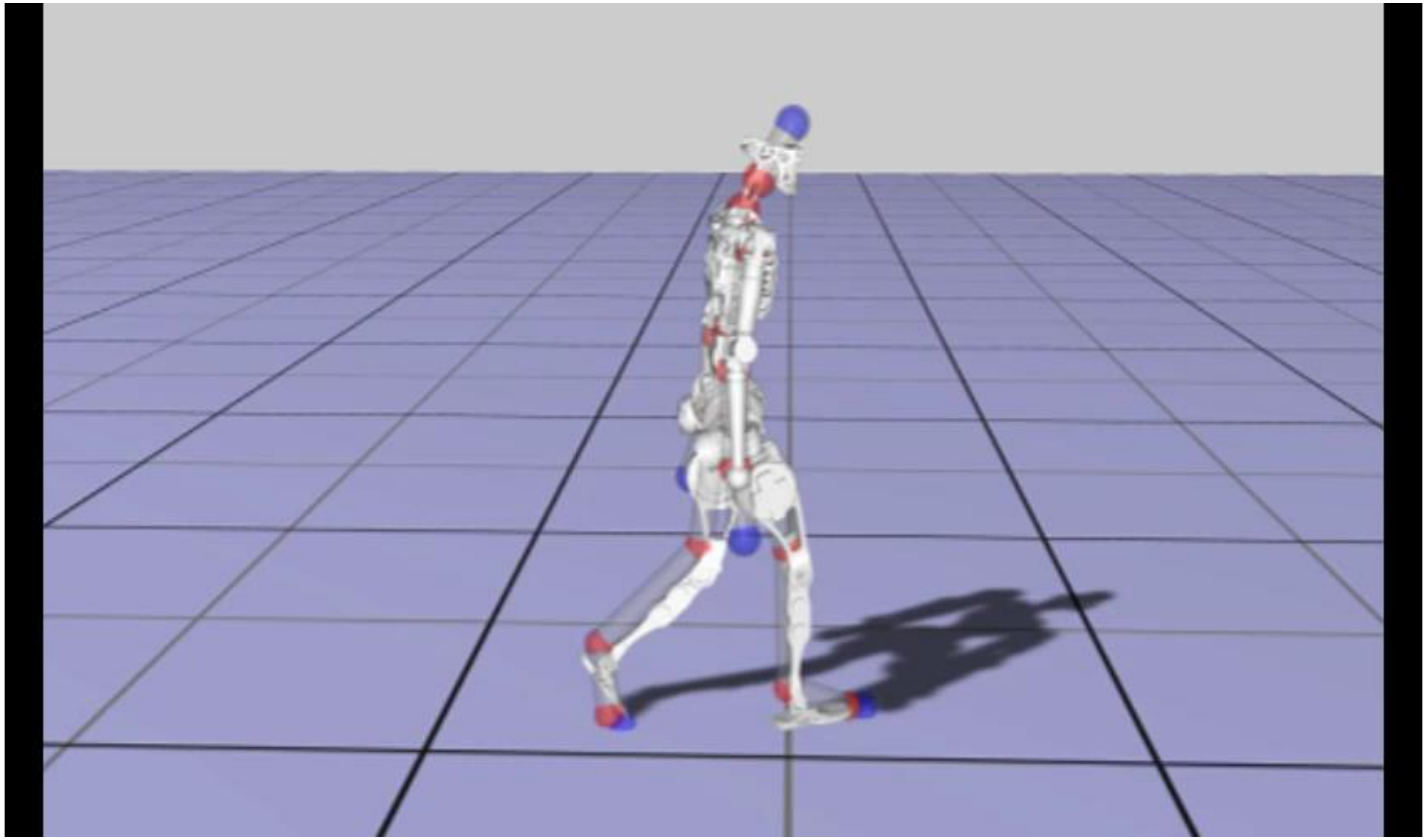
Challenges in character animation

Retargeting



Lee, Sunmin, et al. "Same: Skeleton-agnostic motion embedding for character animation." *SIGGRAPH Asia 2023 Conference Papers*. 2023.

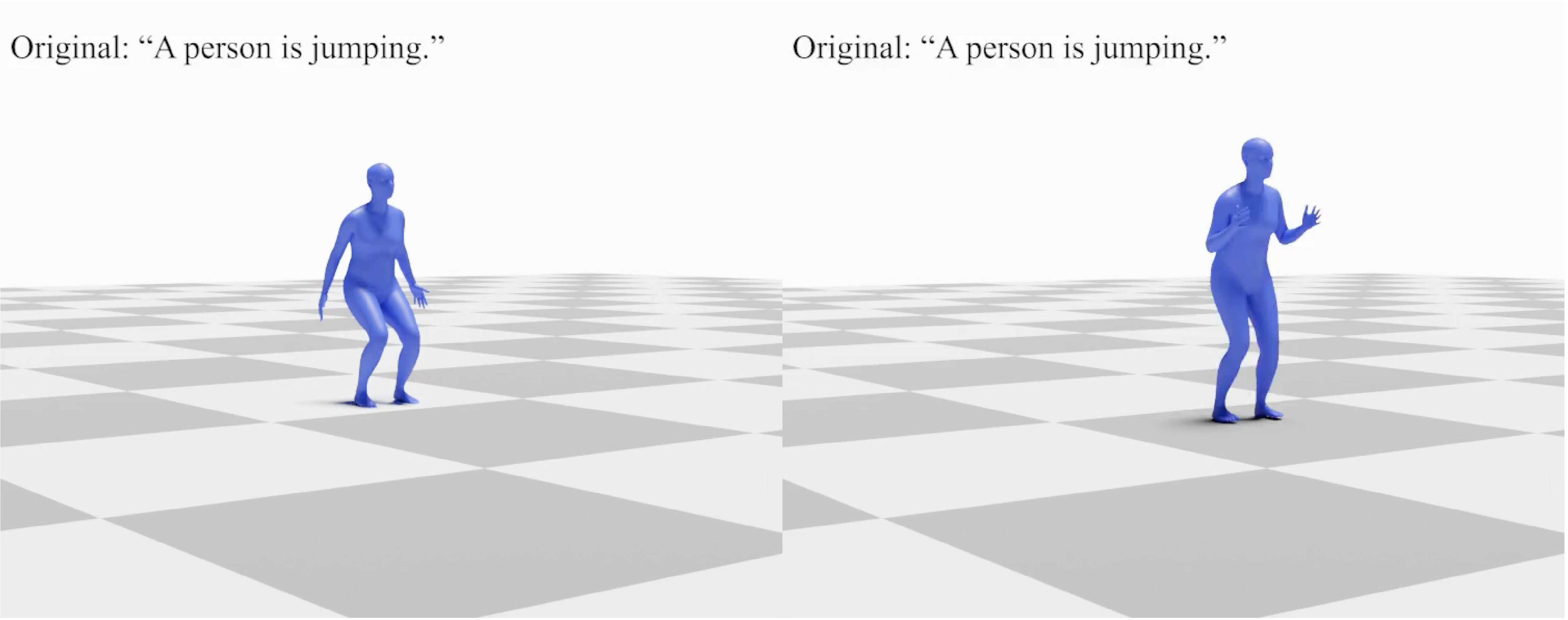
Retargeting



Motion in-betweening / inpainting

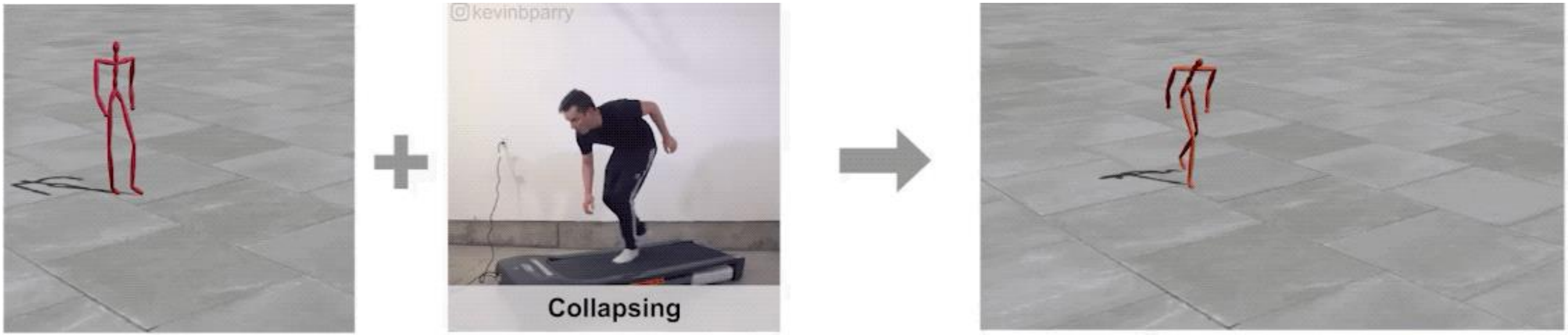
Agrawal, Dhruv, et al. "SKEL-Betweener: a Neural Motion Rig for Interactive Motion Authoring." *ACM Transactions on Graphics (TOG)* 43.6 (2024): 1-11.

Motion editing



Goel, Purvi, et al. "Iterative motion editing with natural language." *ACM SIGGRAPH 2024 Conference Papers*. 2024.

Stylization



Aberman, Kfir, et al. "Unpaired motion style transfer from video to animation." *ACM Transactions on Graphics (TOG)* 39.4 (2020): 64-1.

Thank you!