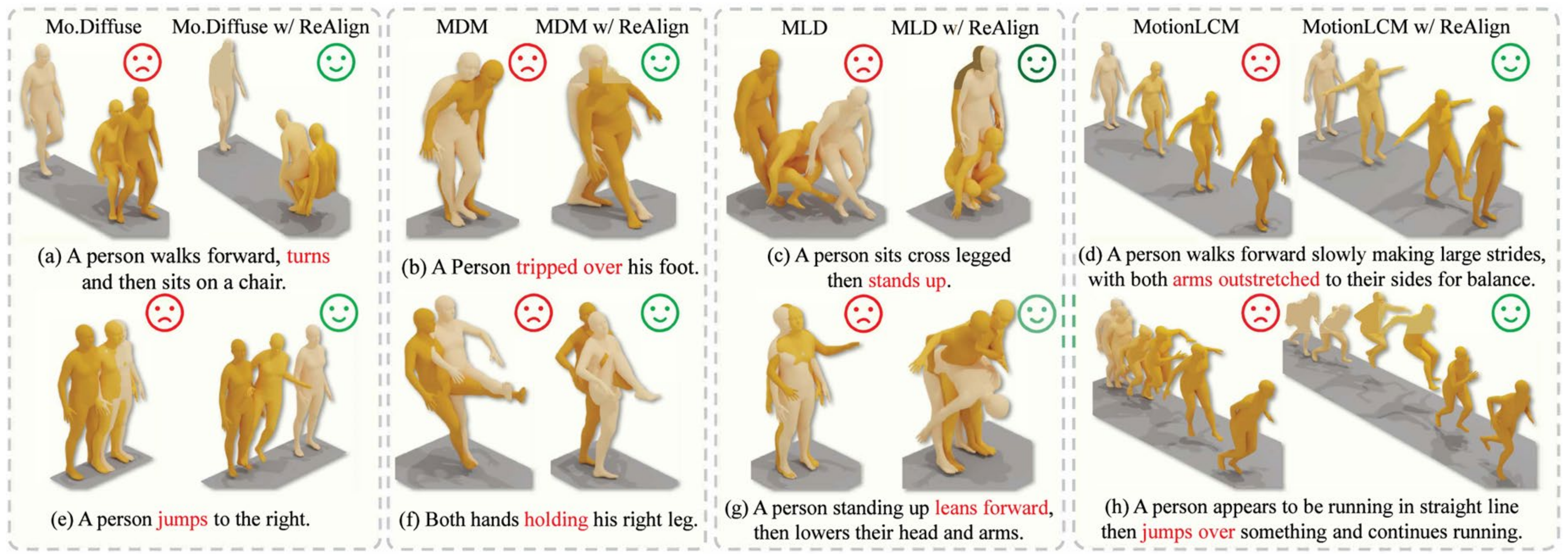
ReAlign: Text-to-Motion Generation via Step-Aware Reward-Guided Alignment

Association for the Advancement of Artificial Intelligence

Wanjiang Weng[1,2,*], Xiaofeng Tan[1,2,*], Junbo Wang[3], Guo-Sen Xie[4], Pan Zhou[5], Hongsong Wang[1,2,†]
[1]Southeast University, [2]PALM Lab, [3]Northwest Polytechnical University,
[4]Nanjing University of Science and Technology, [5]Singapore Management University

Introduction

Issue: Existing text-to-motion methods suffer from semantic misalignment and fail to rectify the denoising process.



Key Observation: We observe that diffusion-based text to motion generation accumulates text-motion misalignment during denoising.

Question: Can we *improve text–motion alignment* and realism *at inference time?*

Contributions: We propose ReAlign, a plug and play reward guided sampling method with a step aware reward model to improve text motion alignment and motion realism during denoising without fine tuning the diffusion model.

Motivation: Diffusion based text to motion models can generate realistic motions but often drift from the text because sampling favors high probability regions and image trained text encoders do not capture motion dynamics. Since paired motion text data is limited, we guide the sampling process with a reward distribution learned from available pairs to improve alignment without fine tuning the diffusion model.

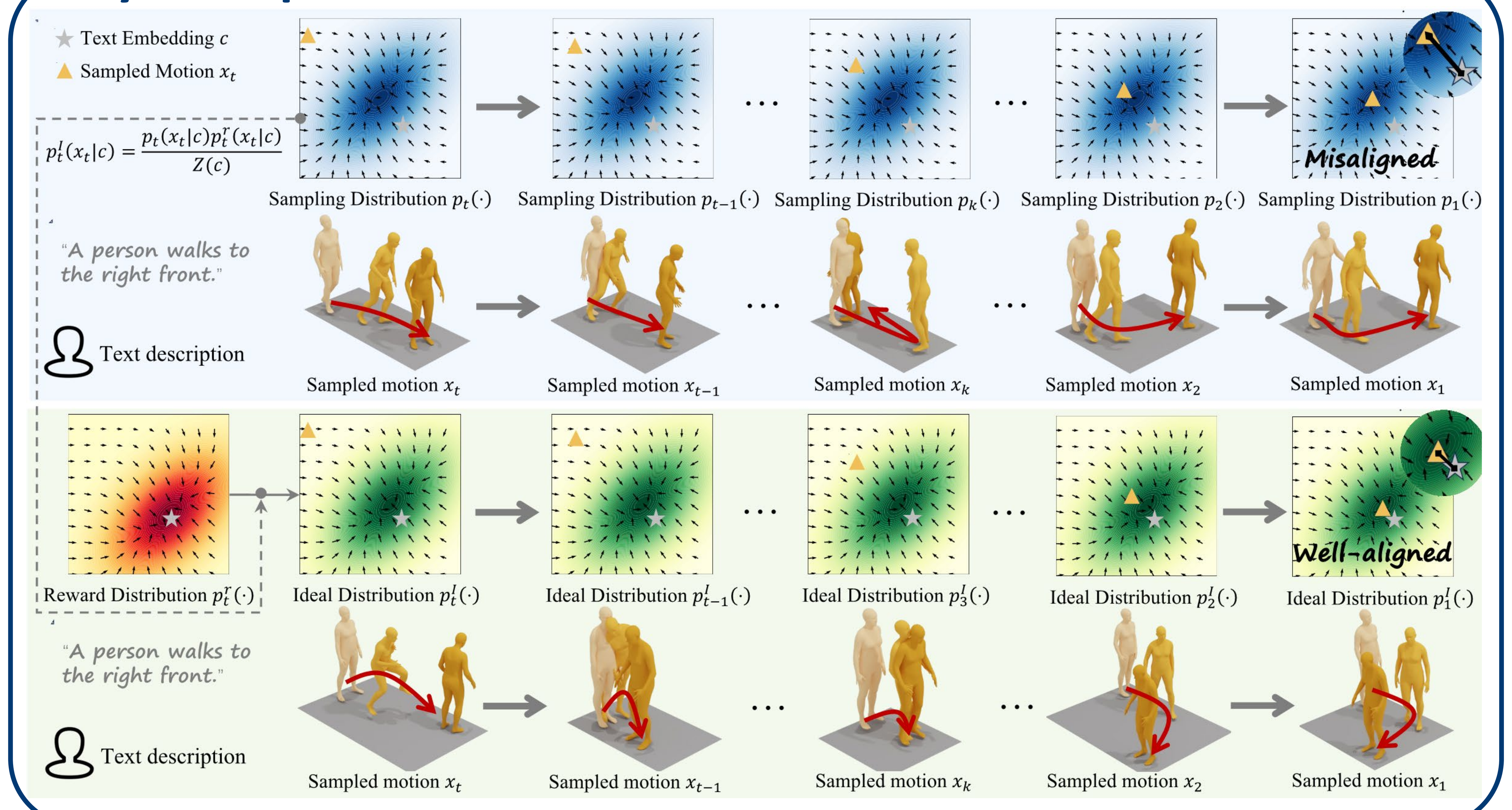
Contact

Email: wjweng@seu.edu.cn
txf0620@gmail.com

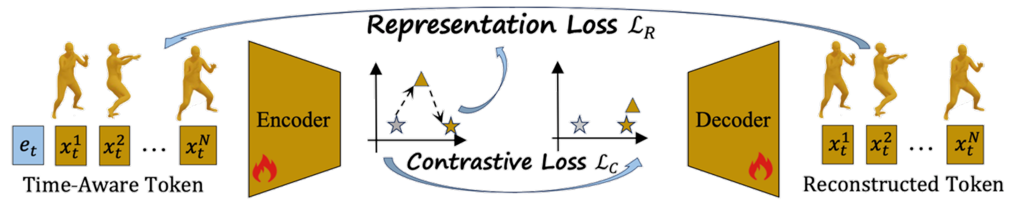
WeChat: wengwanjiang
txf_06_20

Author Project Page Code Paper

Toy Example



Reward-Guided Sampling



Algorithm 1: Training Step-Aware Reward Model

Input: Step-aware reward model R_φ, training set $\mathcal{D}_{tr}$, timestep T range $[t_{\min}, t_{\max}]$, probability parameter ω, noise scheduler α.
Output: Step-aware reward model R_φ.

```
repeat
for (x, c) in D_tr do
    t ← 0                                              ▷ Initialize t
    if Uniform(0,1) > ω then
        t ← Uniform(t_min, t_max)                      ▷ Add noise to motion
    end if
    x_t ~ N(√α_t x, (1 − α_t) I)                       ▷ Forward process
    L_RM(φ; x_t, c) ← L_C(φ; x_t, c) + L_R(φ; x_t, c)  ▷ Compute loss L_RM
    φ ← φ − ∇_φ L_RM(φ)                                ▷ Update parameter
end for
until converged
```

Reward Distribution. With both the step-aware reward model and the motion-to-motion reward, we define the dual-alignment reward as:

$$R(\mathbf{x}_t, c) = \mu R_\varphi(\mathbf{x}_t, c) + \eta R_m(\mathbf{x}_t, c), \tag{9}$$

where μ and η control the contributions of text-based and

Algorithm 2: Reward-Guided Denoise Process

Input: Diffusion model ϵ_θ, reward model R, training set $\mathcal{D}_{tr}$, condition c, timestep T.
Output: Generated motion $\mathbf{x}_0$.

```
x_T ~ N(0, I)
x^c = arg max_{x ∈ D_tr} R_φ(x, c)
for t = T, ..., 1 do
    use x^c to obtain reward score
    ε ~ N(0, I) if t > 1 else ε = 0
    R(x_t, c) ← μ R_φ(x_t, c) + η R_m(x_t, c)       ▷ Compute Reward R(x_t, c)
    x_{t−1} ← (1/√α_t)(x̄_{t−1} + √β_t ε) + ∇R(x_t, c).
                                                    ▷ Denoise guided by reward model
end for
return x_0
```

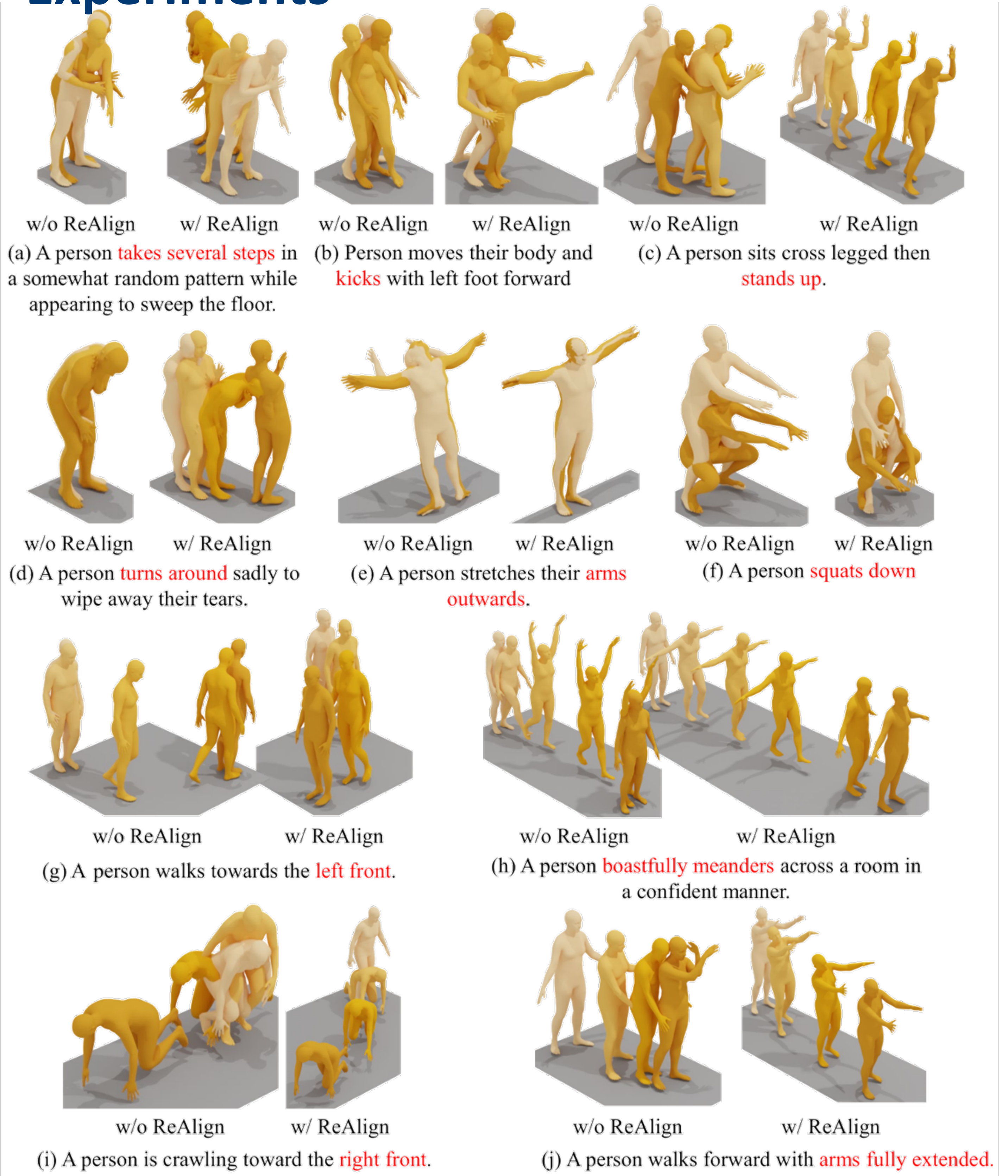
motion-based alignment. This reward formulation defines the reward distribution over noised motion as:

$$p_t^r(\mathbf{x}_t|c) = \exp\left(R(\mathbf{x}_t, c)\right)/Z^r(c). \tag{10}$$

Here, $Z^r(c) = \int \exp(R_\varphi(\mathbf{x}, c))d\mathbf{x}$ is for normalization.

By integrating text-motion and motion-motion alignment, our approach constructs a robust reward signal that captures both semantic consistency and motion coherence. This enables more precise guidance of the diffusion sampling process, ensuring that generated motions are not only probable but also faithful to their textual descriptions.

Experiments



Method	R Precision ↑ Top 1	R Precision ↑ Top 2	R Precision ↑ Top 3	FID ↓	MM Dist ↓	Diversity →
Real	0.511	0.703	0.797	0.002	2.974	9.503
T2M (2022a)	$0.455^{\pm 0.002}$	$0.636^{\pm 0.003}$	$0.736^{\pm 0.003}$	$1.087^{\pm 0.002}$	$3.347^{\pm 0.008}$	$9.175^{\pm 0.002}$
MDM (2023)	$0.455^{\pm 0.006}$	$0.645^{\pm 0.007}$	$0.749^{\pm 0.006}$	$0.489^{\pm 0.047}$	$3.330^{\pm 0.25}$	$9.920^{\pm 0.083}$
T2M-GPT (2023a)	$0.492^{\pm 0.003}$	$0.679^{\pm 0.002}$	$0.775^{\pm 0.002}$	$0.141^{\pm 0.005}$	$3.121^{\pm 0.009}$	$9.722^{\pm 0.082}$
ReMoDiffuse (2023b)	$0.510^{\pm 0.005}$	$0.698^{\pm 0.006}$	$0.795^{\pm 0.004}$	$0.103^{\pm 0.004}$	$2.974^{\pm 0.016}$	$9.018^{\pm 0.75}$
Mo.Diffuse (2024)	$0.491^{\pm 0.001}$	$0.681^{\pm 0.001}$	$0.775^{\pm 0.001}$	$0.630^{\pm 0.001}$	$3.113^{\pm 0.001}$	$9.410^{\pm 0.049}$
OMG (2024)	-	-	$0.784^{\pm 0.002}$	$0.381^{\pm 0.008}$	-	$9.657^{\pm 0.085}$
MotionLCM (2025)	$0.502^{\pm 0.003}$	$0.698^{\pm 0.002}$	$0.798^{\pm 0.002}$	$0.304^{\pm 0.012}$	$3.012^{\pm 0.007}$	$9.607^{\pm 0.066}$
Mo.Mamba (2025b)	$0.502^{\pm 0.003}$	$0.693^{\pm 0.002}$	$0.792^{\pm 0.002}$	$0.281^{\pm 0.011}$	$3.060^{\pm 0.000}$	$9.871^{\pm 0.084}$
CoMo (2024)	$0.502^{\pm 0.002}$	$0.692^{\pm 0.007}$	$0.790^{\pm 0.002}$	$0.262^{\pm 0.004}$	$3.032^{\pm 0.015}$	$9.936^{\pm 0.066}$
ParCo (2025)	$0.515^{\pm 0.003}$	$0.706^{\pm 0.003}$	$0.801^{\pm 0.002}$	$0.109^{\pm 0.005}$	$2.927^{\pm 0.008}$	$9.576^{\pm 0.088}$
MARDM (2024)	$0.500^{\pm 0.004}$	$0.695^{\pm 0.003}$	$0.795^{\pm 0.003}$	$0.114^{\pm 0.007}$	-	-
MG-MotionLLM (2025b)	$0.516^{\pm 0.002}$	$0.706^{\pm 0.002}$	$0.802^{\pm 0.003}$	$0.303^{\pm 0.010}$	$2.952^{\pm 0.009}$	$9.960^{\pm 0.073}$
EnergyMoGen (2025)	$0.526^{\pm 0.003}$	$0.718^{\pm 0.003}$	$0.815^{\pm 0.002}$	$0.176^{\pm 0.006}$	$2.931^{\pm 0.007}$	$\mathbf{9.500}^{\pm 0.091}$
MLD (2023b)	$0.481^{\pm 0.003}$	$0.673^{\pm 0.003}$	$0.772^{\pm 0.002}$	$0.473^{\pm 0.013}$	$3.196^{\pm 0.010}$	$9.724^{\pm 0.082}$
w/ ReAlign (Ours)	$0.567^{\pm 0.003}$ **(+17.9%)**	$0.759^{\pm 0.003}$ **(+12.8%)**	$0.848^{\pm 0.003}$ **(+9.8%)**	$0.195^{\pm 0.005}$ **(+58.8%)**	$2.704^{\pm 0.007}$ **(+15.4%)**	$9.474^{\pm 0.068}$ **(+86.9%)**
MLD++(2025)	$0.548^{\pm 0.003}$	$0.738^{\pm 0.003}$	$0.829^{\pm 0.002}$	$0.073^{\pm 0.003}$	$2.810^{\pm 0.008}$	$9.658^{\pm 0.089}$
w/ ReAlign (Ours)	$\mathbf{0.572}^{\pm 0.002}$ (+4.4%)	$\mathbf{0.764}^{\pm 0.002}$ (+3.5%)	$\mathbf{0.852}^{\pm 0.001}$ (+2.8%)	$\mathbf{0.055}^{\pm 0.003}$ (+24.7%)	$\mathbf{2.648}^{\pm 0.008}$ (+5.8%)	$9.478^{\pm 0.055}$ (+83.9%)

	Methods	Noise	Text-Motion Retrieval↑ R@1	R@2	R@3	R@5	R@10	Motion-Text Retrieval↑ R@1	R@2	R@3	R@5	R@10
HumanML3D	TEMOS (2022)	✗	40.49	53.52	61.14	70.96	84.15	39.96	53.49	61.79	72.40	85.89
	T2M (2022b)	✗	52.48	71.05	80.65	89.66	**96.58**	52.00	71.21	81.11	89.87	<u>96.78</u>
	TMR (2023)	✗	67.16	81.32	86.81	91.43	95.36	67.97	81.20	86.35	91.70	95.27
	LaMP (2025)	✗	67.18	81.90	87.04	**92.00**	95.73	68.02	82.10	87.50	92.20	**96.90**
	ReAlign (ours)	✓	**67.59**	**82.24**	**87.44**	<u>91.97</u>	<u>96.28</u>	**68.94**	**82.86**	**87.95**	**92.44**	<u>96.28</u>
KIT-ML	TEMOS (2022)	✗	43.88	58.25	67.00	74.00	84.75	41.88	55.88	65.62	75.25	85.75
	T2M (2022b)	✗	42.25	62.62	75.12	87.50	96.12	39.75	62.75	73.62	86.88	95.88
	TMR (2023)	✗	49.25	69.75	78.25	87.88	95.00	50.12	67.12	76.88	88.88	94.75
	ReAlign (ours)	✓	**52.84**	**71.66**	**82.96**	**91.19**	**97.59**	**52.98**	**72.87**	**84.38**	**92.61**	**96.87**