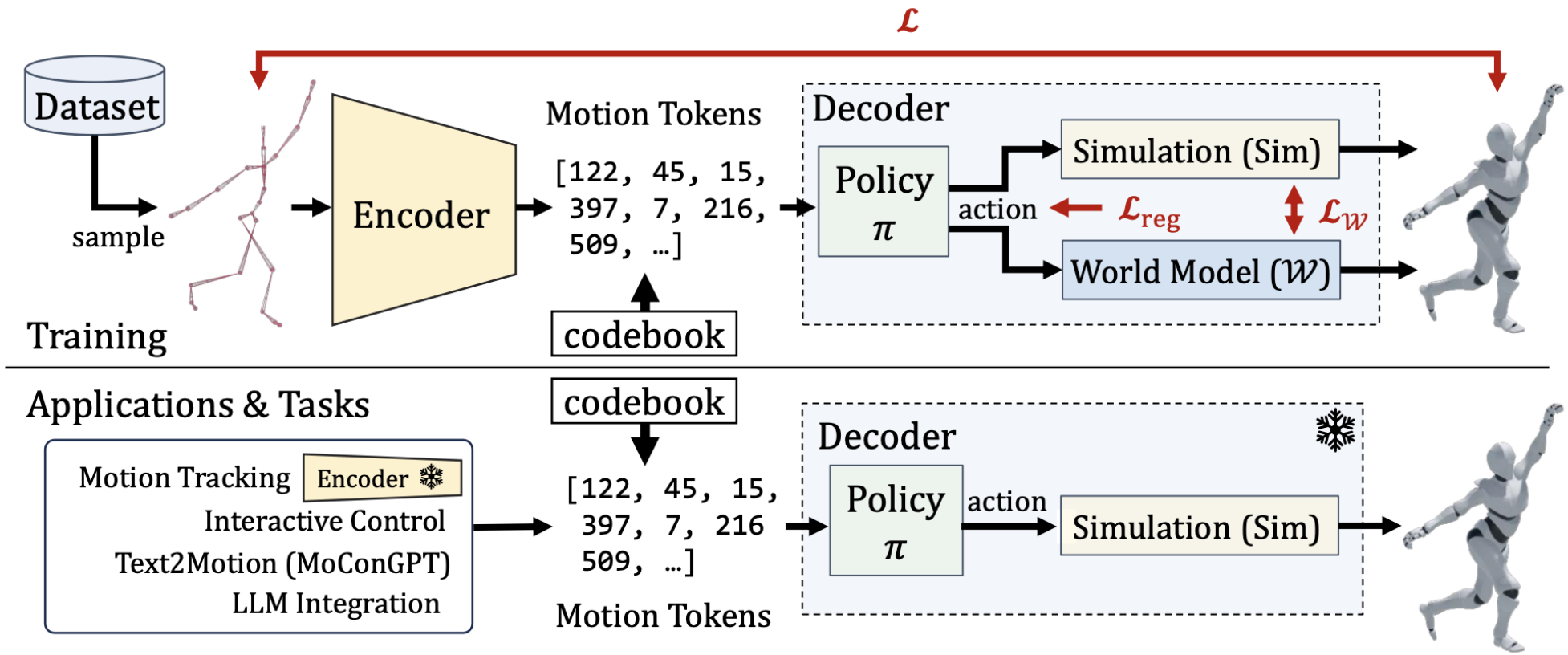



# MoConVQ: Unified Physics-Based Motion Control via Scalable Discrete Representations

SIGGRAPH 2024

Heyuan Yao, Zhenhua Song, Yuyang Zhou, Tenglong Ao, Baoquan Chen, and Libin Liu

Peking University, China and State Key Lab of General AI, China

ACM Trans. Graph. 43, 4, Article 144 (July 2024), 21 pages

# Contents


01 Introduction
- Related Work
- Goal

02 Background
- Reinforcement Learning
- AutoEncoder

03 Method
- Overview
- detail

04 Experiments

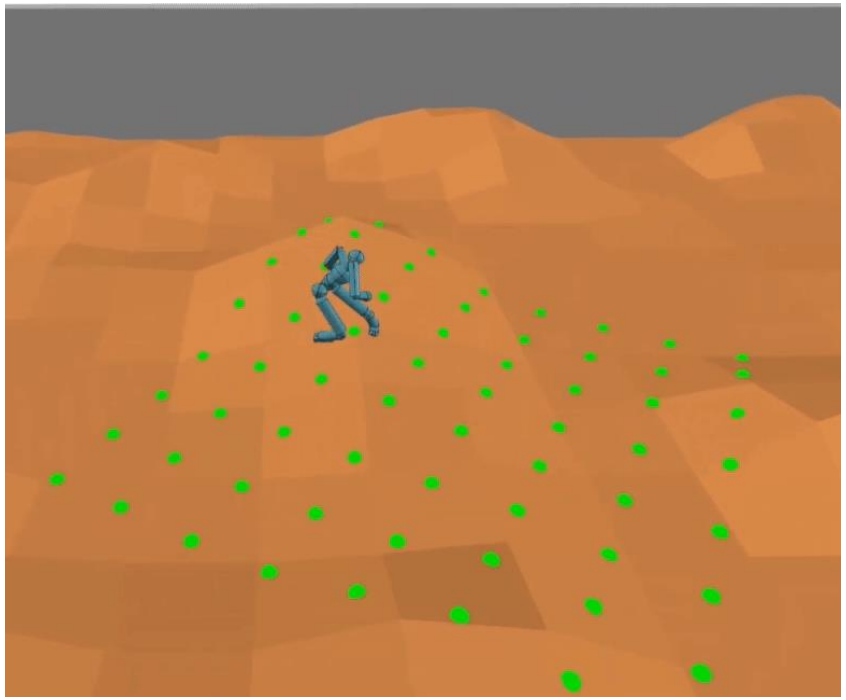
05 Evaluation

05 Reference

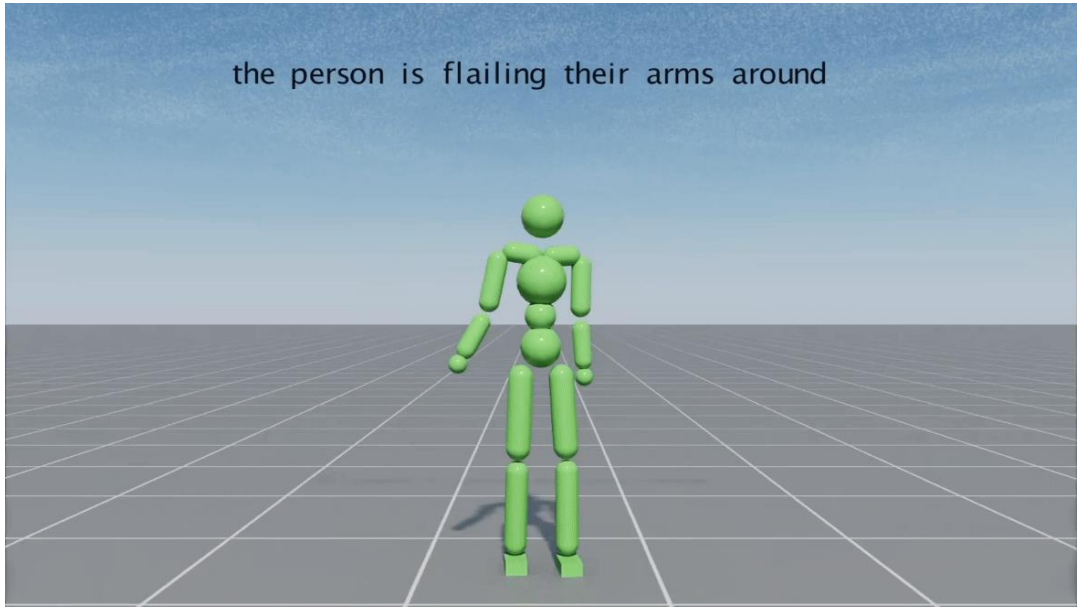
# 01 Introduction

# Introduction – Related Work

Generative model + DRL



<conditional VAEs>

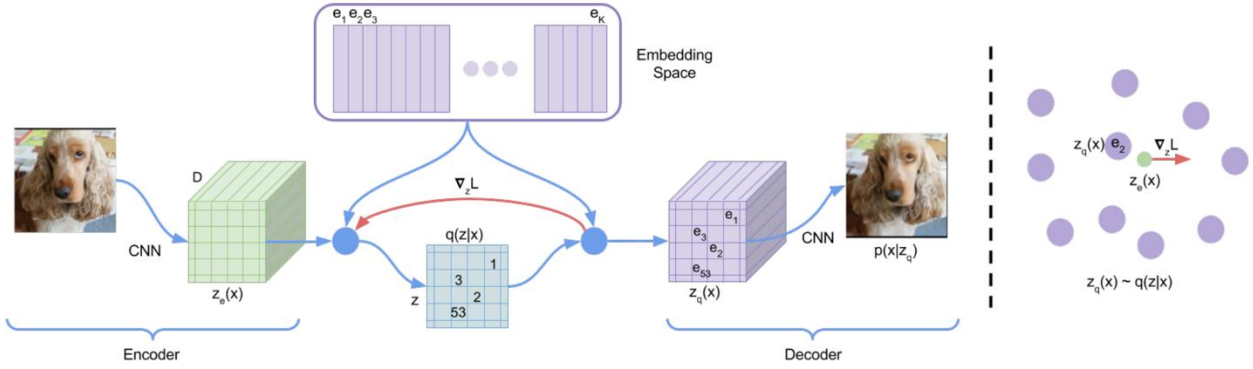



<SuperPADL>

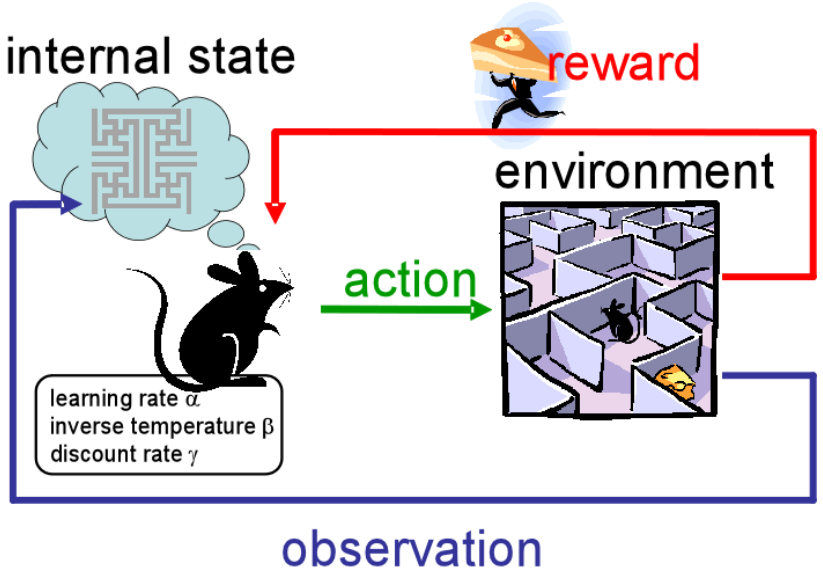

- Jungdam Won, Deepak Gopinath, and Jessica Hodgins. 2022. Physics-based character controllers using conditional VAEs. ACM Trans.
- Jordan Juravsky, Yunrong Guo, Sanja Fidler, and Xue Bin Peng. 2024. SuperPADL: Scaling Language-Directed Physics-Based Control with Progressive Supervised Distillation. In ACM SIGGRAPH 2024 Conference Papers

# Introduction

Physics-based motion control leveraging scalable discrete representations




<VQ-VAE>

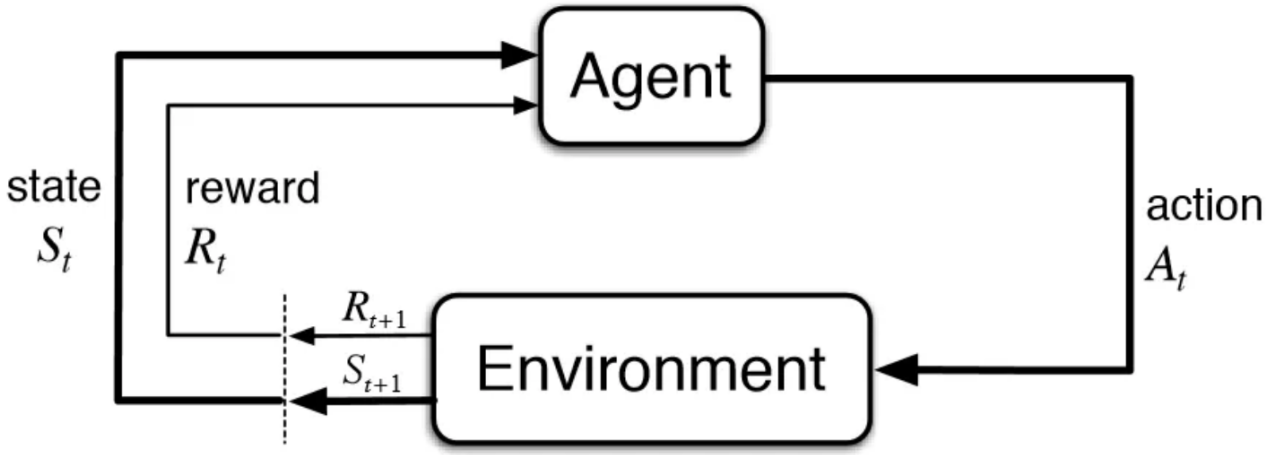



<model-based RL>

- van den Oord, O. Vinyals, and K. Kavukcuoglu, Neural Discrete Representation Learning. 2018. [Online]. Available: https://arxiv.org/abs/1711.00937
- https://becominghuman.ai/the-very-basics-of-reinforcement-learning-154f28a79071

# 02 Background

# Background – Reinforcement Learning




<Structure of RL>

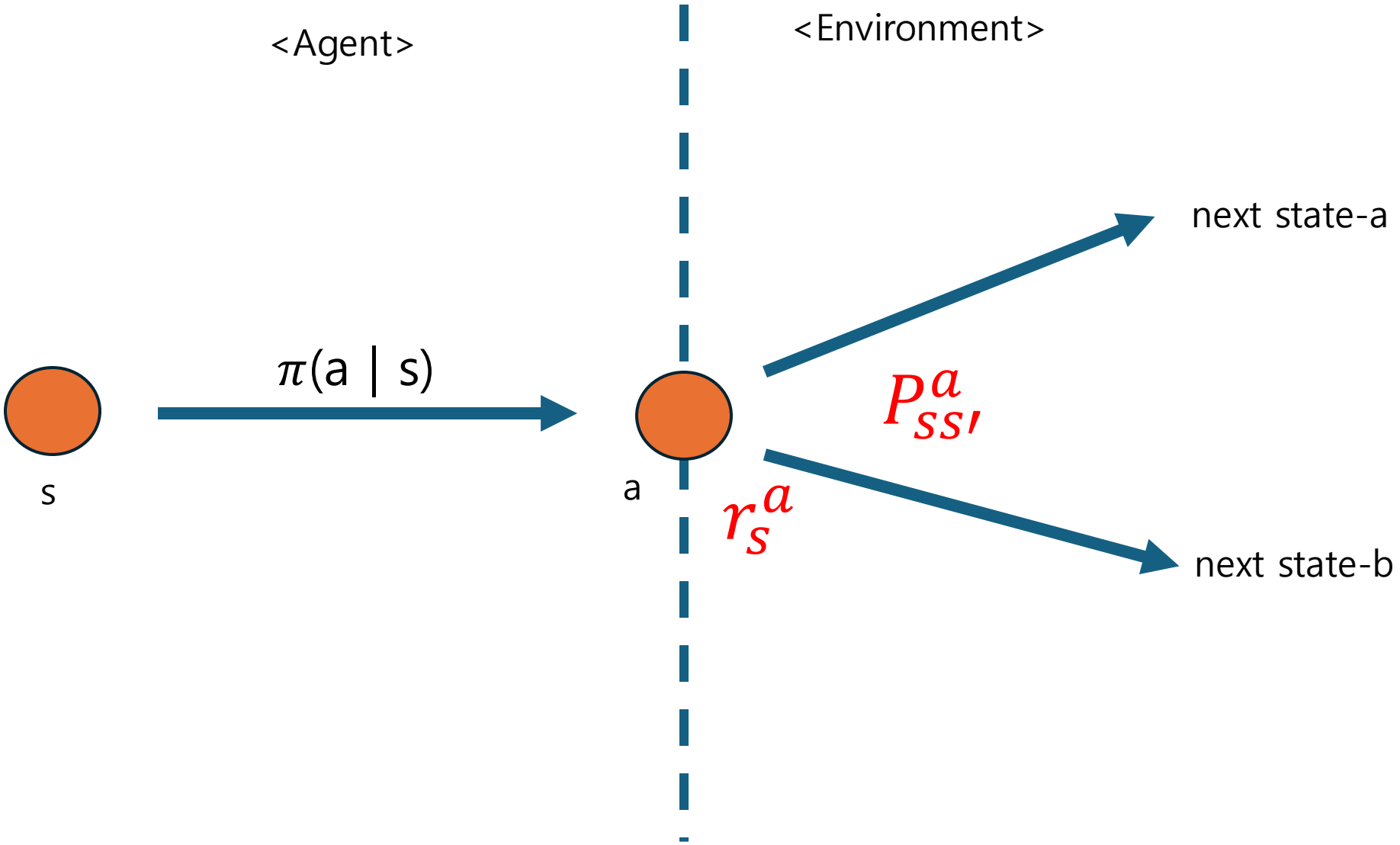
<Model-based: planning>

Reward function
Transition Probability

<Model-free>

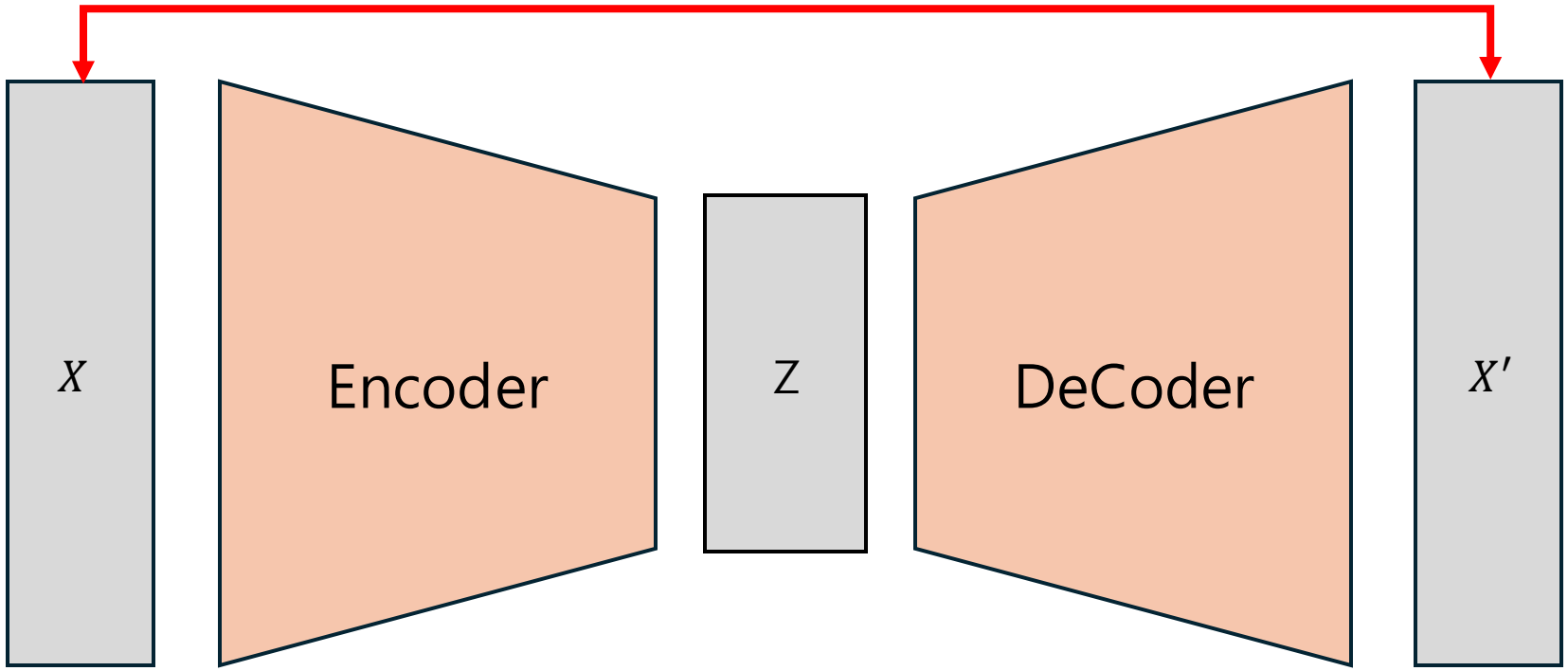
Unknown Reward function
Unknown Transition Probability

# Background – Reinforcement Learning



# Background – AutoEncoder

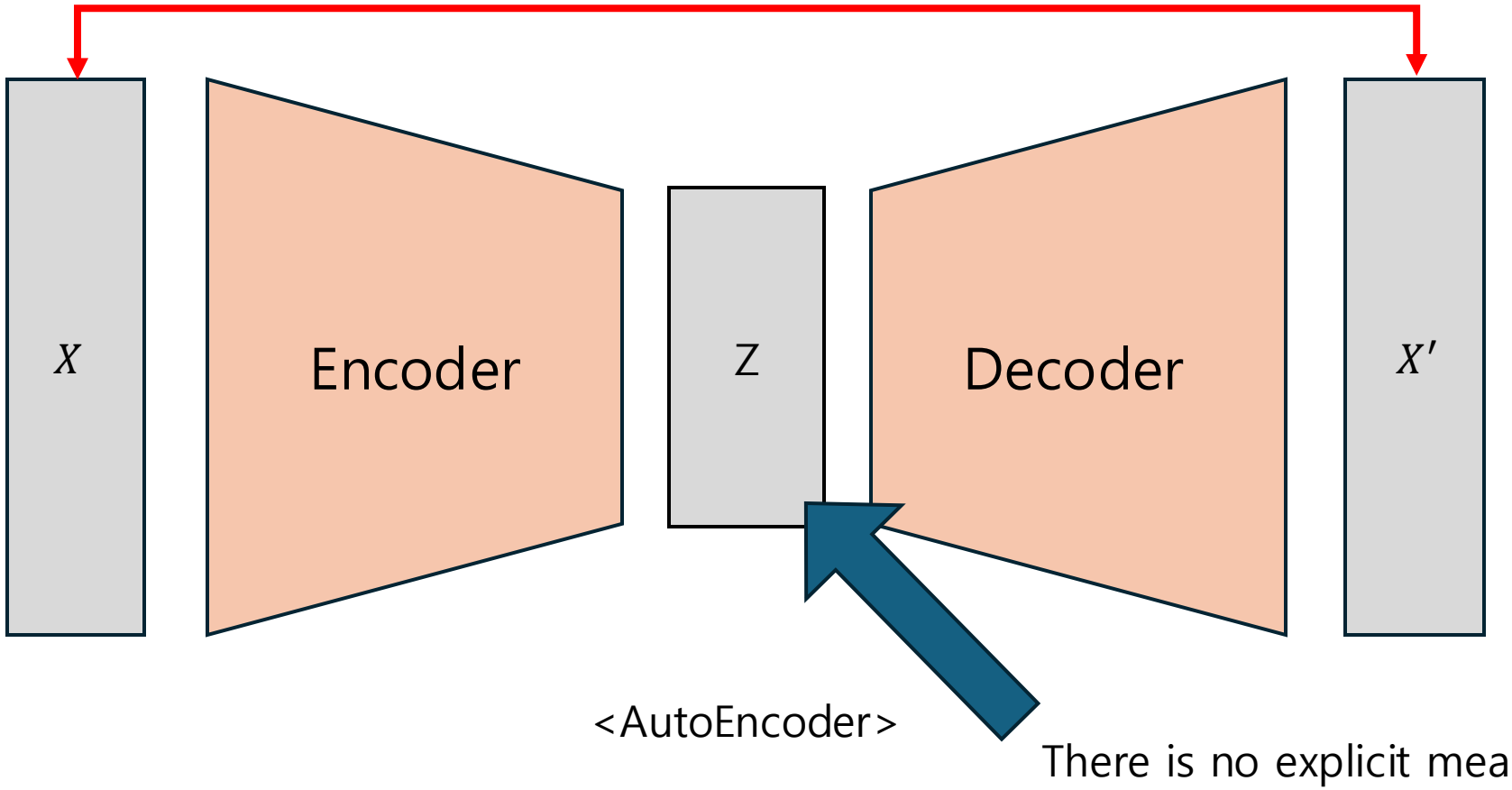
$$L = ||X - X'||_2^2$$



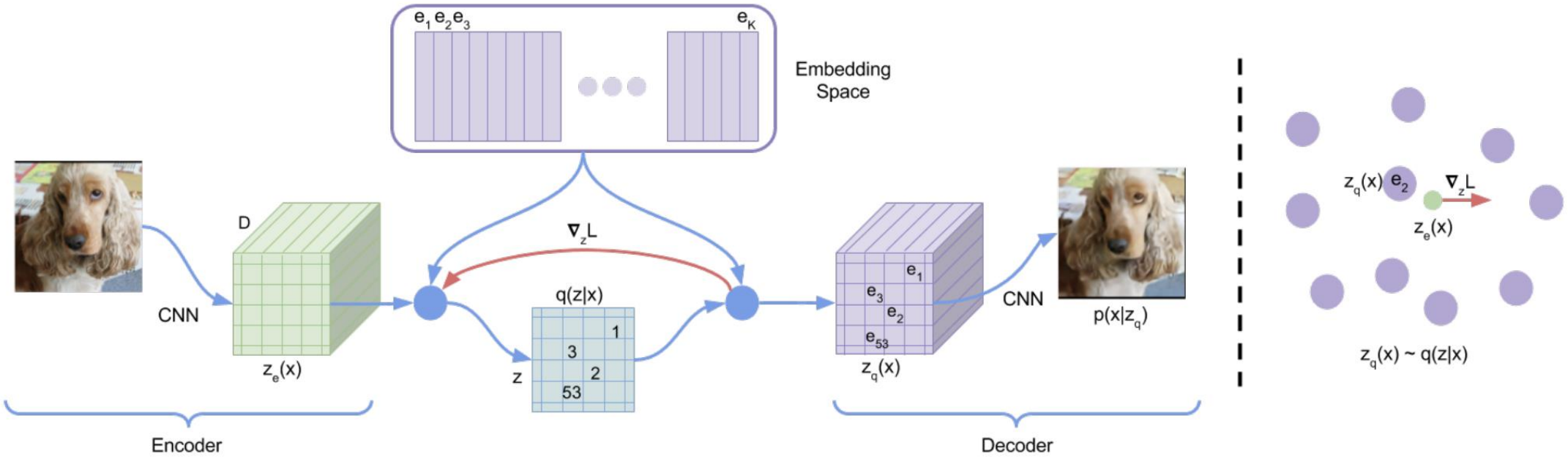

<AutoEncoder>

# Background – AutoEncoder

$$L = ||X - X'||_2^2$$



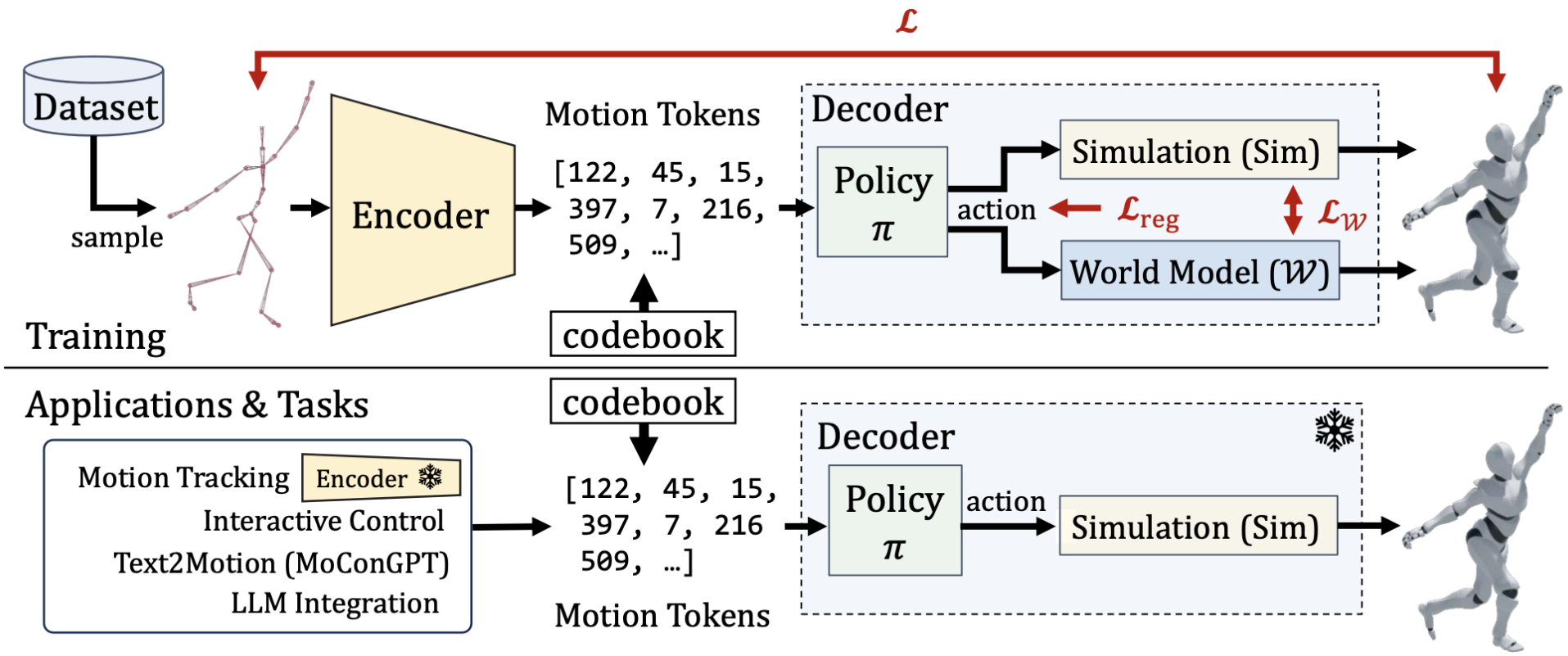
# Background – AutoEncoder




<VQ-VAE>

# 03 Method

# Method – Overview




<System Overview>

# Method

Overview : V = E(M), Z = B(V), $\widetilde{M} = D(Z)$

motion Clip M : $[S^t]$

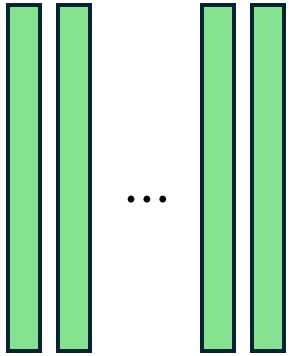
Encoder : $E$

Discrete codebook : B

Decoder : D

Simulation(Sim)

World Model




<codebook>

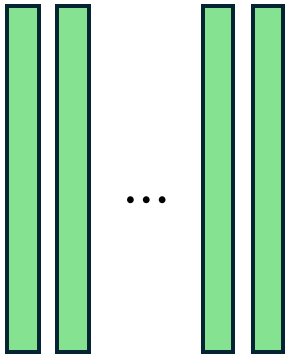
A list of learnable latent codes which is used to quantize the latent representation

latent codes : $\{z_i\}_{i=1}^{N_B}, z_i \in Z$, quantized version of V

$$\boldsymbol{z}^k, I^k = \arg\min_{\boldsymbol{z_i} \in \mathcal{B}} \|\boldsymbol{z_i} - \boldsymbol{v}^k\|_2^2$$

# Method

**VQ-VAE**

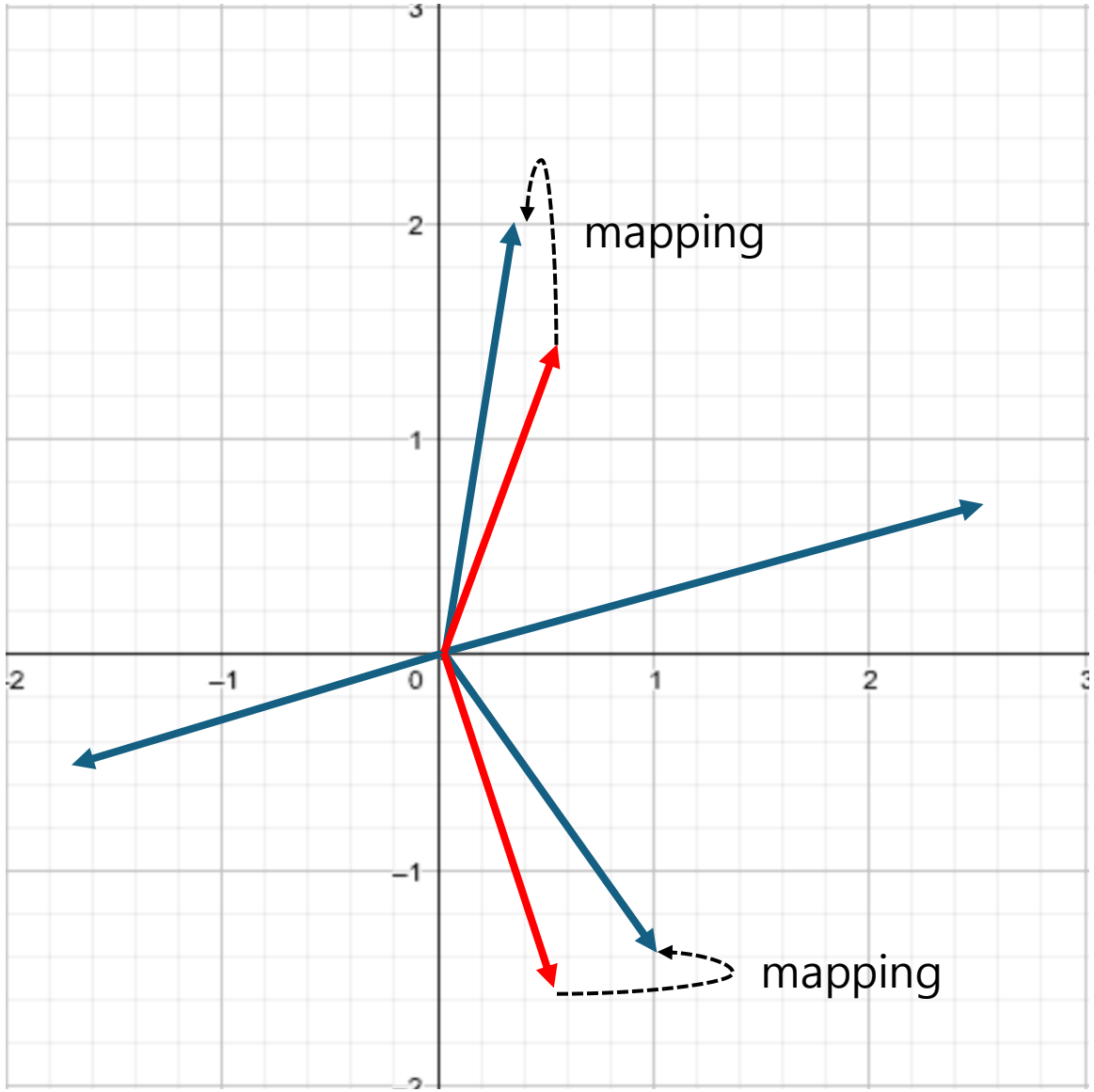



<codebook>

$$z^k, I^k = \arg\min_{z_i \in \mathcal{B}} \|z_i - v^k\|_2^2$$

quantized Vector

V = E(M)



# Method

**Simulation(Sim)**

PhysX(Physics-engine)

# Method

**Environment**

<Simulation> : Explicit Physical Law Model

**Key Characteristics**

- Physical Accuracy
- High-Precision Calculations
- Based on fixed physical laws

**dis-Advantage**

- Computational Cost

<World Model> : Date-Driven Approximation Model

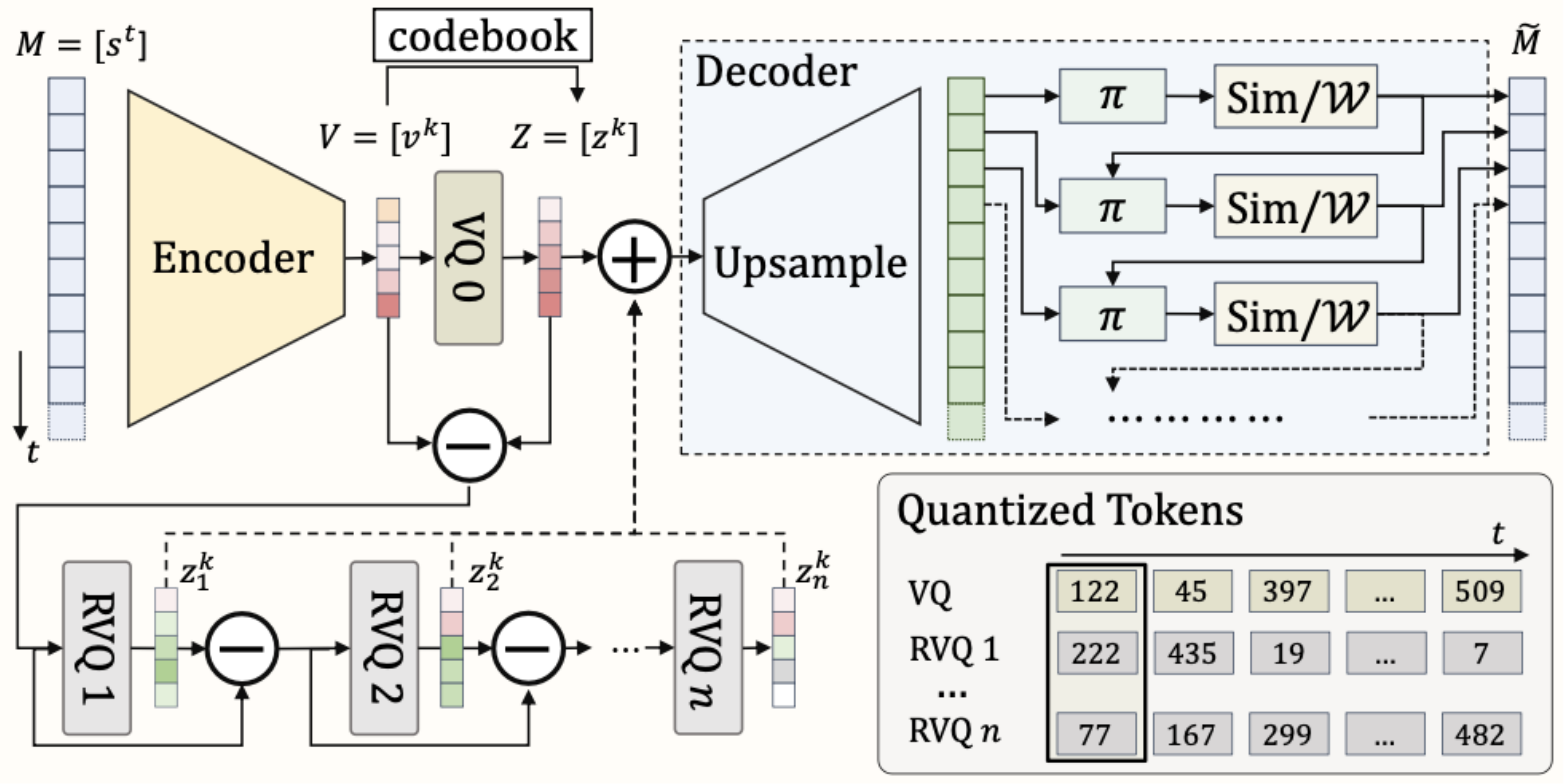
**Key Characteristics**

- Predictive Model
- Differentiability
- trainable model

**dis-Advantage**

- Approximation

# Method

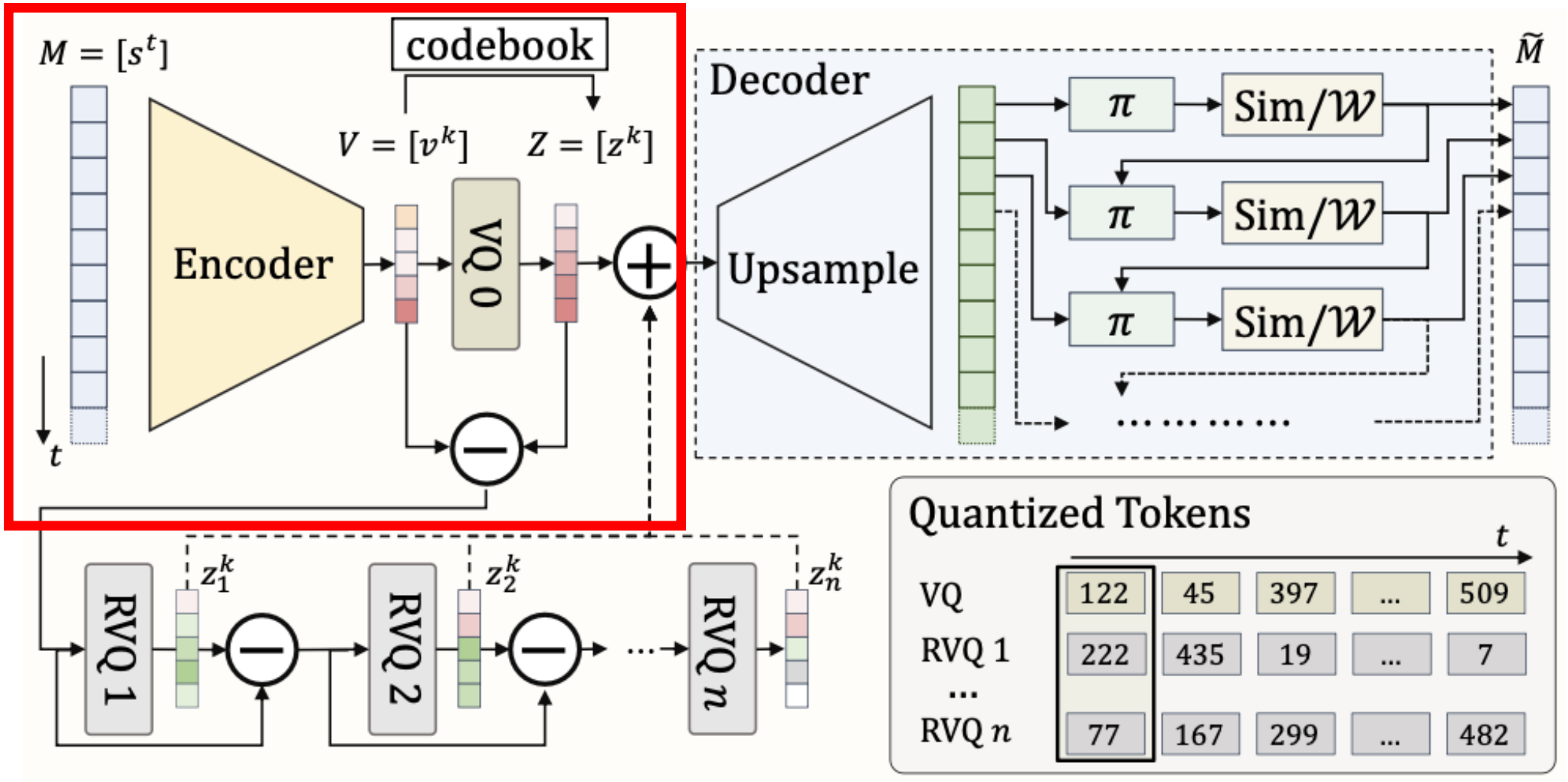



<MoConVQ Architecture>

**<Elements of MoConVQ>**

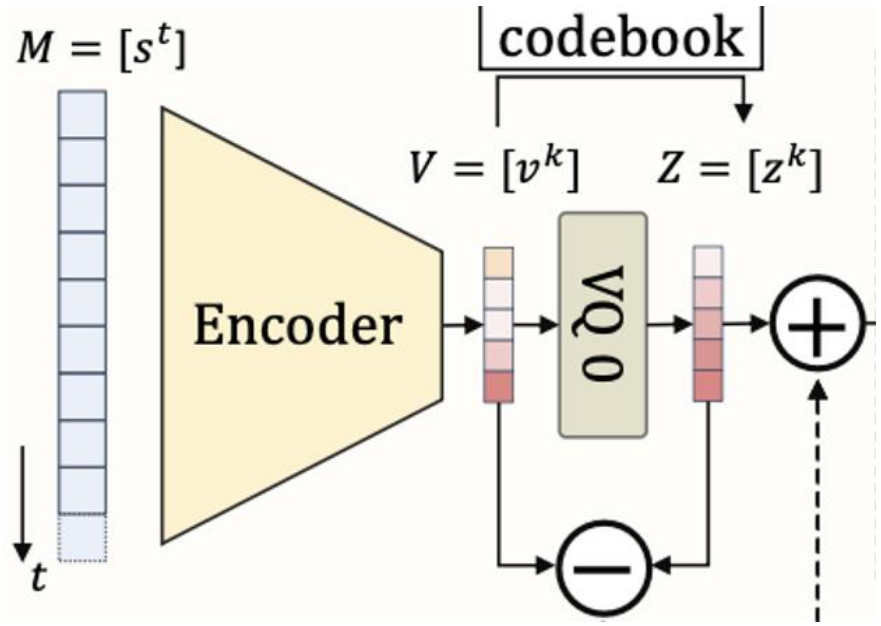
- **motion Encoder**
- Physics-based Decoder
- series of codebooks
- Simulator
- World model
- RVQ

# Method – Overview

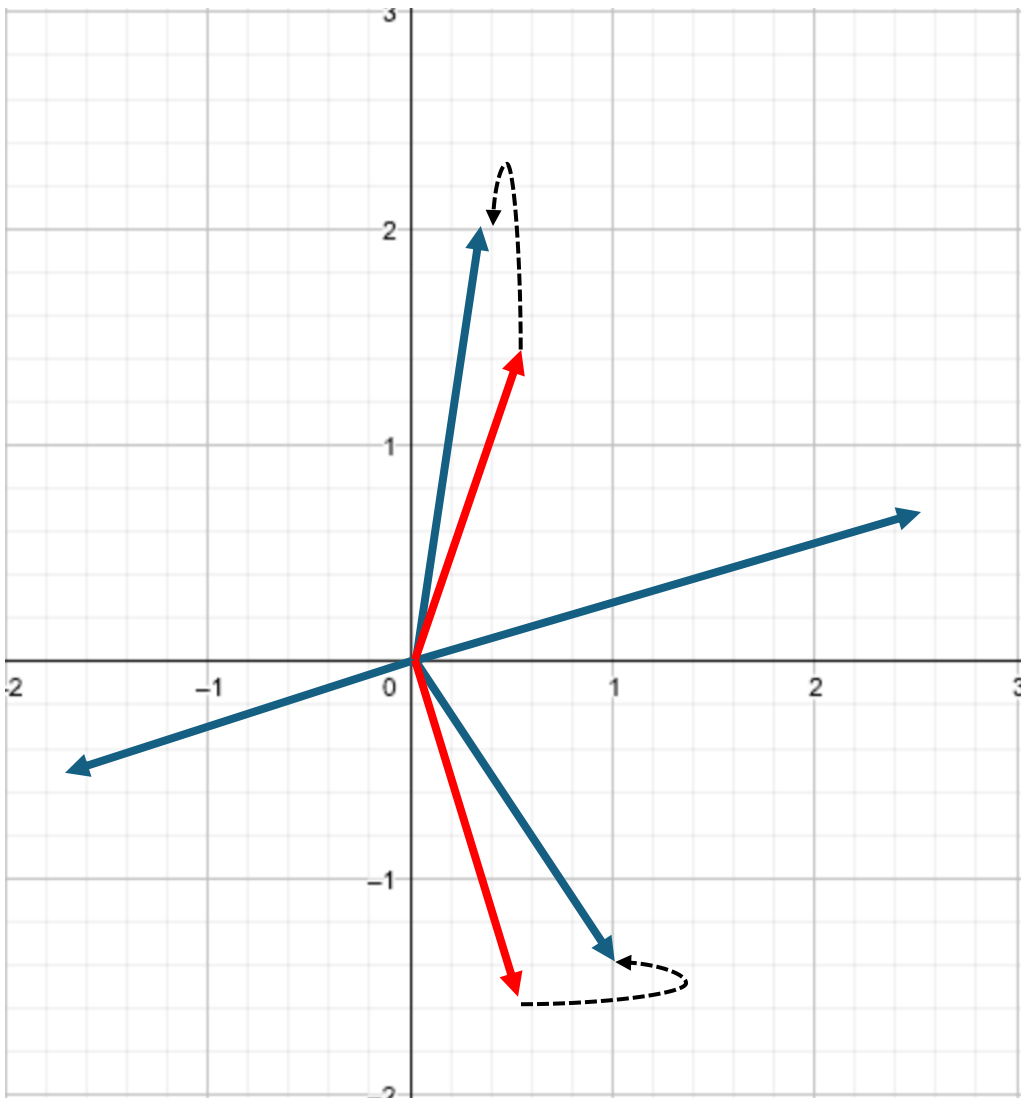


# Method – Motion Encoding Process

**Motion Encoding Process**




<Encoder : 1D CNN Structure>

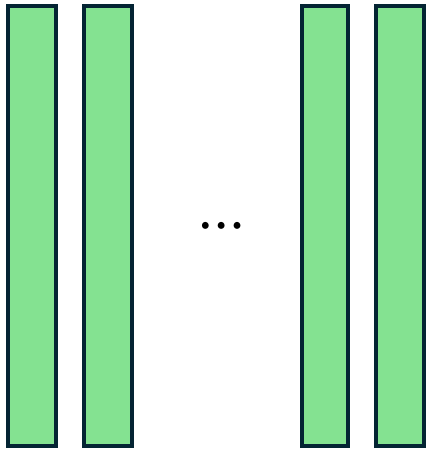



<VQ-VAE>

# Method – Motion Encoding Process

**Motion Encoding Process**

index: 1 2 … k-1 k



<codebook>

$$V = [v^1\ v^2\ v^3\ \dots\ v^{k-1}\ v^k]$$

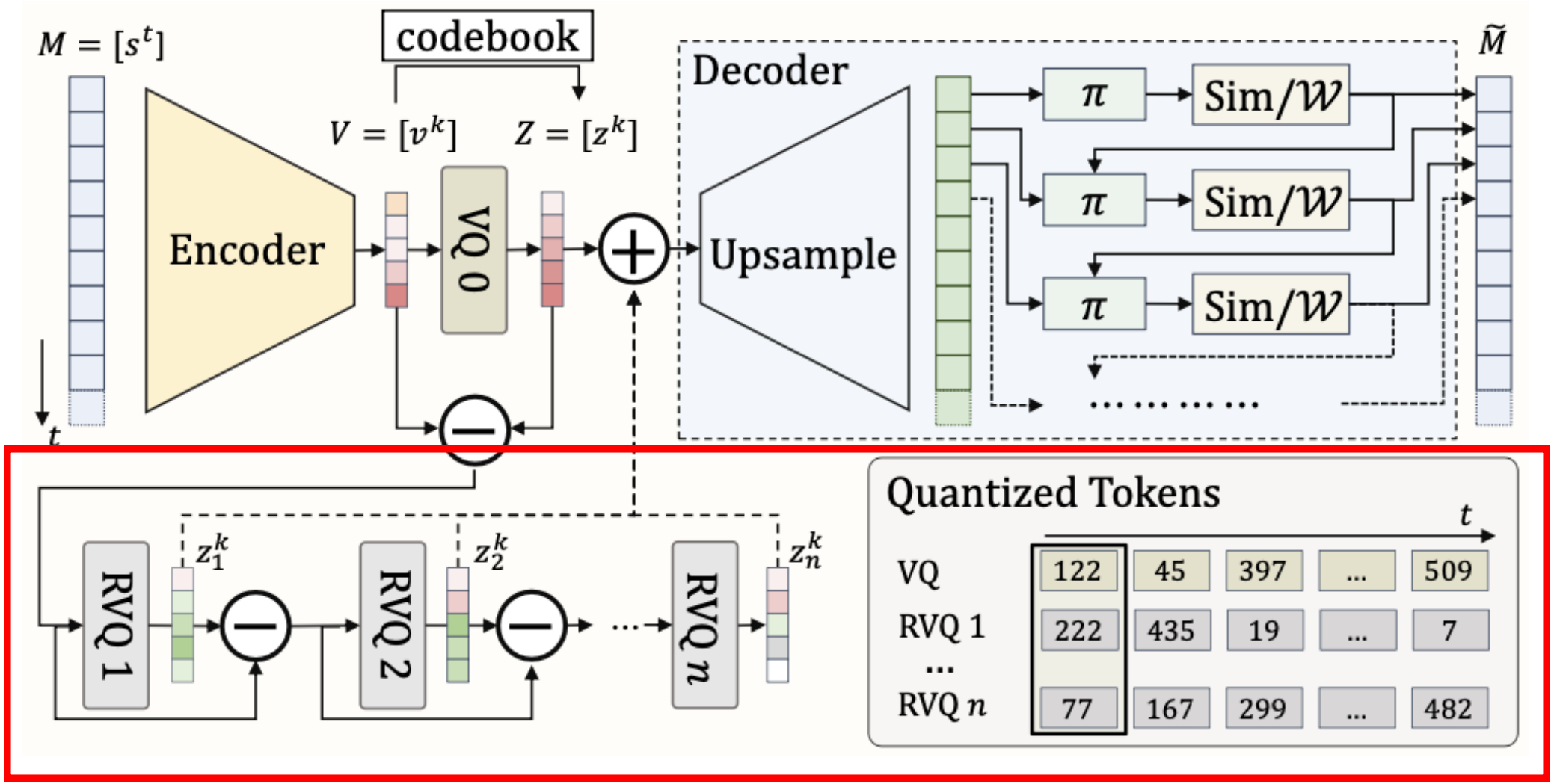
quantization

$$Z = [z^1\ z^2\ z^3\ \dots\ z^{k-1}\ z^k]$$

Z can be represented using its corresponding indices

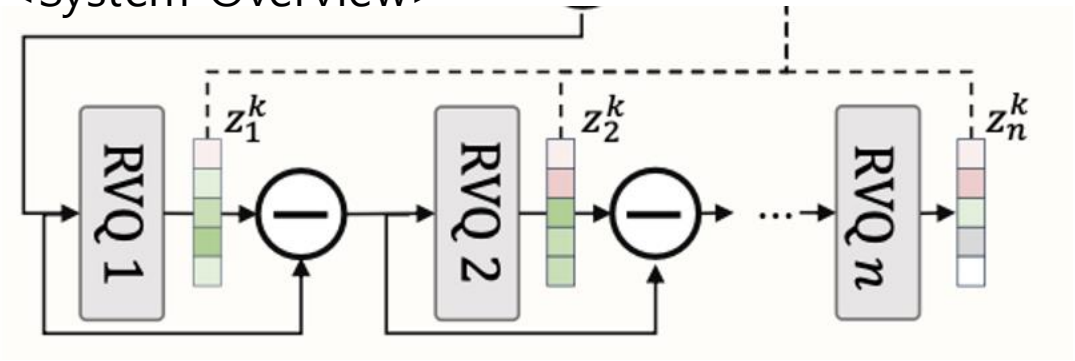
$$S = [I^1\ I^2\ I^3\ \dots\ I^{k-1}\ I^k]$$

# Method


$M = [s^t]$
codebook
$V = [v^k]$
$Z = [z^k]$
Encoder
VQ 0
Decoder
Upsample
$\pi$
Sim/$\mathcal{W}$
$\widetilde{M}$
$t$
RVQ 1
RVQ 2
RVQ $n$
$z_1^k$
$z_2^k$
$z_n^k$
Quantized Tokens
VQ
RVQ 1
RVQ $n$
122 45 397 ... 509
222 435 19 ... 7
77 167 299 ... 482

# Method

<System Overview>




$r_d = v - \sum_{j<d} z_j$
$B_d$: $r_d$ -> $z_d$

RVQ n

$r_1 = v - z$
$B_1$: $r_1$ -> $z_1$

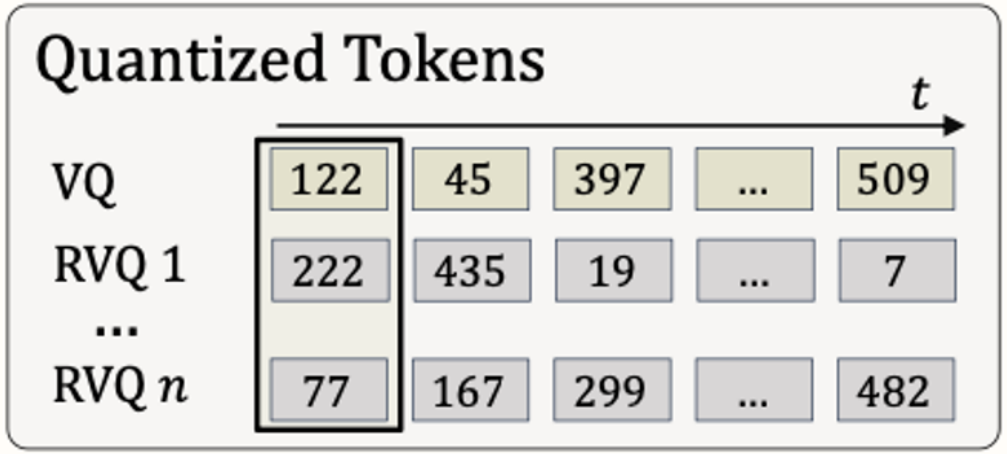
RVQ 1

$r_2 = v - z_1$
$B_2$: $r_2$ -> $z_2$

RVQ 2

$r_3 = v - (z_1 + z_2)$
$B_2$: $r_3$ -> $z_3$

RVQ 3



# Method

**ALGORITHM 1:** Algorithm of Residual Vector Quantization

**Function Residual Vector Quantization**$(\boldsymbol{v})$:

```
// v ∈ V is the continuous latent code
// computed using Equation (1)
```

$\boldsymbol{z} \leftarrow 0$

$\boldsymbol{r} \leftarrow \boldsymbol{v}$

**for** $d \leftarrow 0$ **to** $N$ **do**

  $\boldsymbol{z}_d, I_d \leftarrow \arg\min_{\boldsymbol{z}_i \in \mathcal{B}_d} \|\boldsymbol{r} - \boldsymbol{z}_i\|_2^2$

  $\boldsymbol{r} \mathrel{-}= \boldsymbol{z}_d$

  $\boldsymbol{z} \mathrel{+}= \boldsymbol{z}_d$

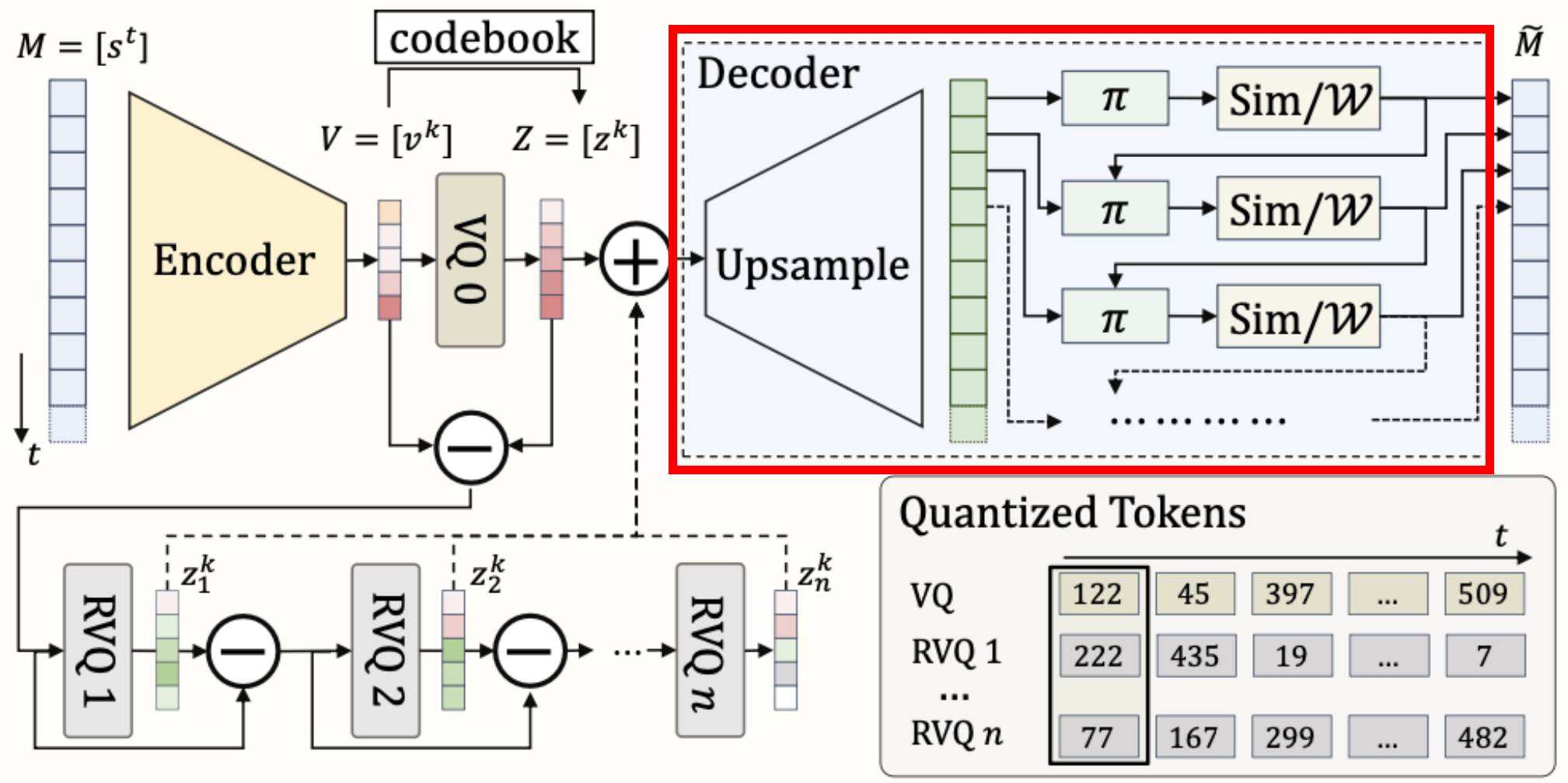
**end**

$I \leftarrow (I_0, I_1, \ldots, I_N)$

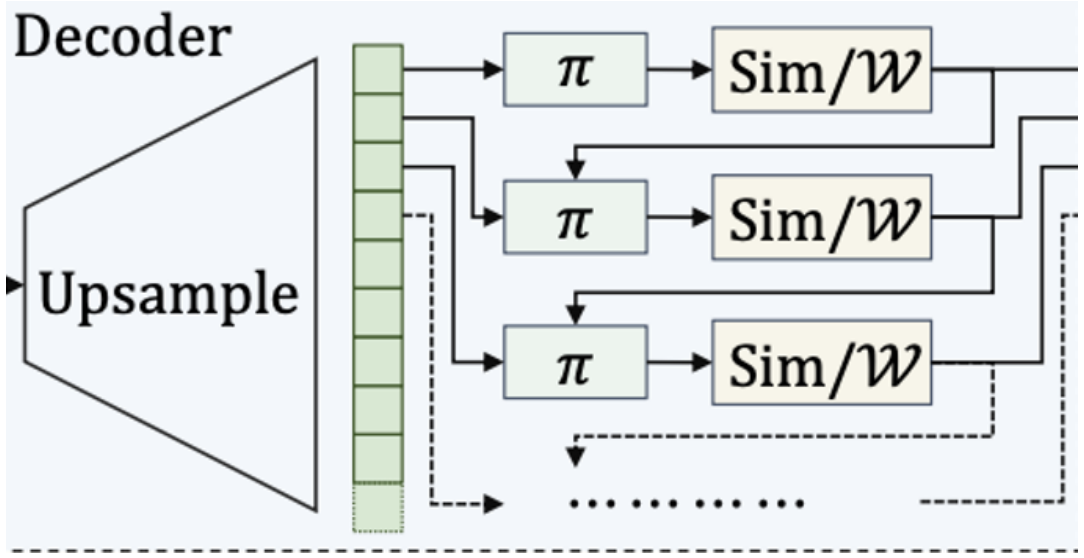
**return** $\boldsymbol{z}, I$

**end**

# Method


$M = [s^t]$
codebook
$V = [v^k]$
$Z = [z^k]$
Encoder
VQ 0
Decoder
Upsample
$\pi$
Sim/$\mathcal{W}$
$\tilde{M}$
$t$
RVQ 1
RVQ 2
RVQ $n$
$z_1^k$
$z_2^k$
$z_n^k$
Quantized Tokens
VQ
RVQ 1
RVQ $n$
122 45 397 ... 509
222 435 19 ... 7
77 167 299 ... 482

# Method




**Components**

- Decoder(Upsampling)
- Policy
- Simulation(Sim)
- World Model(W)

**simulation**

- treat the simulation as a black box
- framework independent of specific physics engines
- it makes the decoder non-differentiable

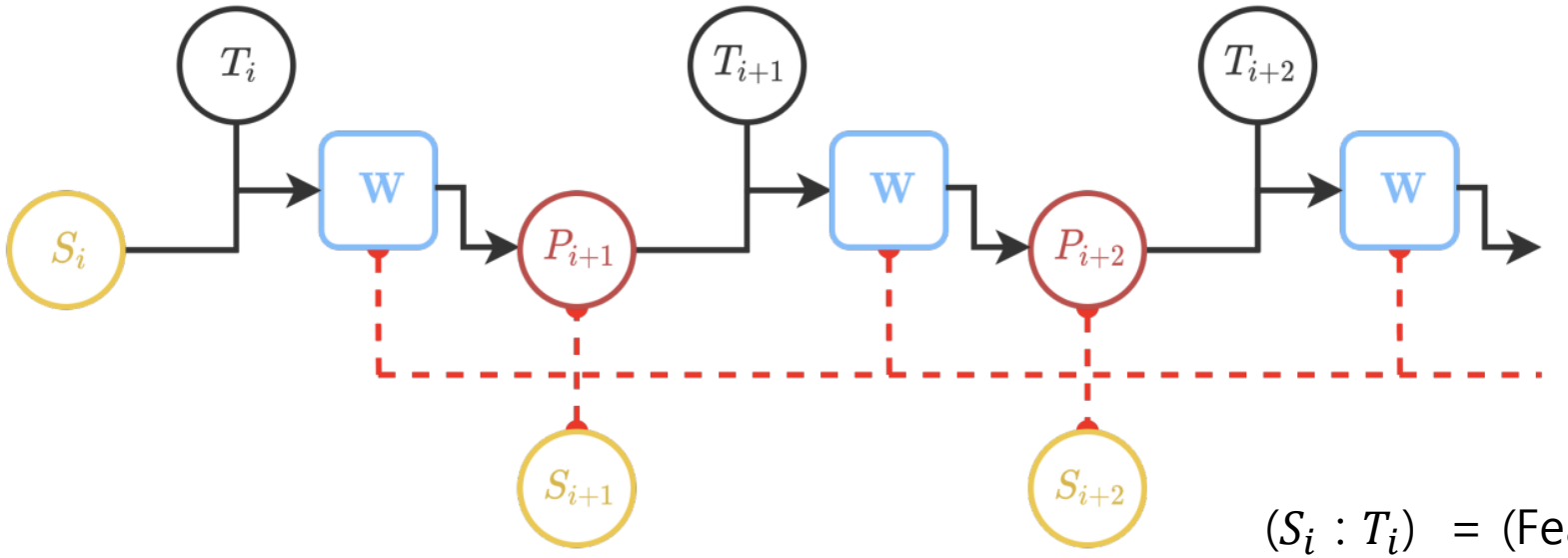
**World Model**

- For model-based reinforcement learning
- approximate the black box simulator

# Method

**Simulation → World Model : Approximation**

Inspired by…
SuperTrack: motion tracking for physically simulated characters using supervised learning




- W : world model
- $P_i$ : predicted State
- $S_i$ : GT state
- $T_i$ : the policy PD-target output (sampled action from Simulation)

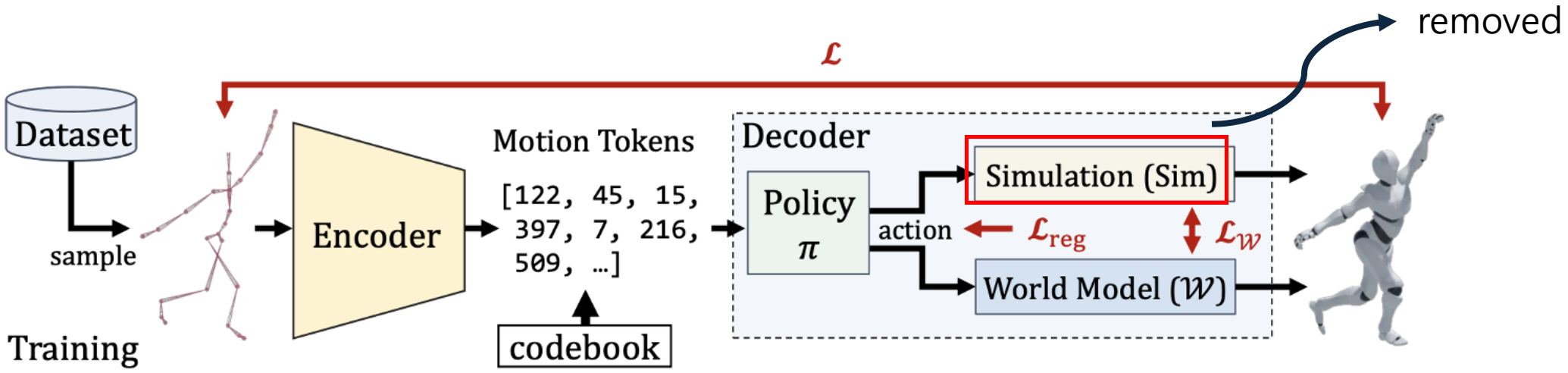
$(S_i : T_i)$ = (Features : label) -> Supervised Learning

$$\mathcal{L}_{\mathcal{W}} = \|\tilde{M}_{\text{sim}} - \tilde{M}_{\mathcal{W}}\|$$

Levi Fussell, Kevin Bergamin, and Daniel Holden. 2021. SuperTrack: motion tracking for physically simulated characters using supervised learning. ACM Trans. Graph. 40, 6, Article 197 (December 2021), 13 pages. https://doi.org/10.1145/3478513.3480527

# Method

**Training**




$$\hat{\boldsymbol{a}}^t = \boldsymbol{\pi}(\hat{\boldsymbol{s}}^t, \boldsymbol{u}^t), \quad \hat{\boldsymbol{s}}^{t+1} = \mathcal{W}(\hat{\boldsymbol{s}}^t, \hat{\boldsymbol{a}}^t)$$

$$\begin{aligned}\mathcal{L} = &\|\boldsymbol{M} - \tilde{\boldsymbol{M}}_{\mathcal{W}}\| \\ &+ \beta_1 \|\mathcal{E}(\boldsymbol{M}) - \text{sg}(\boldsymbol{Z})\| + \beta_2 \|\text{sg}(\mathcal{E}(\boldsymbol{M})) - \boldsymbol{Z}\| \\ &+ \beta_3 \mathcal{L}_{\text{reg}}\end{aligned}$$

# Method

**Loss function**

$$\mathcal{L} = \boxed{\|\boldsymbol{M} - \tilde{\boldsymbol{M}}_{\mathcal{W}}\|}$$
$$+ \boxed{\beta_1 \|\mathcal{E}(\boldsymbol{M}) - \mathrm{sg}(\boldsymbol{Z})\| + \beta_2 \|\mathrm{sg}(\mathcal{E}(\boldsymbol{M})) - \boldsymbol{Z}\|}$$
$$+ \boxed{\beta_3 \mathcal{L}_{\mathrm{reg}}}$$

Standard VQ-VAE Losses

Regularization term for action

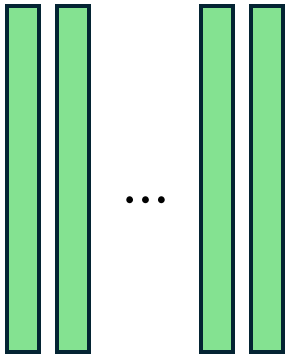
sg : stop gradient operator

# Method – Overview

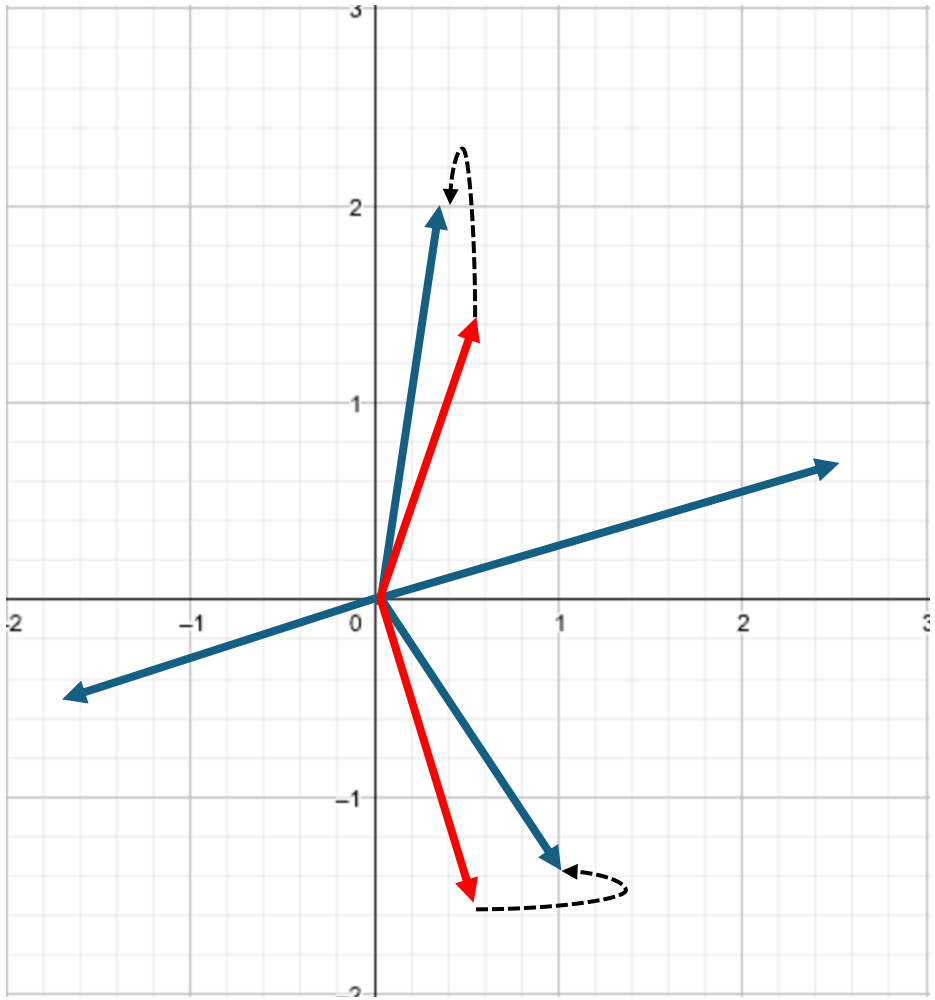
**Updating the codebook B**

EMA Update(Exponential Moving Average)

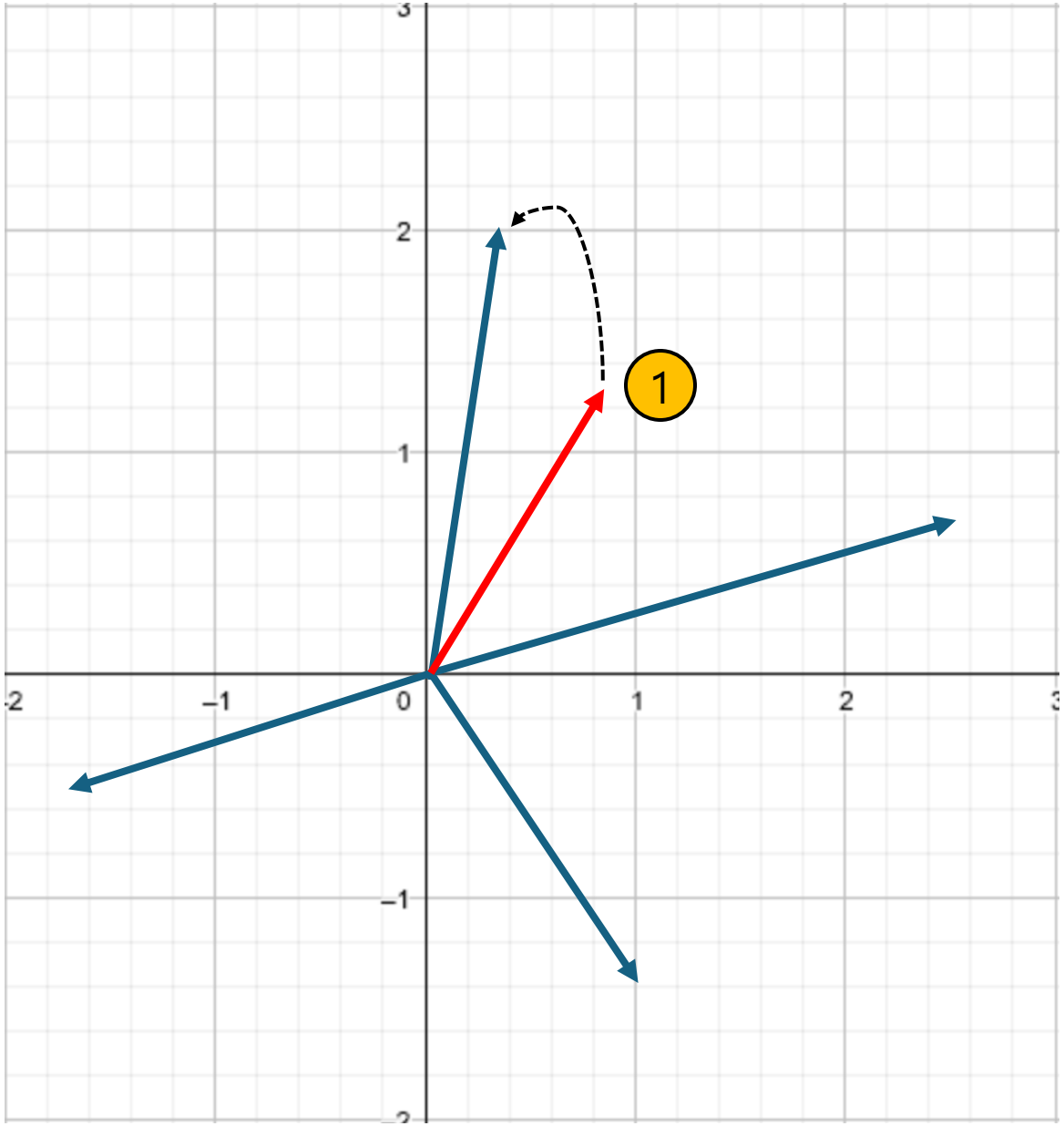
$$\mathcal{B} \leftarrow (1-\gamma)\mathcal{B} + \gamma\hat{\mathcal{B}}$$



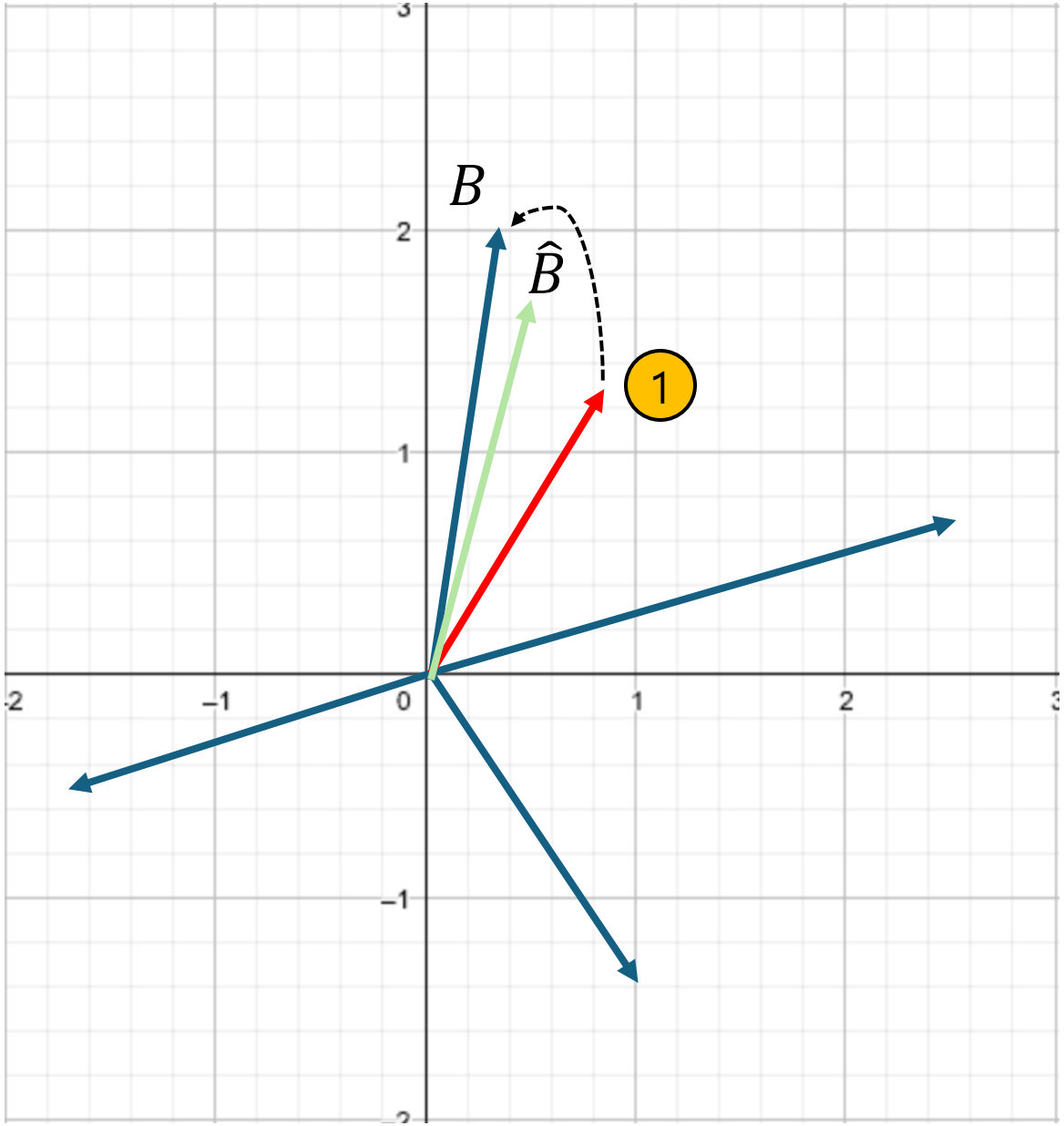

<codebook>



# Method – Overview

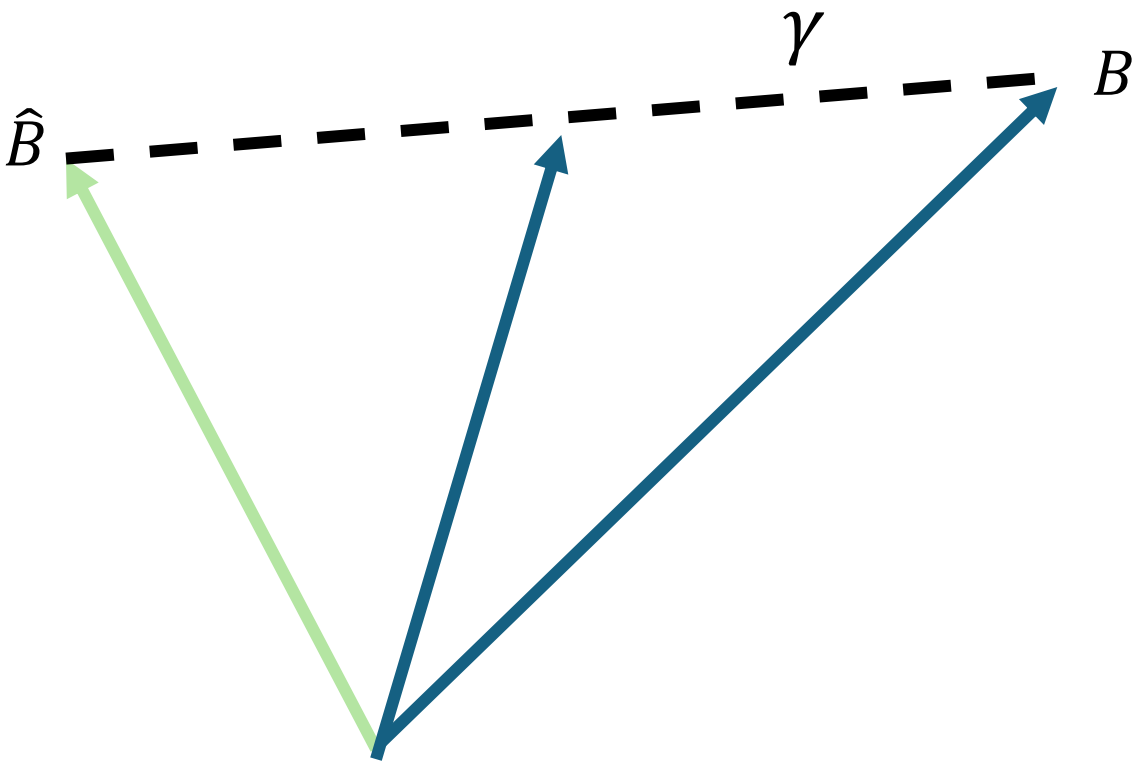


# Method – Overview



# Method

$$\mathcal{B} \leftarrow (1-\gamma)\mathcal{B} + \gamma\hat{\mathcal{B}}$$



# Method

**Regularization term**

$$\bar{\boldsymbol{a}}^t = (1-\beta)\bar{\boldsymbol{a}}^{t-1} + \beta\hat{\boldsymbol{a}}^t$$

EMA of action(soft constraint)

$$\mathcal{L}_{\text{reg}} = \sum_t w_1 \|\hat{\boldsymbol{a}}^t - \bar{\boldsymbol{a}}_t\| + w_2 \|\hat{\boldsymbol{a}}^t\|$$

$\bar{a}^t$ : EMA of action

$\hat{a}^t$ : action from policy

# Method

**properties of State & Action**

**Action: PD Control**

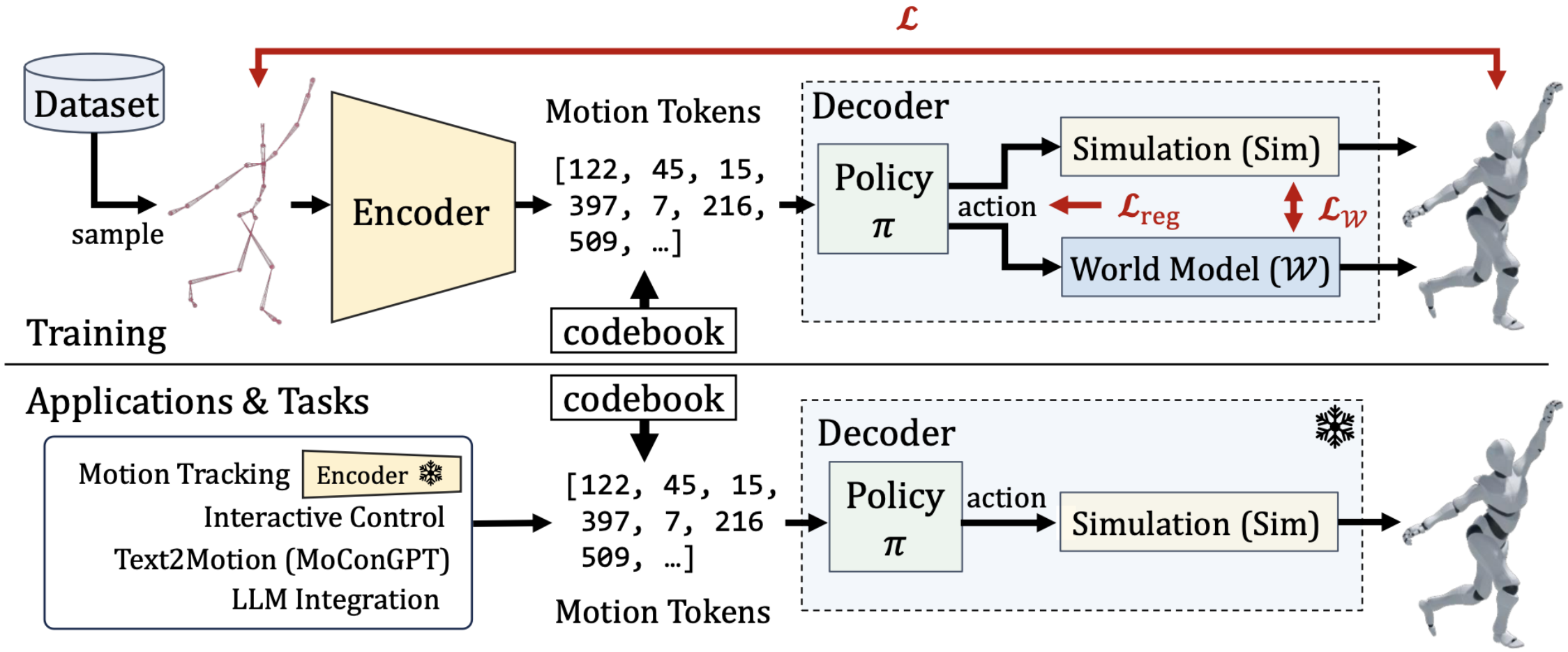
$\bar{\theta}$: desired joint rotation -> output of policy
$\theta$: current joint rotation

$$\tau = k_p(\bar{\theta} - \theta) - k_d\dot{\theta}$$

**State**

- r: rotation
- p: position
- v: velocity
- w: angular velocity
- h: height
- $y_0$: Up direction of the root joint

# Method


$\mathcal{L}$
Dataset
sample
Encoder
Motion Tokens
[122, 45, 15,
397, 7, 216,
509, ...]
codebook
Decoder
Policy
$\pi$
action
Simulation (Sim)
$\mathcal{L}_{\text{reg}}$
$\mathcal{L}_{\mathcal{W}}$
World Model ($\mathcal{W}$)
Training
Applications & Tasks
Motion Tracking
Encoder
Interactive Control
Text2Motion (MoConGPT)
LLM Integration
codebook
[122, 45, 15,
397, 7, 216
509, ...]
Motion Tokens
Decoder
Policy
$\pi$
action
Simulation (Sim)

# 04 Experiments

# Experiments

| Dataset | Length (min) | Motions |
|---|---|---|
| SFU | 10.90 | Balance, Dance, Kendo, Skip, ... |
| ACCAD | 14.61 | Run, Swagger, Punch, Turn, ... |
| BMLmovi | 103.32 | Walk, Stretch, Throw, Catch, ... |
| BMLrub | 180.52 | Lift, Sit, Kick, Throw, ... |
| CMU | 375.24 | Sports, Dance, Pantomime, ... |
| KIT | 392.29 | Wipe, Bend, Parkour, Drink, ... |
| LaFAN | 213.14 | Walk, Dance, Aiming, Fight, ... |

<DataSet>

| Method | Motion Length |
|---|---|
| AMP [2021] | 7 minutes |
| ASE [2022] | 30 minutes |
| PhysicsVAE [2022] | 10 minutes |
| ControlVAE [2022] | 30 minutes |
| NCP [2023] | 26 ~ 30 minutes |
| Ours | 23.2 hours |

<DataSet size>

# Experiments

Sim : ODE(customed Open Dynamics Engine)

GPU : A Single RTX 3090

CPU : Intel® Xeon E5-2678 v3 @2.50Hz

motion length : 24 frames

K(Latent vector) : 6

codebooks : 8(VQ, RVQ7)

|| B || = 512

epochs : 40k

training time : 6 days

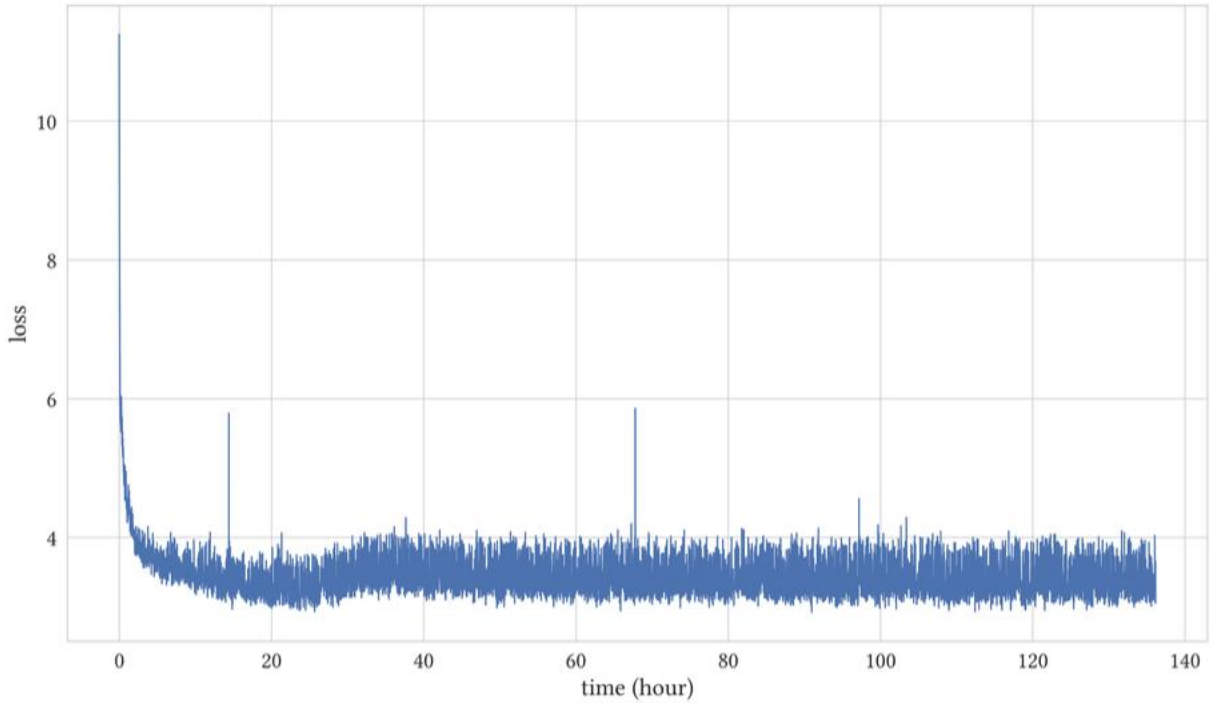



Fig. 6. A typical learning curve of MoConVQ.

# 04 Evaluation

# Evaluation

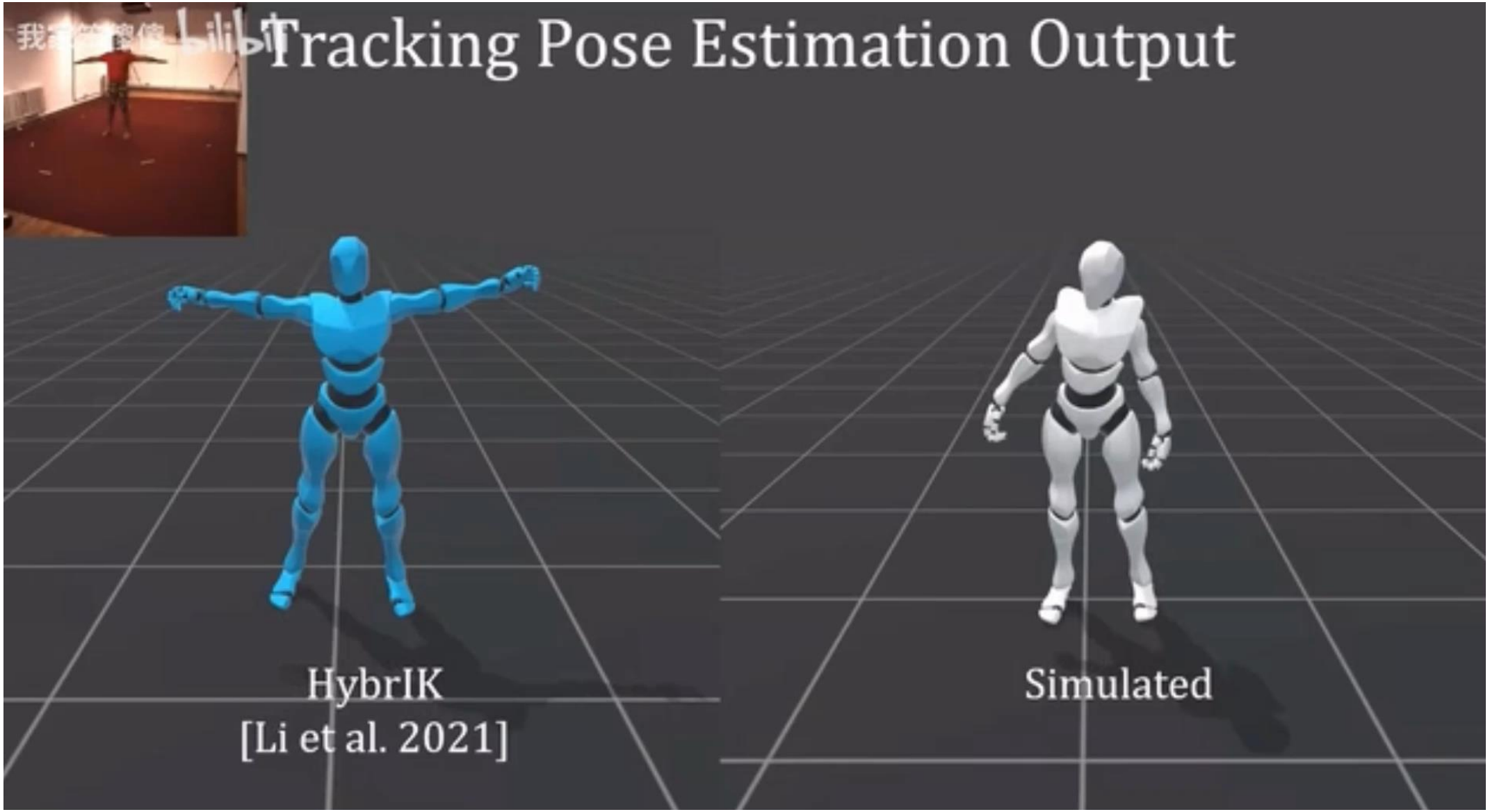
| Methods | Physics | MPJPE ↓ (mm) | PA-MPJPE ↓ (mm) |
|---|---|---|---|
| HybrIK | none | 54.4 | 34.5 |
| PhysCap | approx | 97.4 | 65.1 |
| SimPoE | approx | 56.7 | 41.6 |
| SimPoE (w/o ResForce) | full | 115.2 | 65.1 |
| Ours | full | 125.6 | 69.3 |

# Evaluation

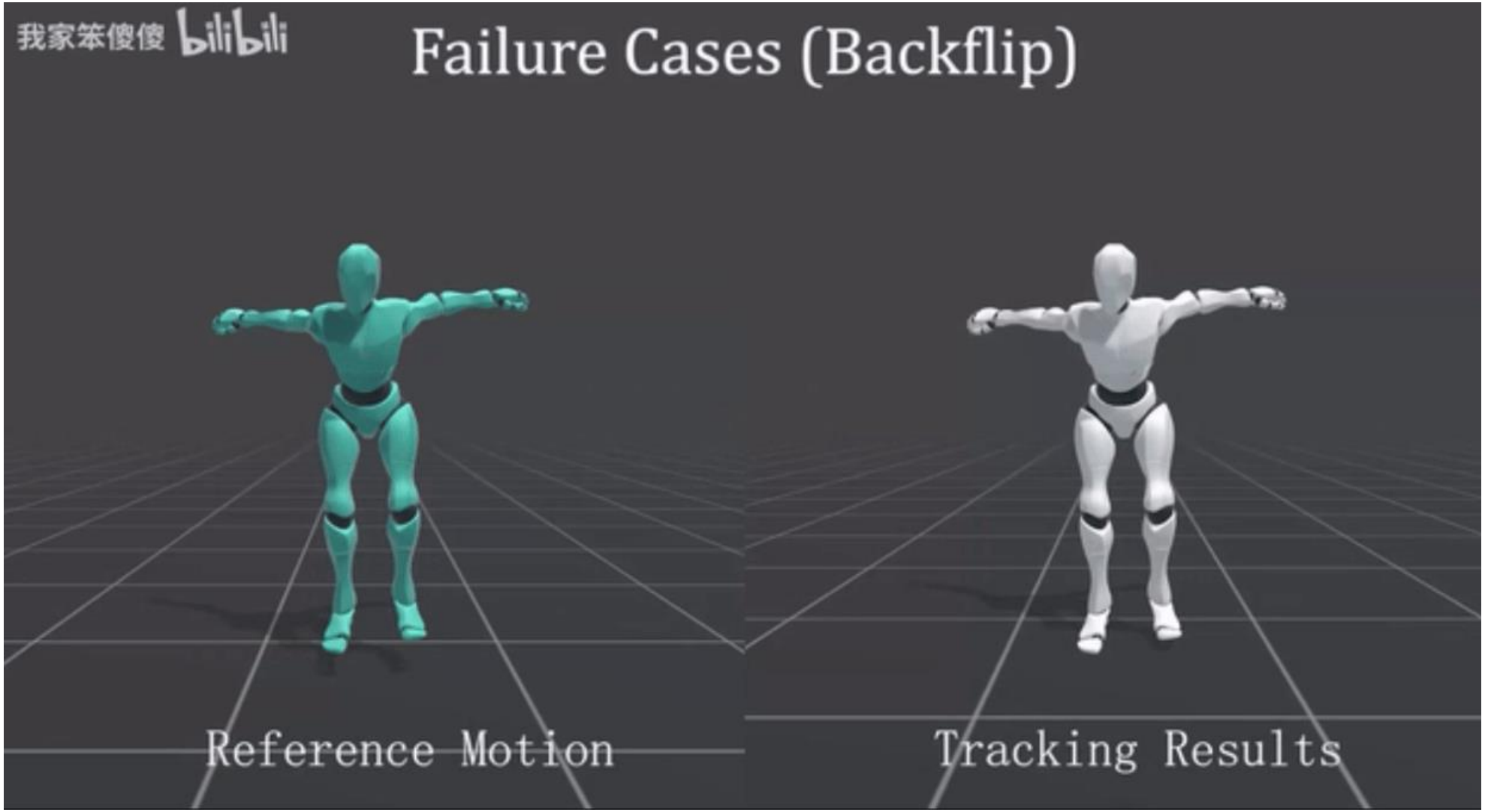
| Methods | $e_{\text{smooth}} \downarrow$ | $\sigma_{\text{smooth}} \downarrow$ | Accel $\downarrow$ | FS $\downarrow$ | GP $\downarrow$ |
|---|---|---|---|---|---|
| HybrIK | 5.9 | 3.1 | 10.9 | - | - |
| PhysCap | 7.2 | 6.9 | - | - | - |
| SimPoE | - | - | 6.7 | 3.4 | 1.6 |
| SimPoE (w/o ResForce) | - | - | 23.5 | 6.1 | 3.2 |
| Ours | **3.4** | **2.9** | **5.1** | **0.87** | **0.29** |

FS: Foot Skating
GP: Ground Penetration
w/o ResForce : without Residual Force

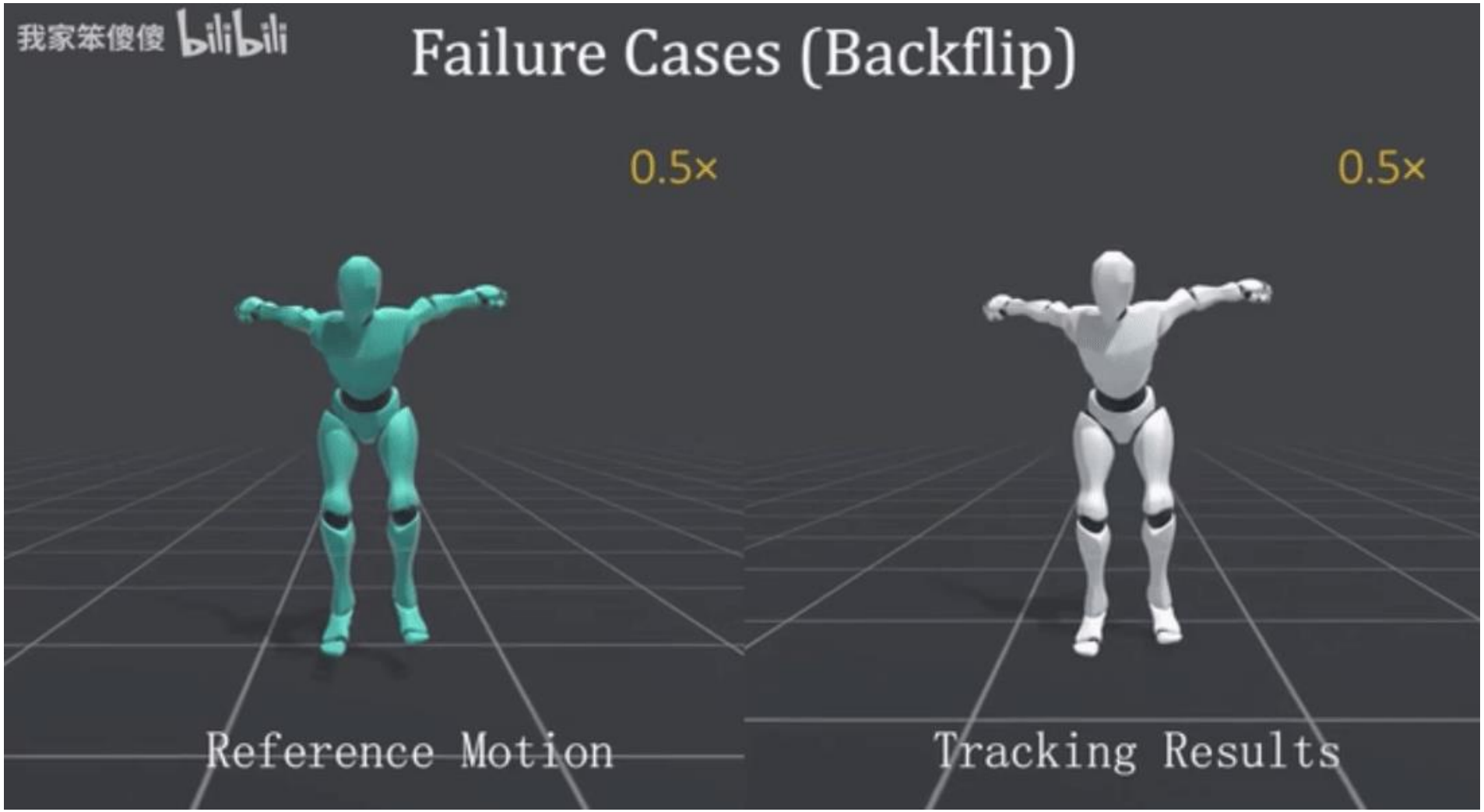
# Evaluation


Tracking Pose Estimation Output
HybrIK
[Li et al. 2021]
Simulated

# Evaluation


我家笨傻傻 bilibili
Failure Cases (Backflip)
Reference Motion
Tracking Results

# Evaluation


我家笨傻傻 bilibili
Failure Cases (Backflip)
0.5×
0.5×
Reference Motion
Tracking Results

# 05 Reference

# Reference


- A. van den Oord, O. Vinyals, and K. Kavukcuoglu, Neural Discrete Representation Learning. 2018. [Online]. Available: https://arxiv.org/abs/1711.00937
- Jungdam Won, Deepak Gopinath, and Jessica Hodgins. 2022. Physics-based character controllers using conditional VAEs. ACM Trans. Graph. 41, 4, Article 96 (July 2022), 12 pages. https://doi.org/10.1145/3528223.3530067
- Levi Fussell, Kevin Bergamin, and Daniel Holden. 2021. SuperTrack: motion tracking for physically simulated characters using supervised learning. ACM Trans. Graph. 40, 6, Article 197 (December 2021), 13 pages. https://doi.org/10.1145/3478513.3480527
- https://www.bilibili.com/video/BV13D421N7zp/
- Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2017. Neural discrete representation learning. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates Inc., Red Hook, NY, USA, 6309–6318.