MotorNerve

A Character Animation System using Machine Learning

Yuchen Liao

Senior Engine Programmer, Tencent Games

Songnan Li

Deputy Director, Tencent Games

Tencent Games

Tencent Games, founded in 2003, is the world's leading game developer and operator. As an advocate and practitioner of the concept of "super digital scene", Tencent Games pays high attention to and attaches great importance to the healthy development of minors, and is committed to promoting the game as an important driving force to promote the development of cutting-edge technology, promote the promotion of excellent culture, incubate innovative talents, and increase the efficiency of social public welfare through technological innovation, creative stimulation, combination of production, education and research, global layout, and public welfare practice, Create more breakthrough and constructive value for the development of industry and society.

At the same time, Tencent games also actively promote the development of the e-sports industry, and work with global partners to build an open, collaborative, co-prosperity and symbiosis industrial ecosystem, creating high-quality digital life experience for users.

A Character Animation System using Machine Learning

Product	**MotorNerve**	
Scenario	Locomotion Animation	Interactive Animation
Technology	Learned Motion Matching	Motion In-Betweening

MotorNerve team members

About TiMi Studio Group

TiMi Studio Group is a subsidiary of Tencent Games and a leading global game development, operations and publishing team that strives to improve global players' entertainment quality.

Our studio team within TiMi focuses on racing and anime-style RPG games for PC and mobile. Our main products include ***QQ Speed*** (PC/mobile), the upcoming ***Need for Speed Mobile*** and an unannounced, in-development Anime-style RPG title.

Tencent Games GDC TiMi STUDIO GROUP

01 Motion Matching

For Locomotion

02 Motion In-Betweening

For Motion Transitions

01 Motion Matching

For Locomotion

Outline of Part 1

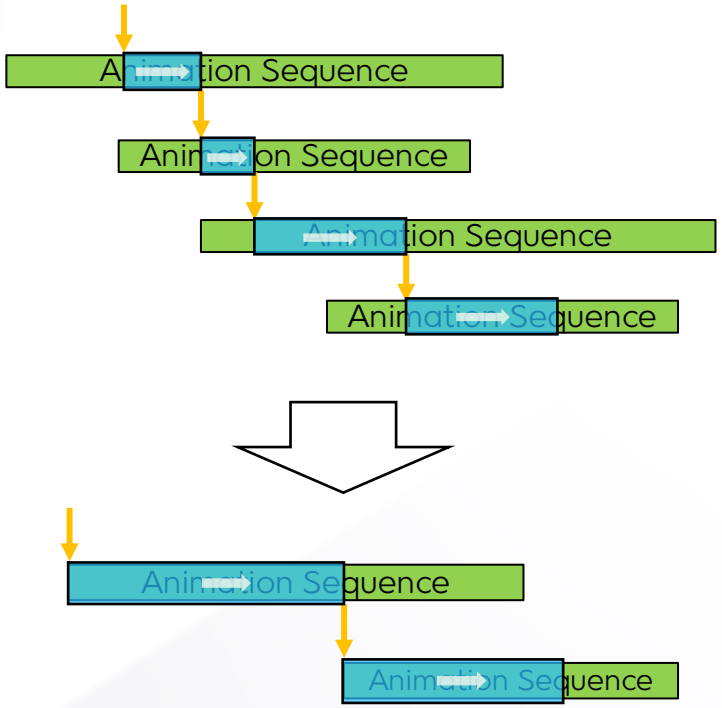
- Animation Polish
- BMM vs LMM (Why we use machine learning)
- Performance

Locomotion Demo with Motion Matching

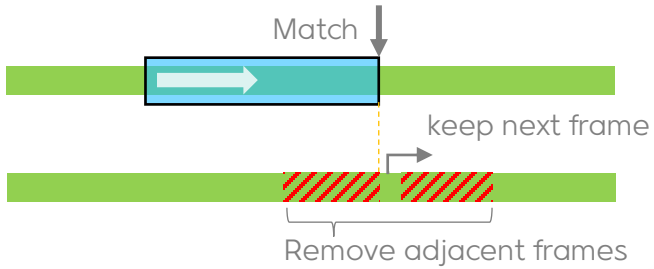
Unarmed Locomotion (Veer)

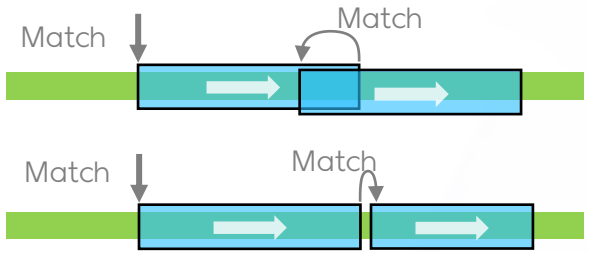
Aiming Locomotion (Strafe)

Some of our methods for animation polish



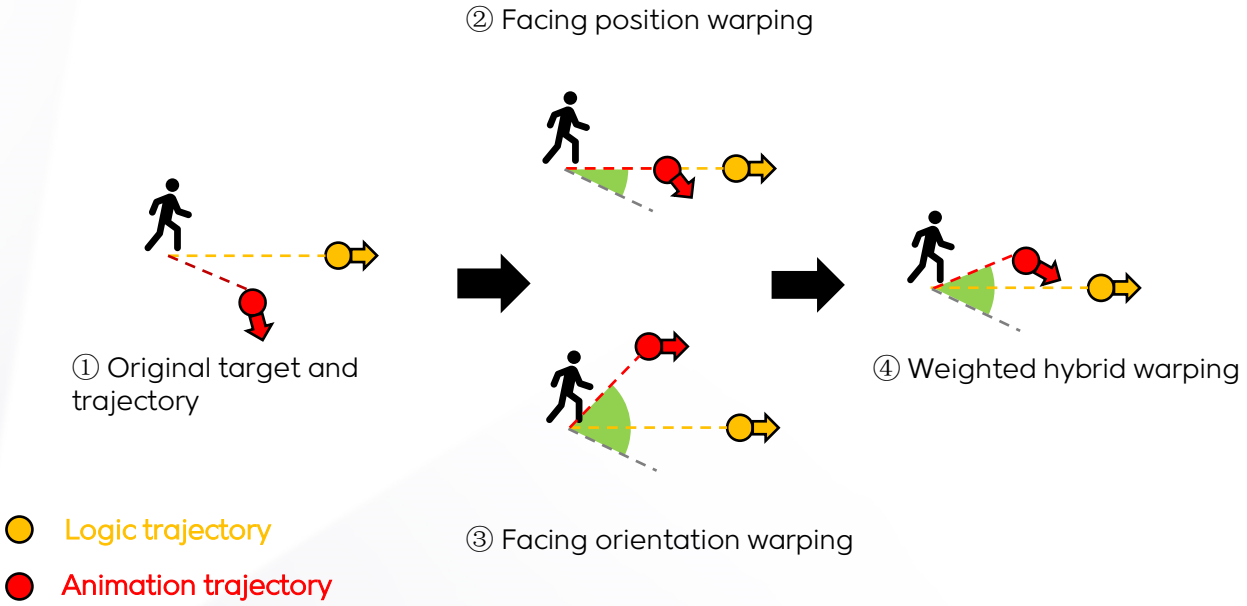
Completeness optimization

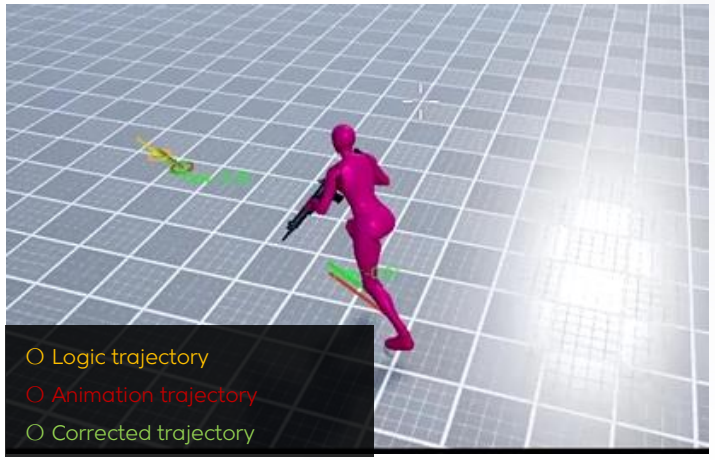




Continuity optimization

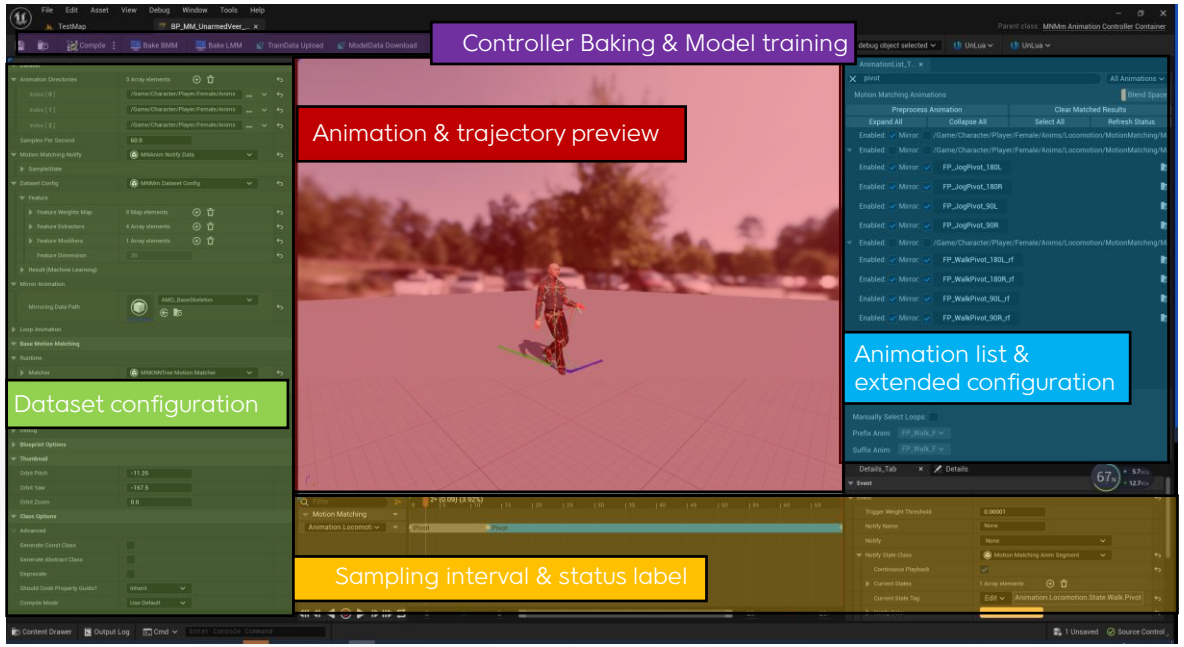
Some of our methods for animation polish



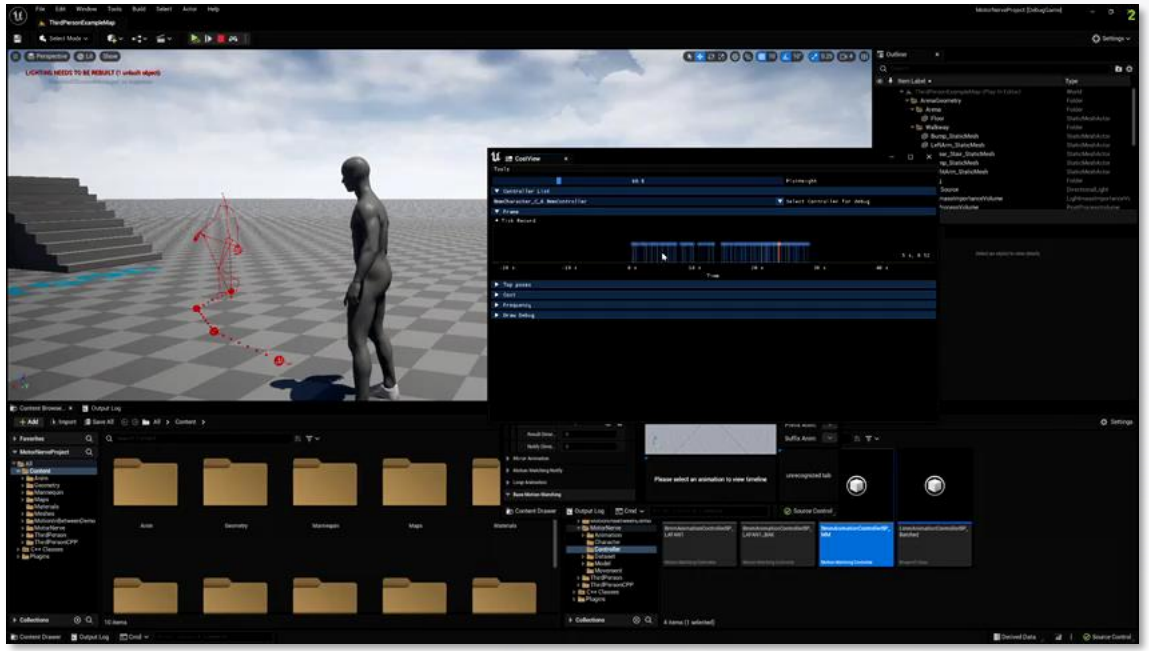


Runtime trajectory warping

Toolchains to help animation polish



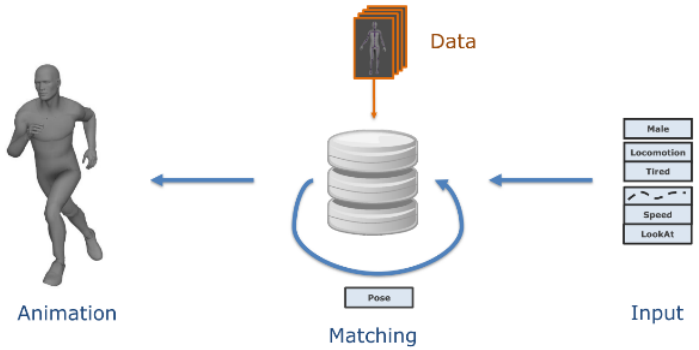
Motion Matching Editor



Debug Tools

Why we use machine learning in Motion Matching?

For Development

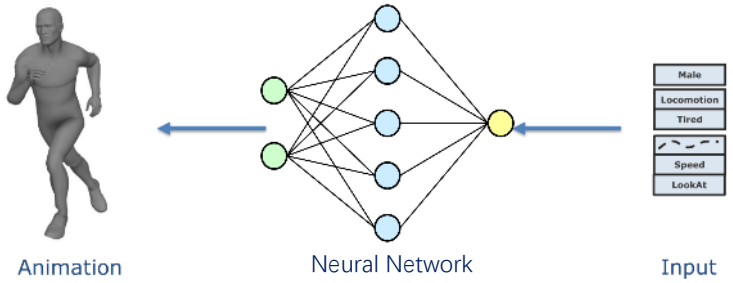


Base Motion Matching (BMM)

✓ Rapid iteration

× High memory usage

For Runtime

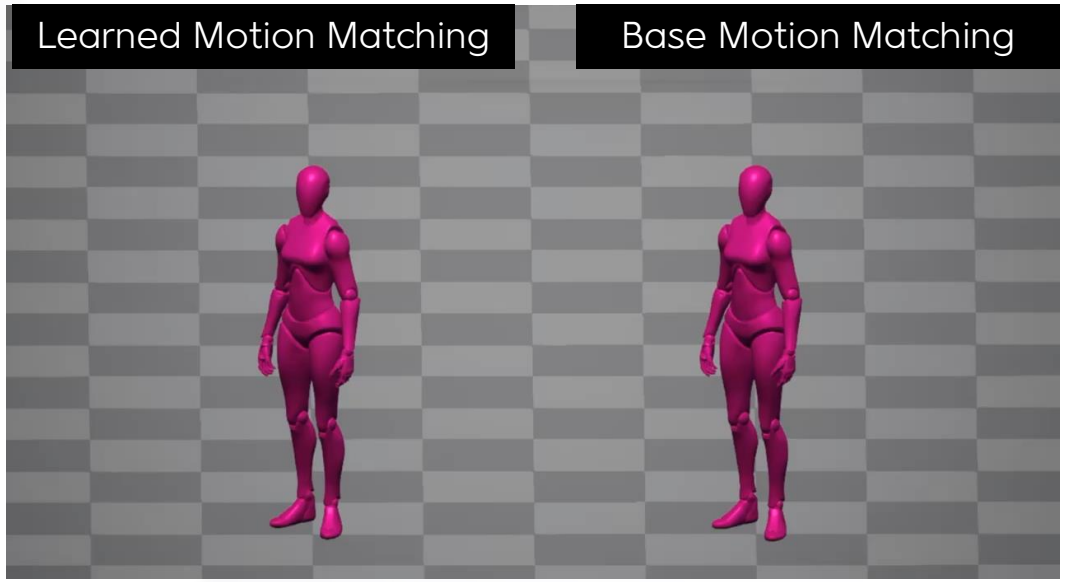


Learned Motion Matching (LMM)

✓ High performance

× Hard to iteration

Motion Quality of LMM and BMM



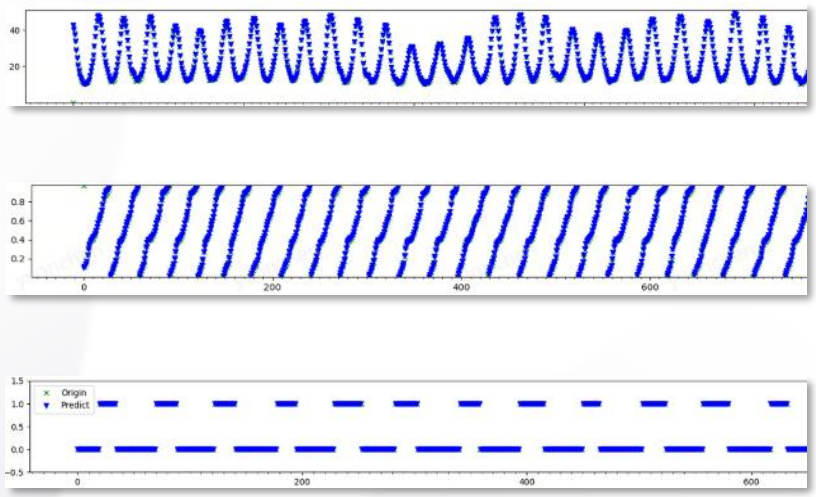
motion of LMM is identical with BMM

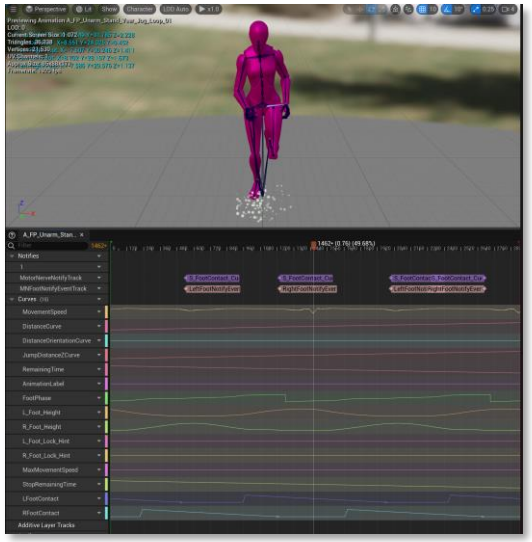
Subjective experimental results

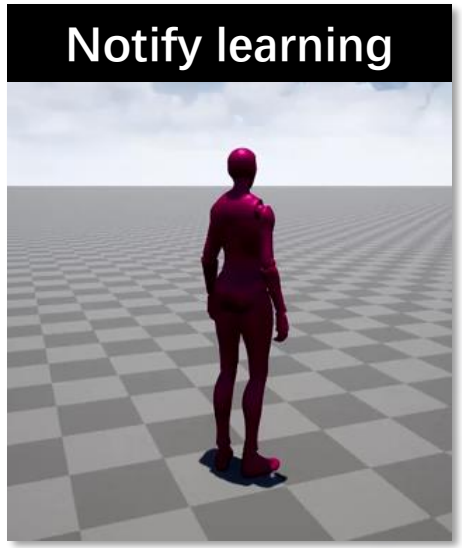
Clip type	Clip name	L_{rc}	L_f(LMM)	L_f(BMM)
Training	RandRec_0	-4.588	17.102	**15.687**
Training	RandRec_1	-4.654	**30.280**	30.980
Training	RandRec_2	-5.404	34.510	**30.268**
Training	RandRec_3	-5.038	**31.816**	33.237
Test	TestRec_1	-0.042	17.210	**14.706**
Test	TestRec_2	-3.723	**26.446**	27.758
Test	TestRec_3	-3.552	**23.072**	26.276

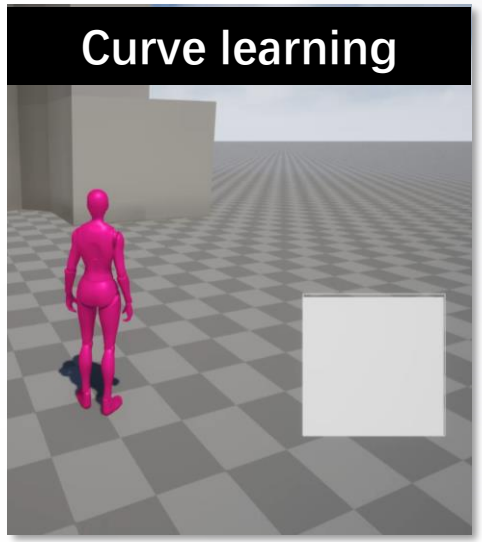
Objective quality data

Animation curve and notify in LMM





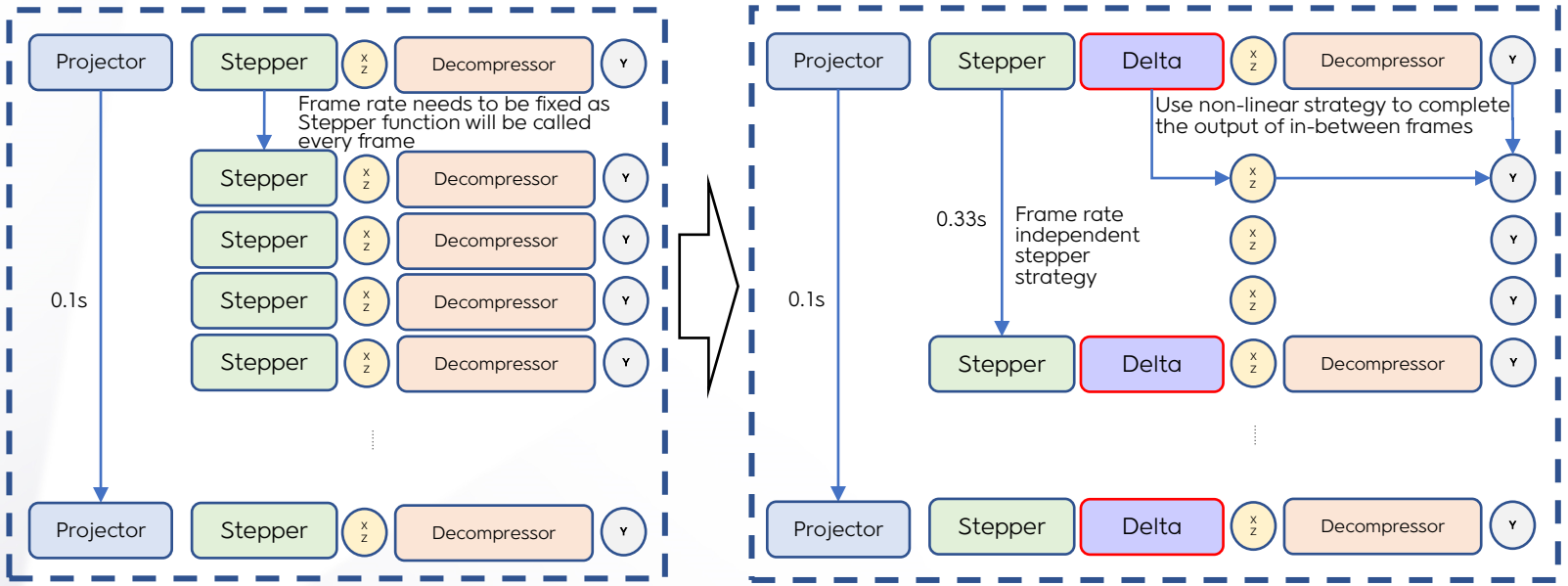


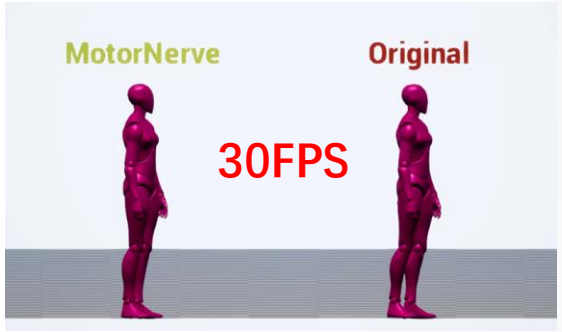


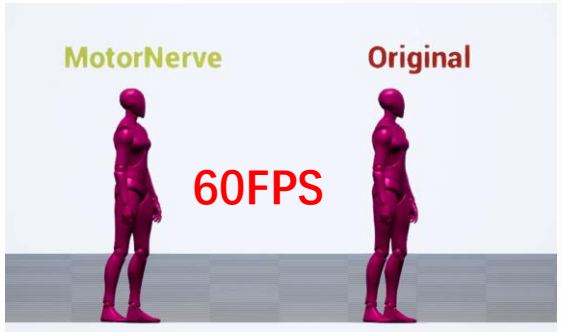
Curves generated through curve fitting learning

Automatically process animation curves and Notify information

Inference architecture optimization

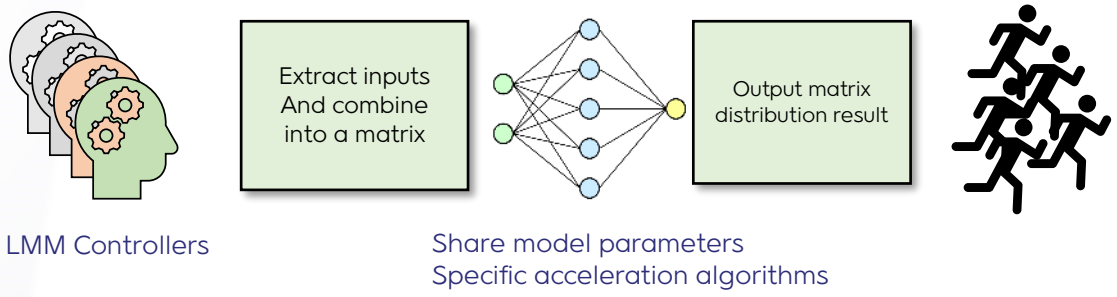






- ✓ Adapt to different frame rates
- ✓ Computationally efficient

Batch inference optimization



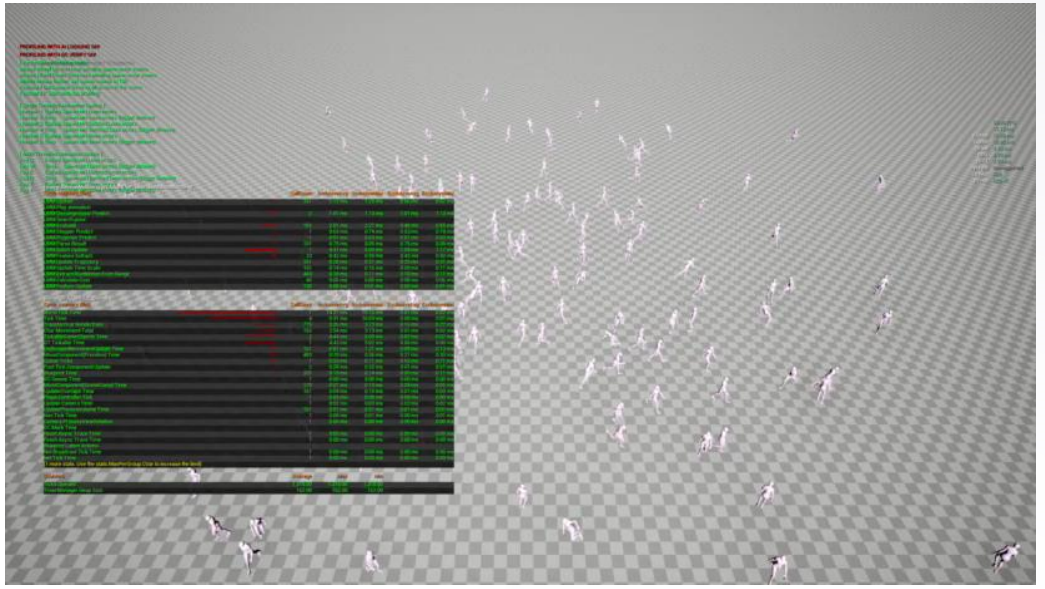
LMM Controllers

Share model parameters
Specific acceleration algorithms

50%

When more than 30 characters

Saved CPU Cost



160 characters; 60 FPS
(Single-threaded)

Part1 Take Aways

- We provide some methods and toolchains to help polish animation
- We ensure LMM is Identical with BMM, including motion ,curves and notifications
- We optimized the performance of LMM, especially when multiple characters

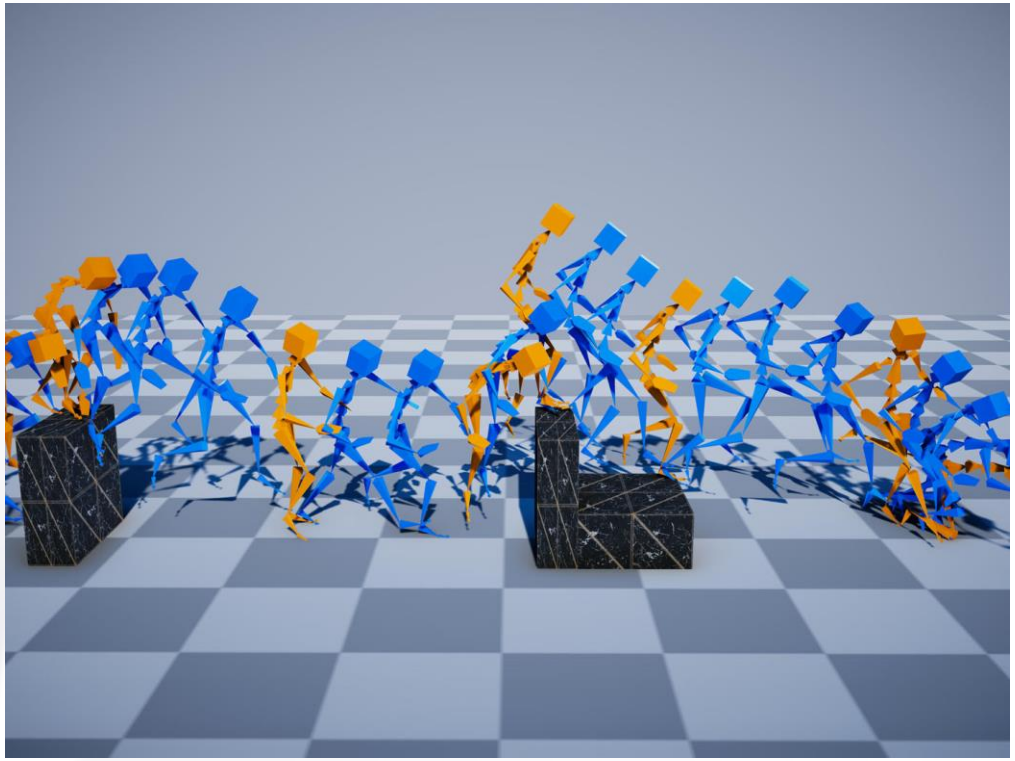
02 Motion In-Betweening

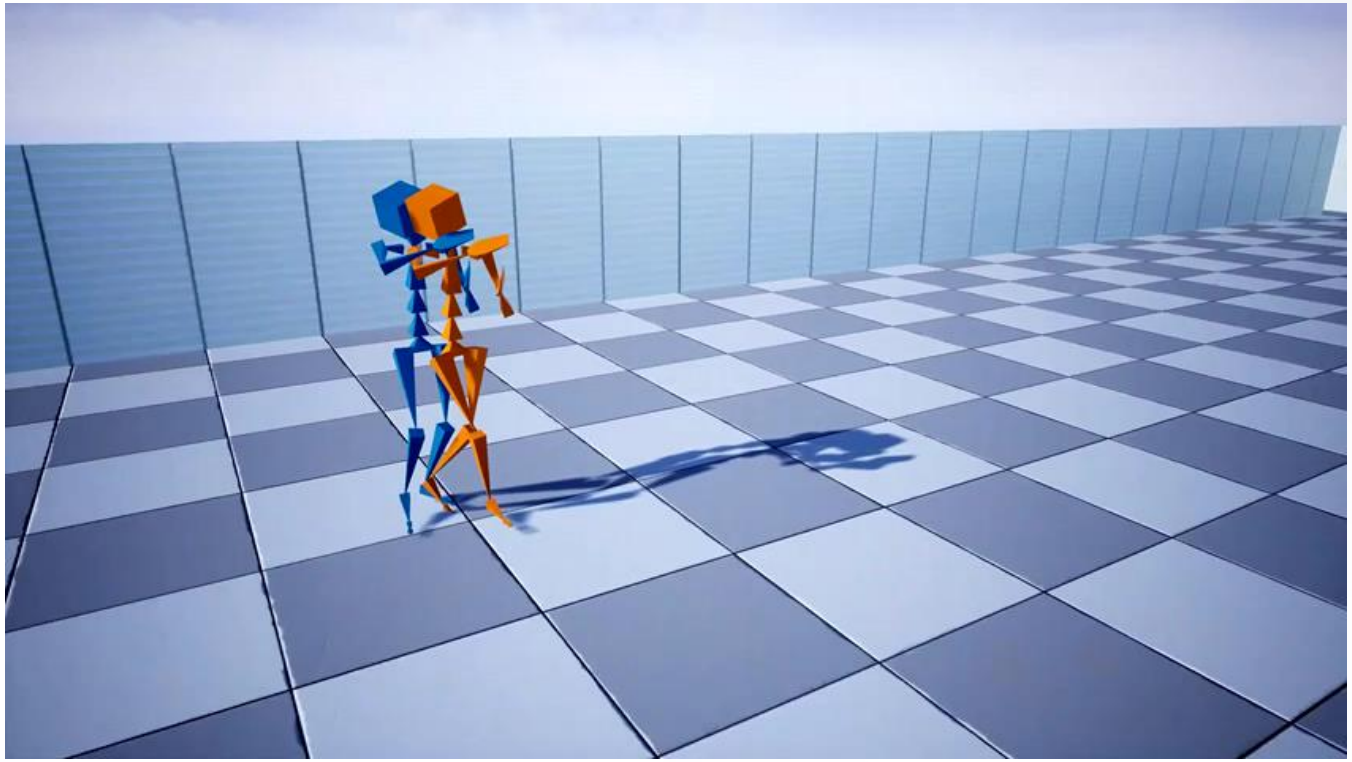
For Motion Transitions

Outline of Part 2

- What is Motion In-Betweening (MIB) and Why We Need It
- Classification of MIB Methods
- Our Method
- Applications: Interaction and locomotion

What is Motion In-Betweening (MIB)?

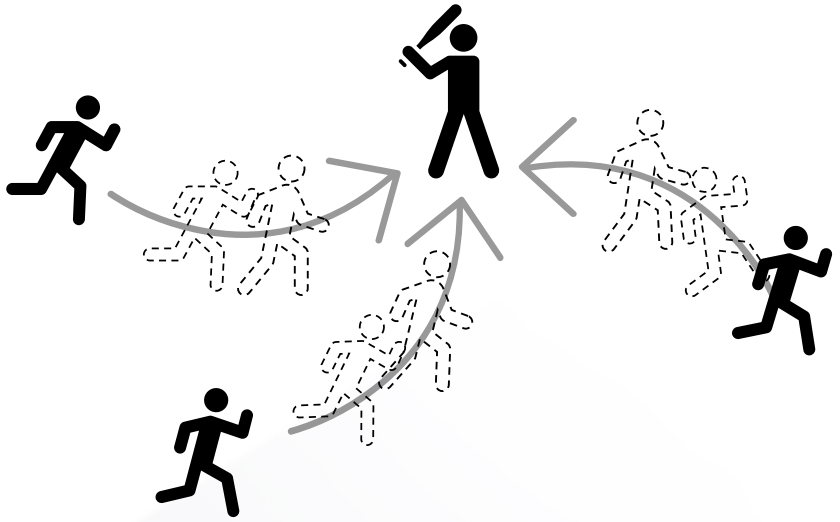




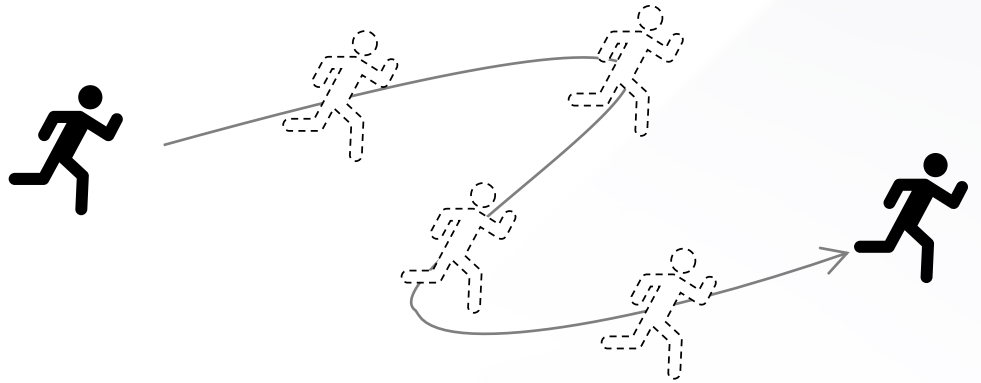
Zhejiang University & Tencent Games

Real-time Controllable Motion Transition for Chataracters,
SIGGRAPH 2022

Why we use MIB in MotorNerve?

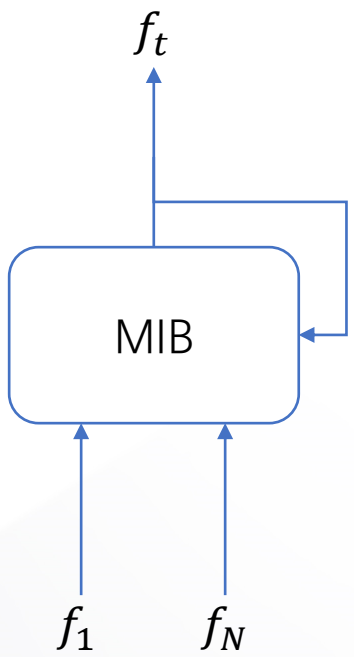


Transition for interactive animation

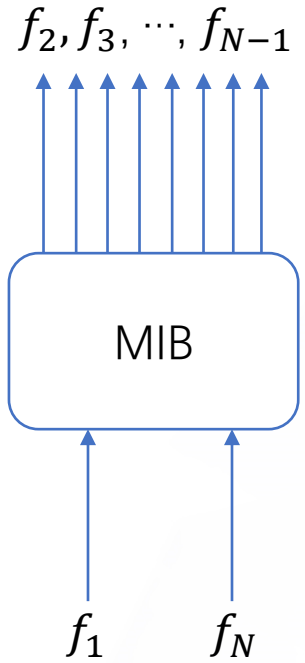


Transition for locomotion

Classification of MIB Methods

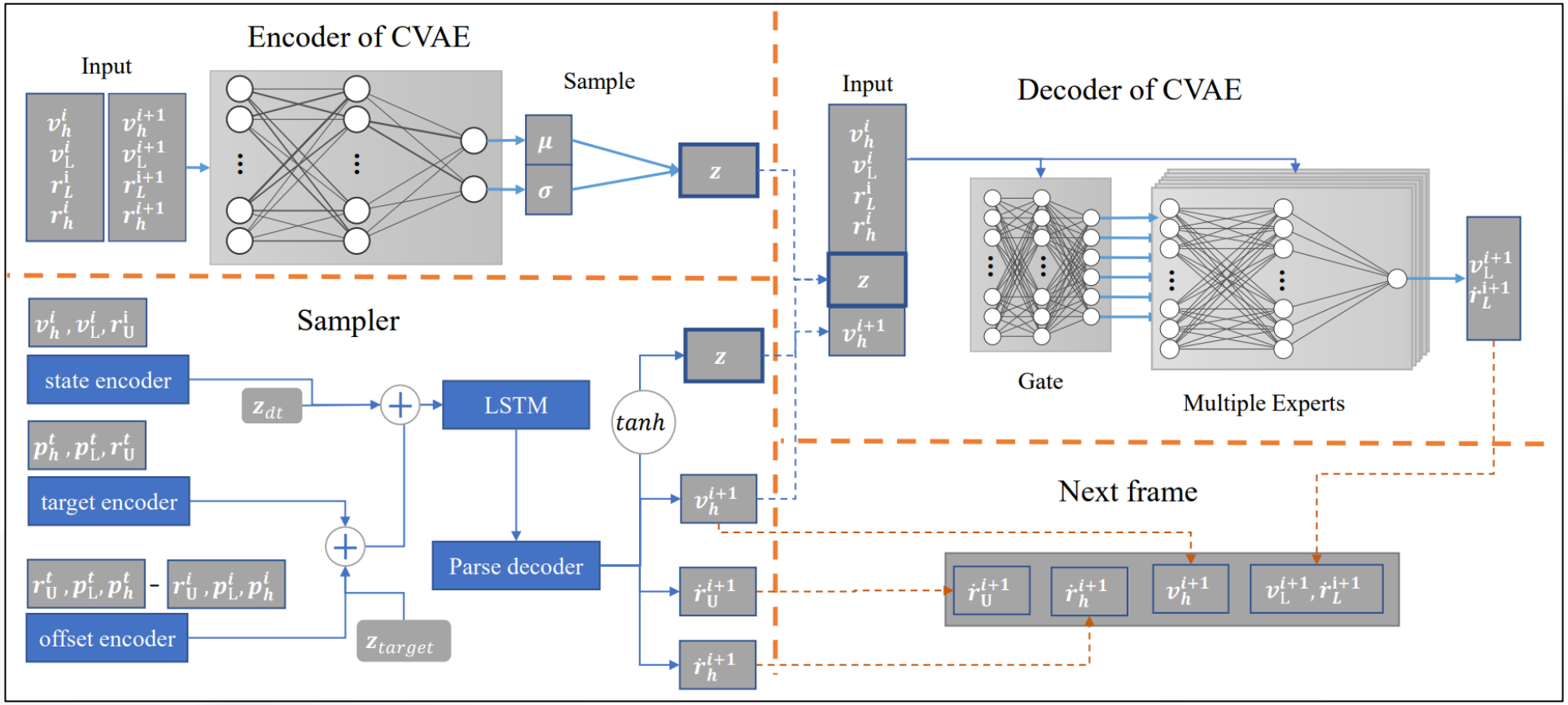


Autoregressive
(Online)



Transformer Encoder/Diffusion Model
(Offline)

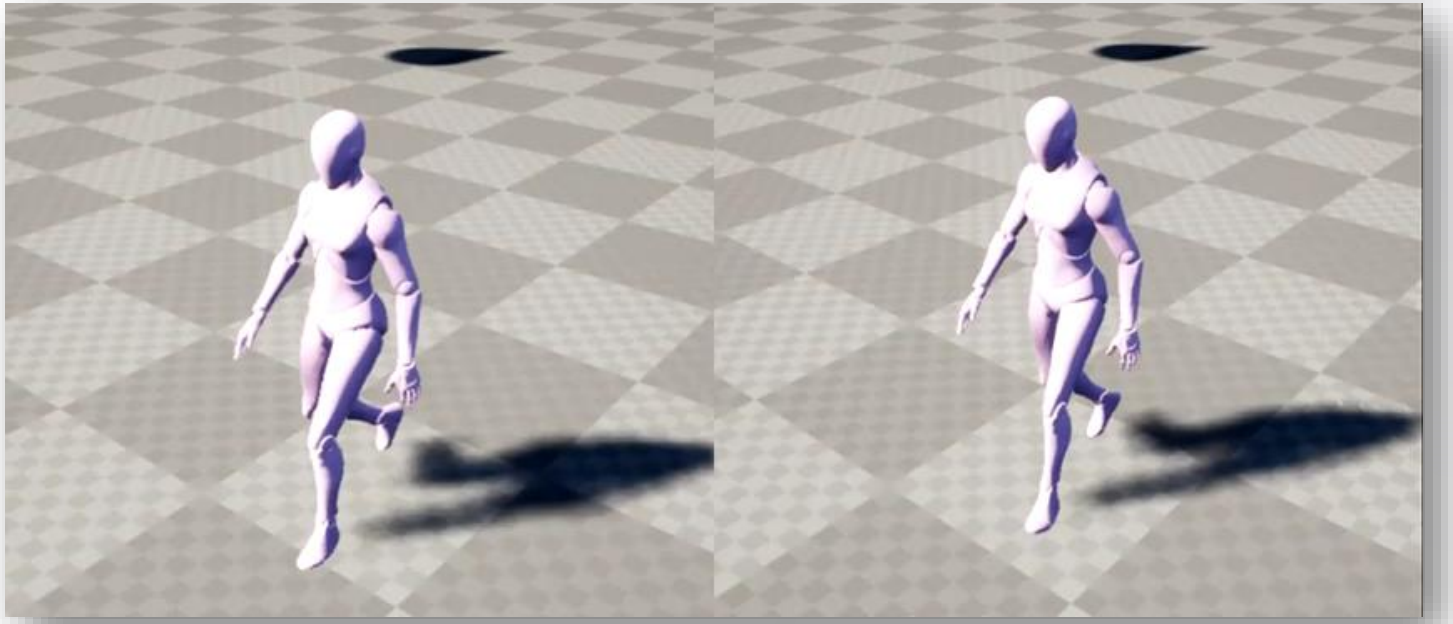
Framework of Our Method



Real-time Controllable Motion Transition for Chataracters, SIGGRAPH 2022

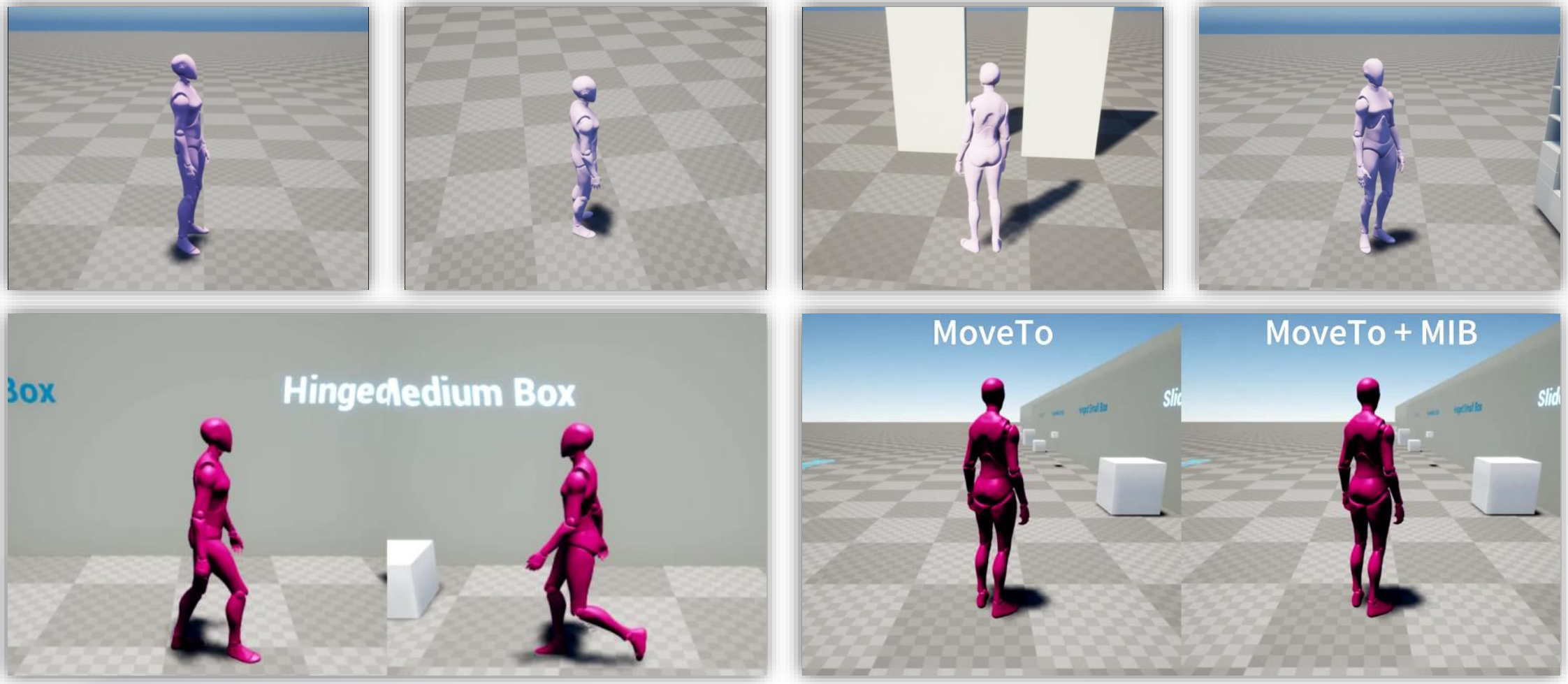
Further Extension on the paper's work in MotorNerve

Trajectory Control



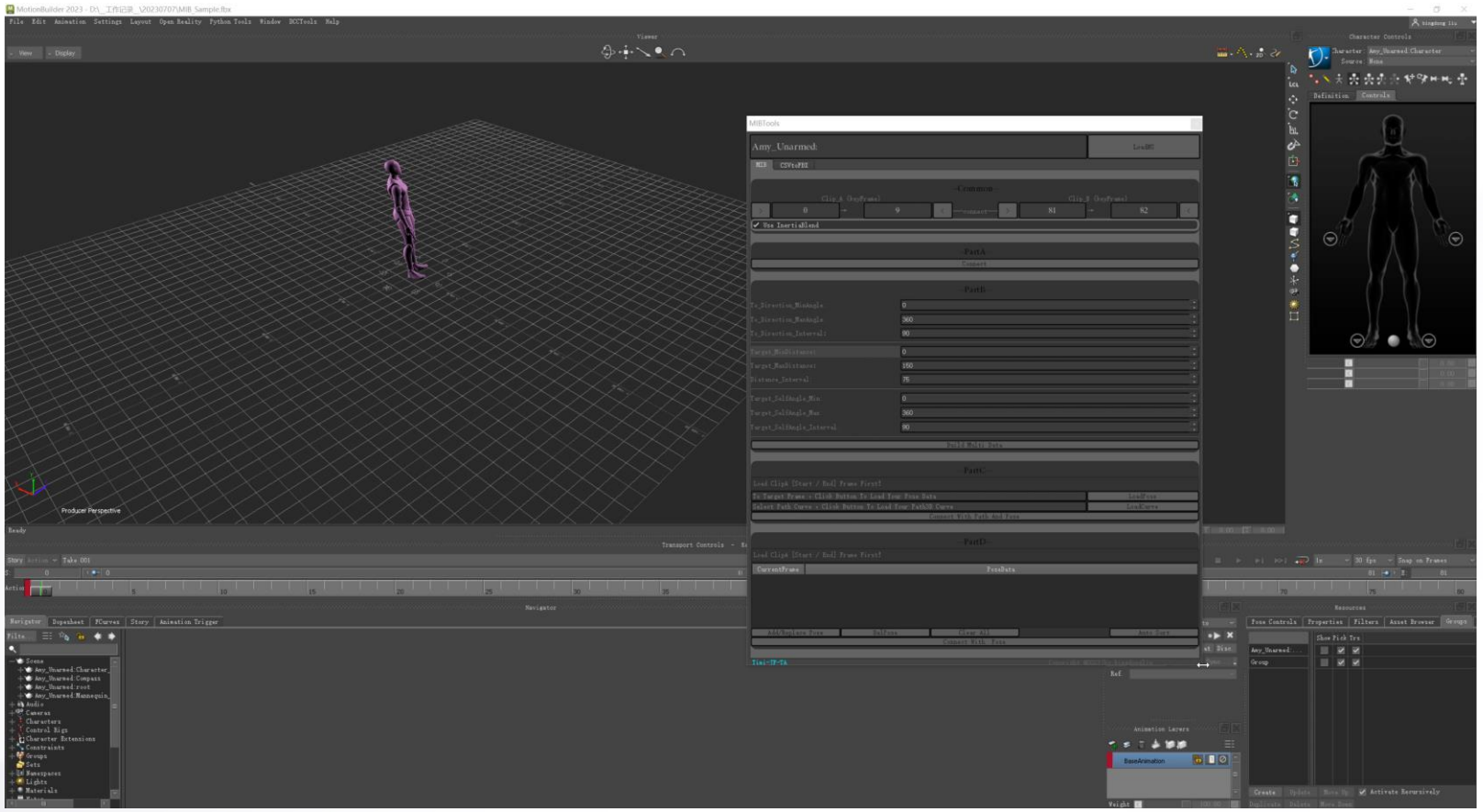
Frame Number Prediction
[Left: fixed frame number, Right: adaptive frame number]

Transition for Interactive Animation



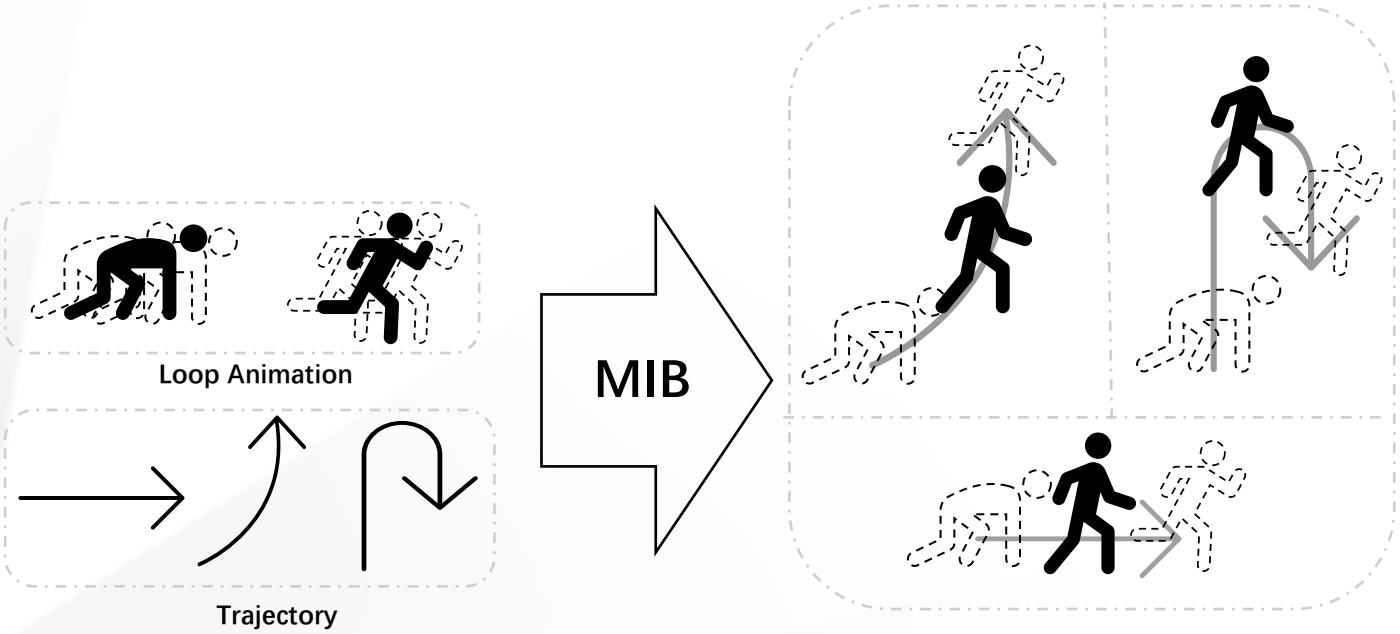
Transition for Interactive Animation

Integration into the DCC tools

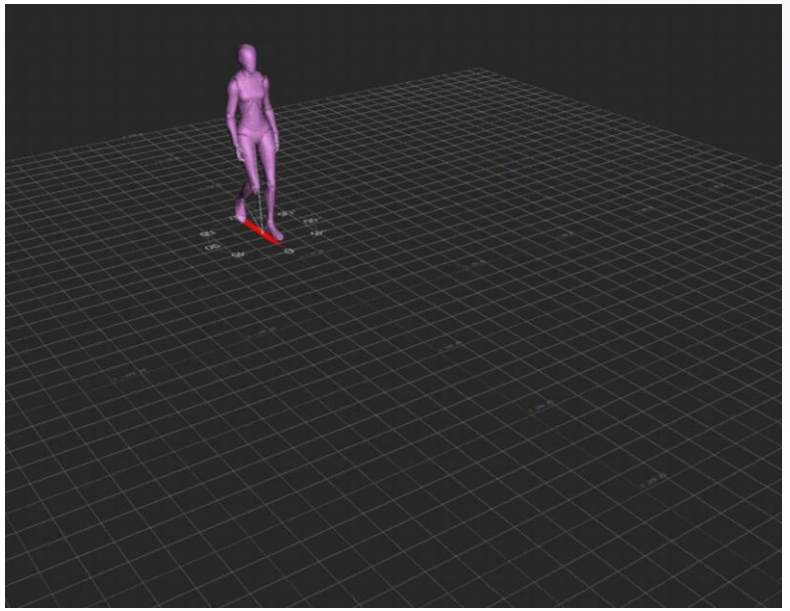


Transition for Locomotion

To generate locomotion that matches trajectory

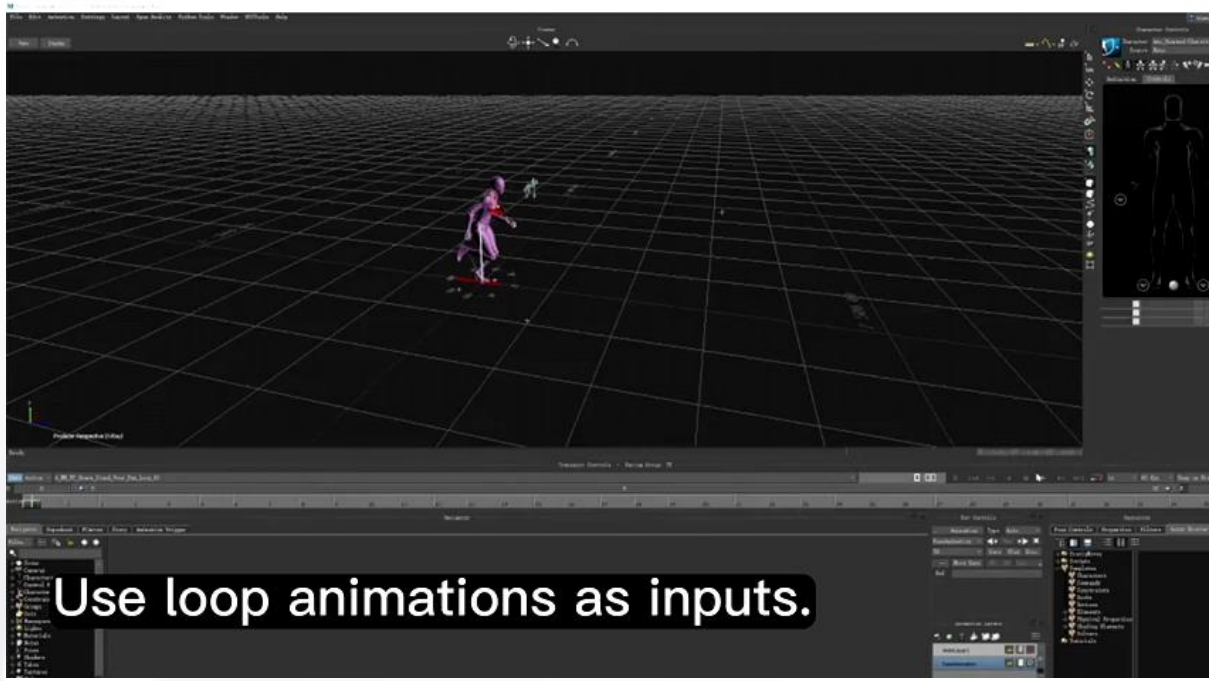


Locomotion that matches given trajectory

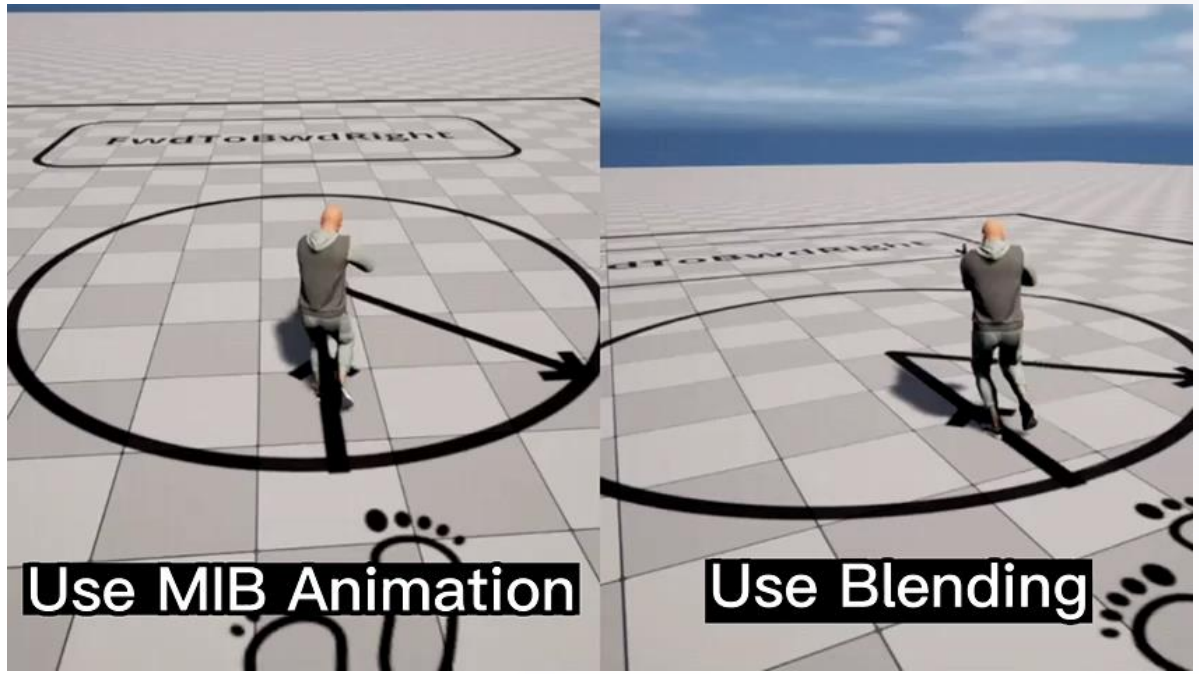


Walking to walking transition (changing direction)

Transition for Locomotion



Batched MIB generation in MotionBuilder



Comparision with blending animations

Part2 Take Aways

- MIB is a deep-learning-based motion generation method
- We use it to generate motion transition for interaction and locomotion
- We can use it online (in game) or offline (to generate animation in a DCC tool)

Thank You

Tencent Games

GDC

TiMi STUDIO GROUP