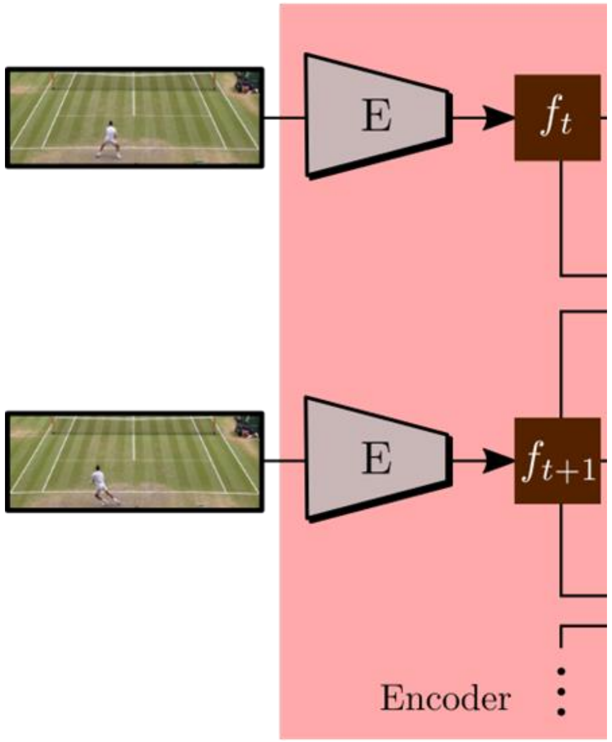
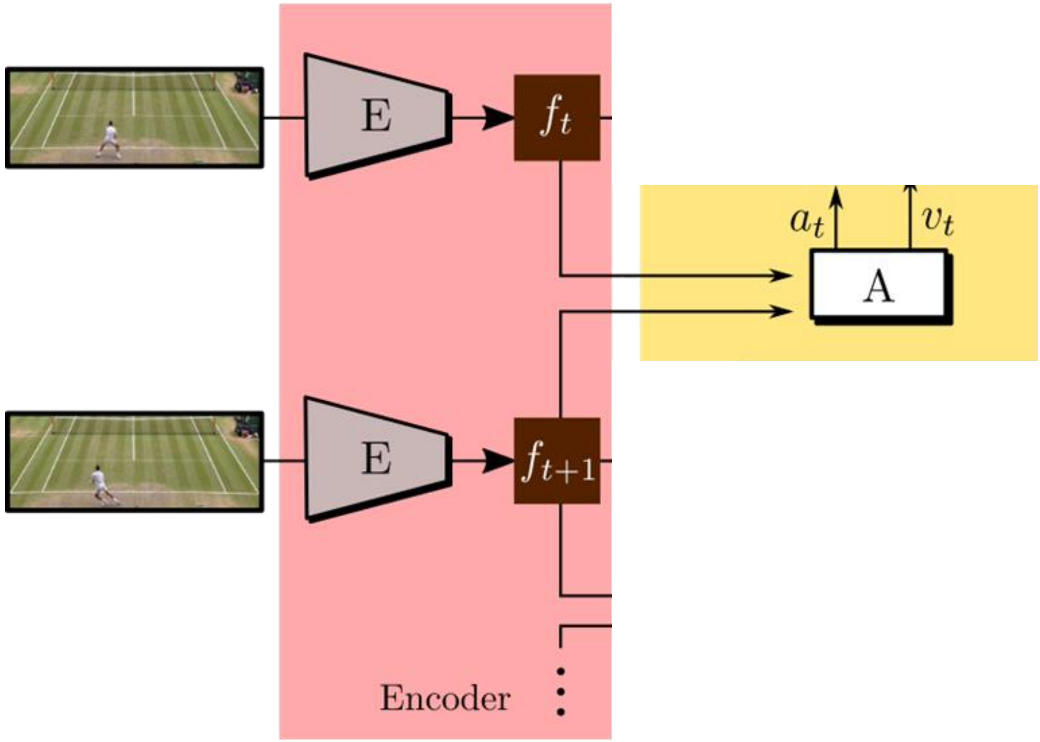
Architecture



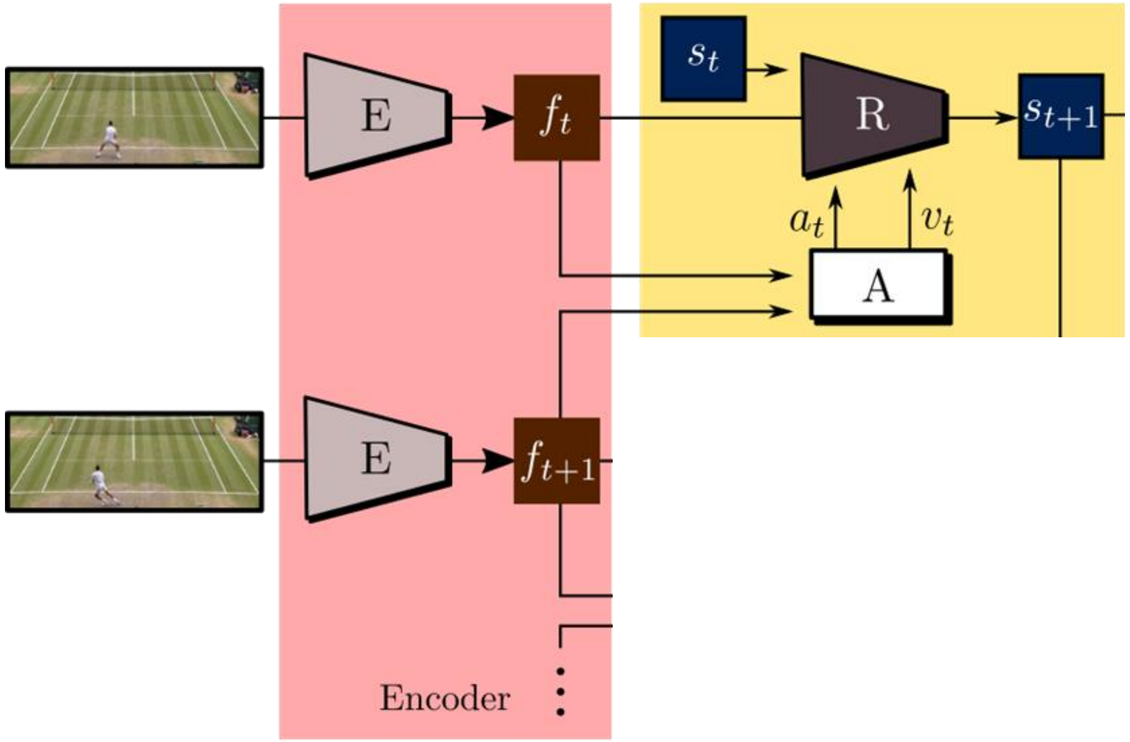
- First we sample an input sequence and use an encoder network to extract frame features

Architecture



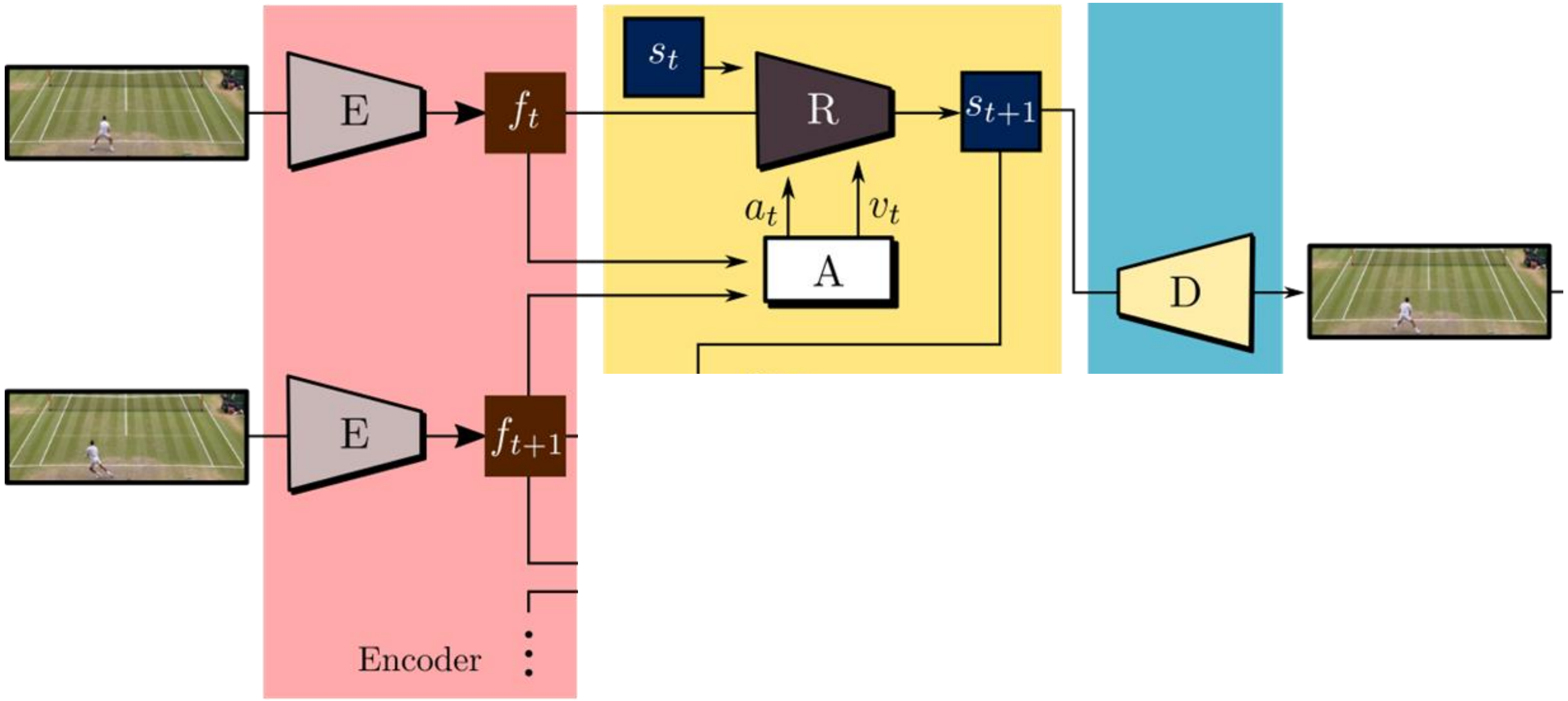
- Use then pairs of successive features to infer the action that was performed by the agent in the corresponding transition using an action network

Architecture



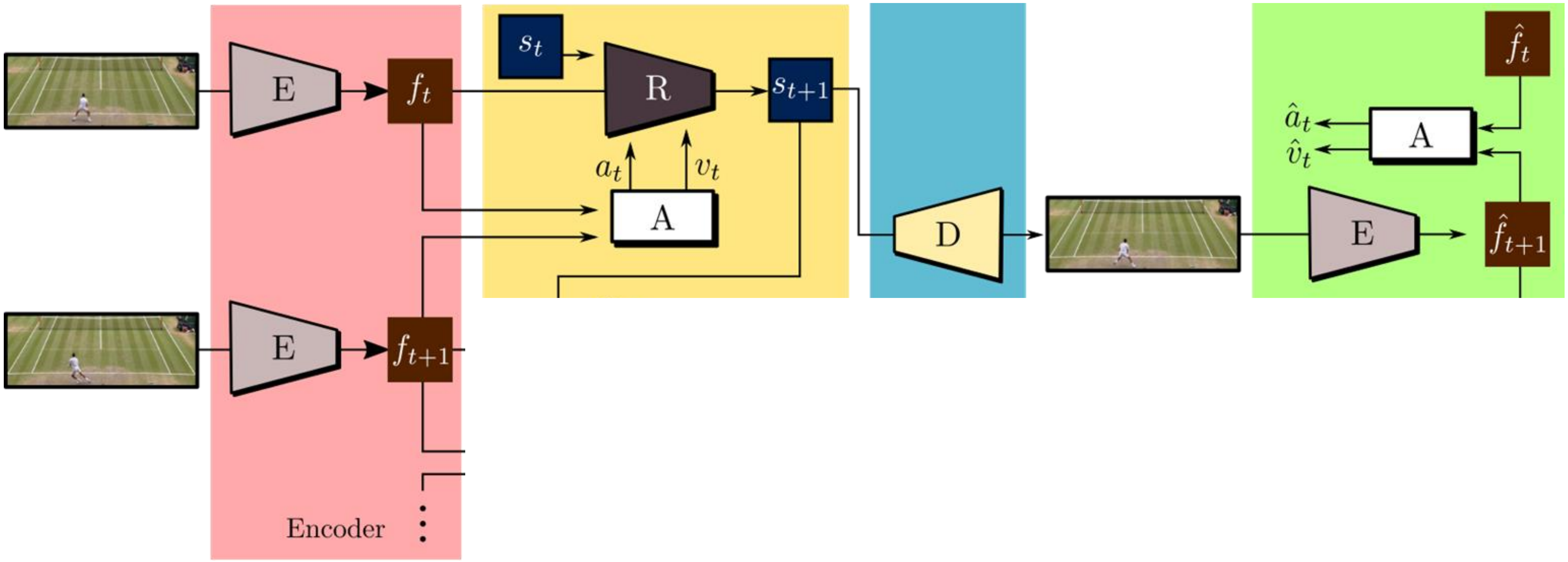
- Given the frame features and the action, a recurrent model is used to produce features representing the successive state

Architecture



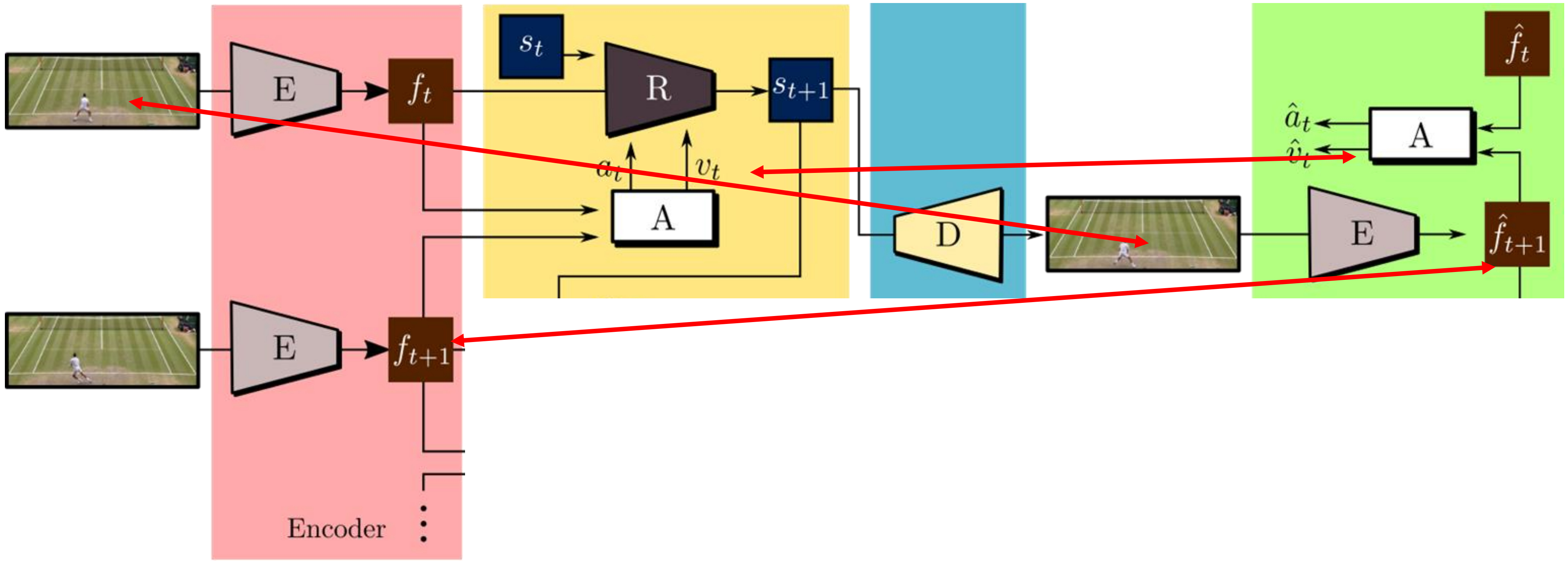
- The successive state is translated back to an image using a decoder network

Architecture



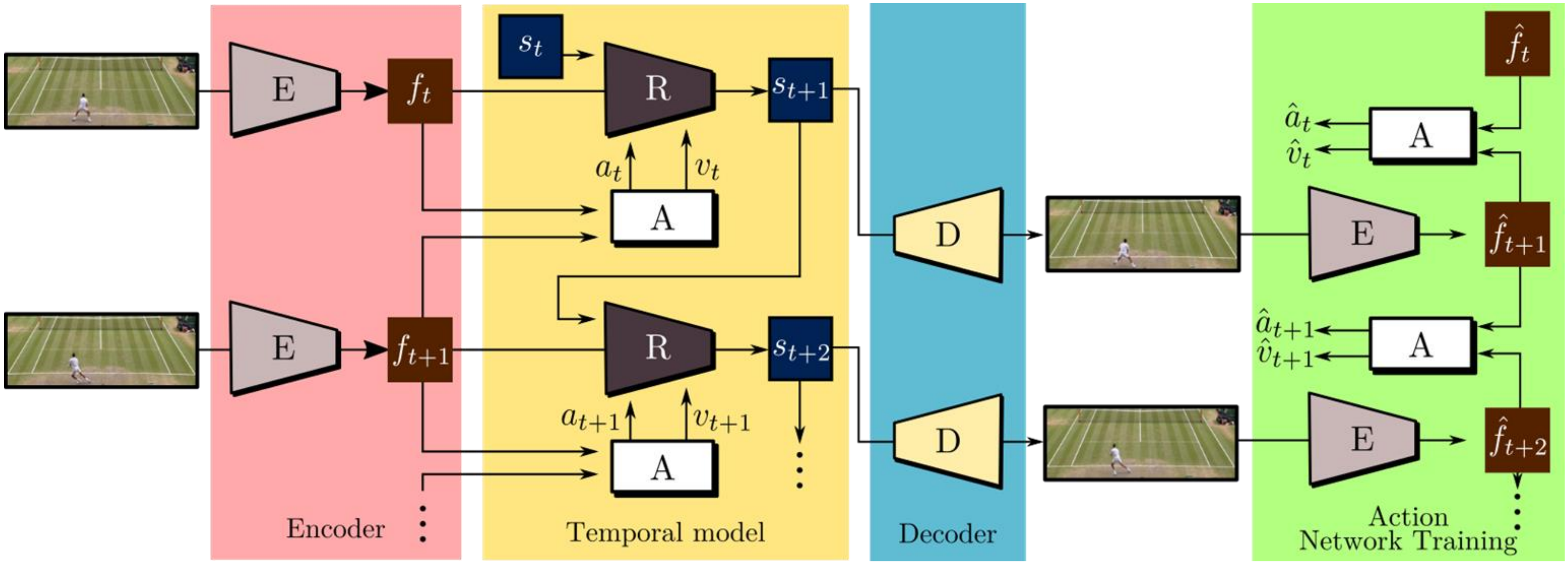
- For extra supervision, we encode back the produced frame using the encoder and the action network

Architecture



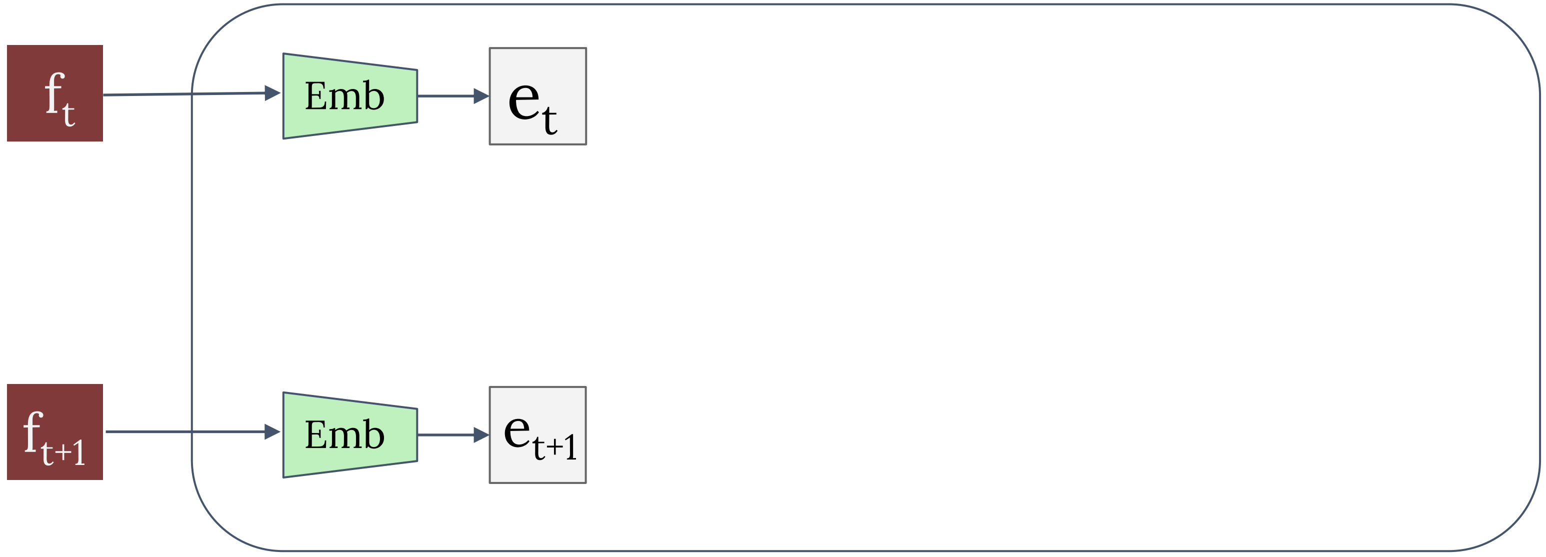
- Impose different self supervision losses on the frames, the frame features and the produced actions: use a mutual information maximization loss between actions and reconstructed actions as the main driving loss for action learning

Architecture



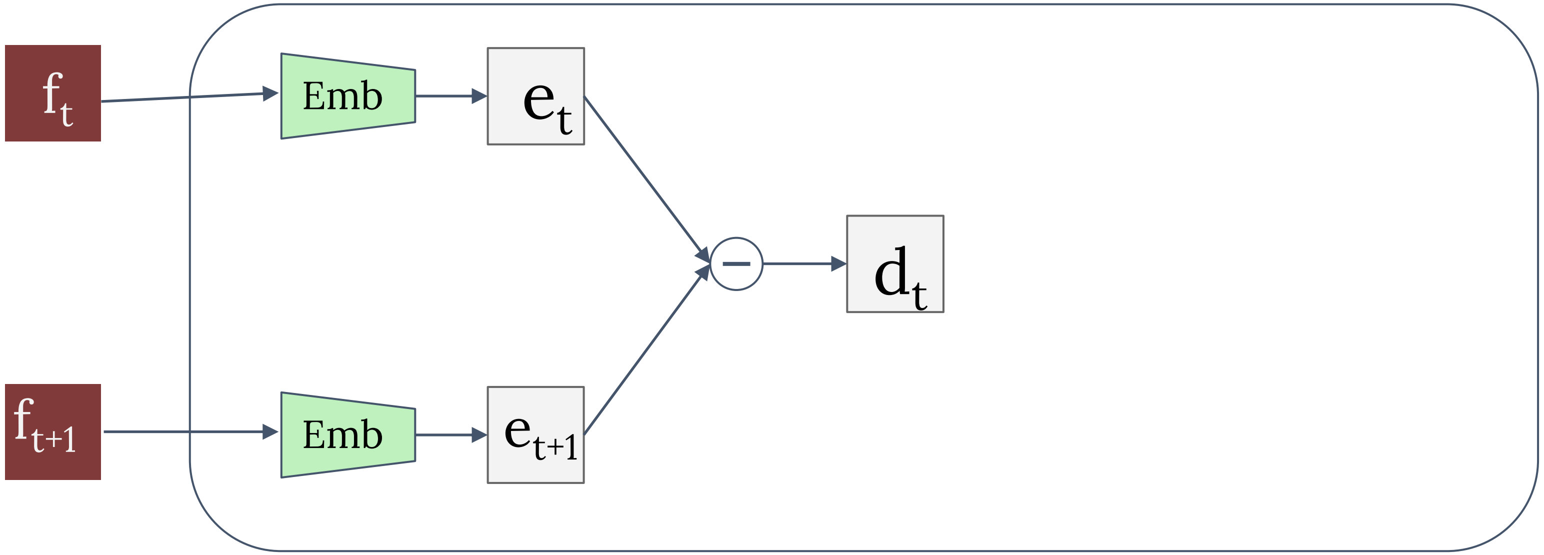
- The model is then unrolled over the whole sequence

Action Network



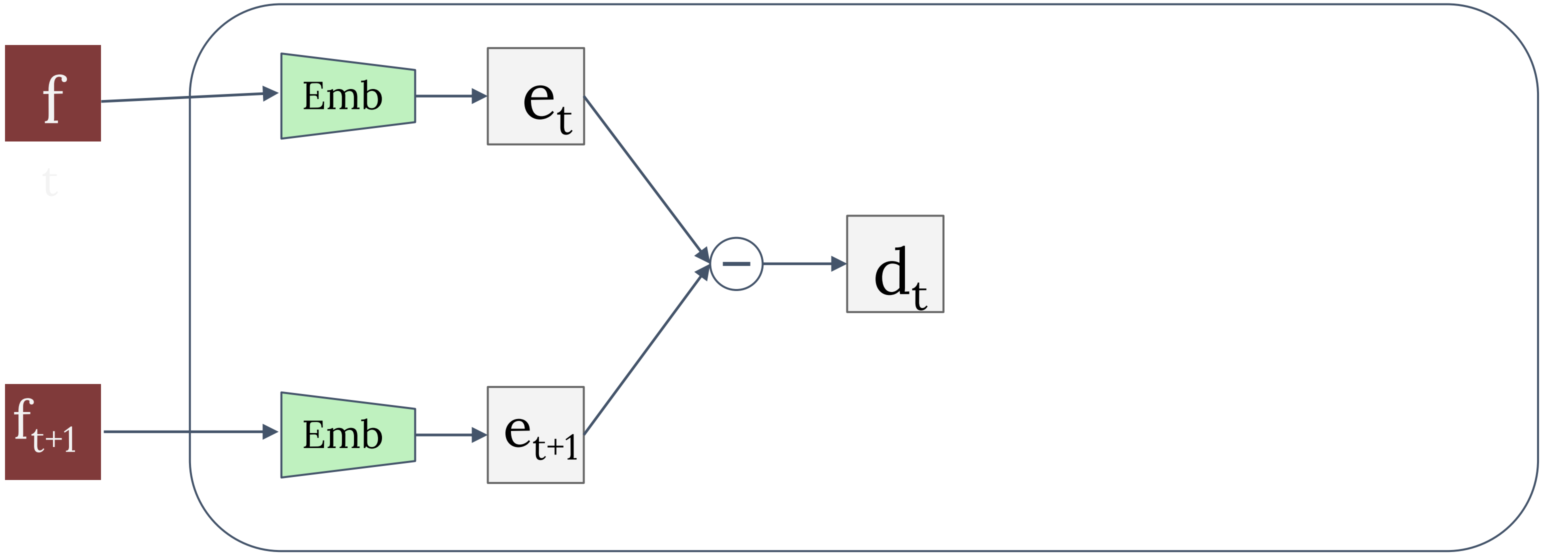
- The action network first encodes the frame features using a Multi Layer Perceptron to produce two embeddings

Action Network



- We take the difference between these embedding as the representation of the transition between two frames: action direction d_t

Action Network



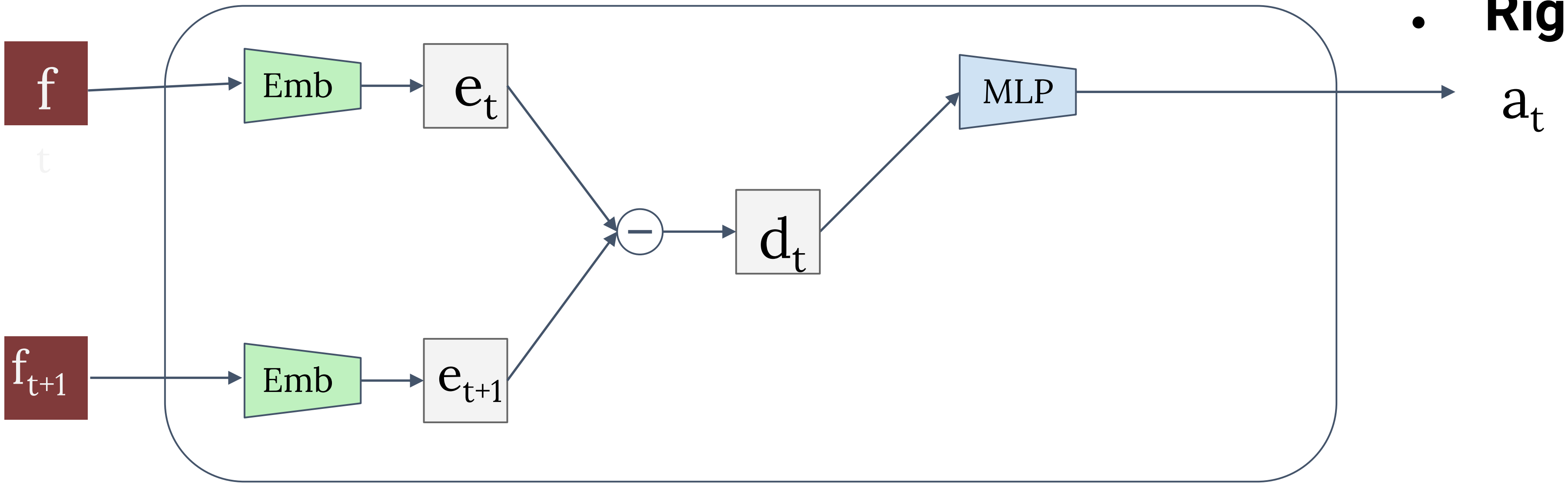
t-SNE plot of d_t

- When visualized, the learned space of action directions is a representation of the different types of transitions that are observed in the training videos

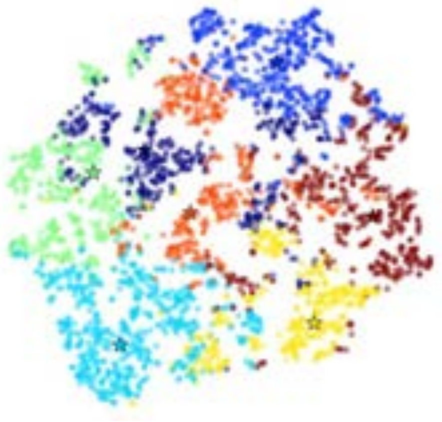
Action Network

Which action is done

- **Left**
- **Right**

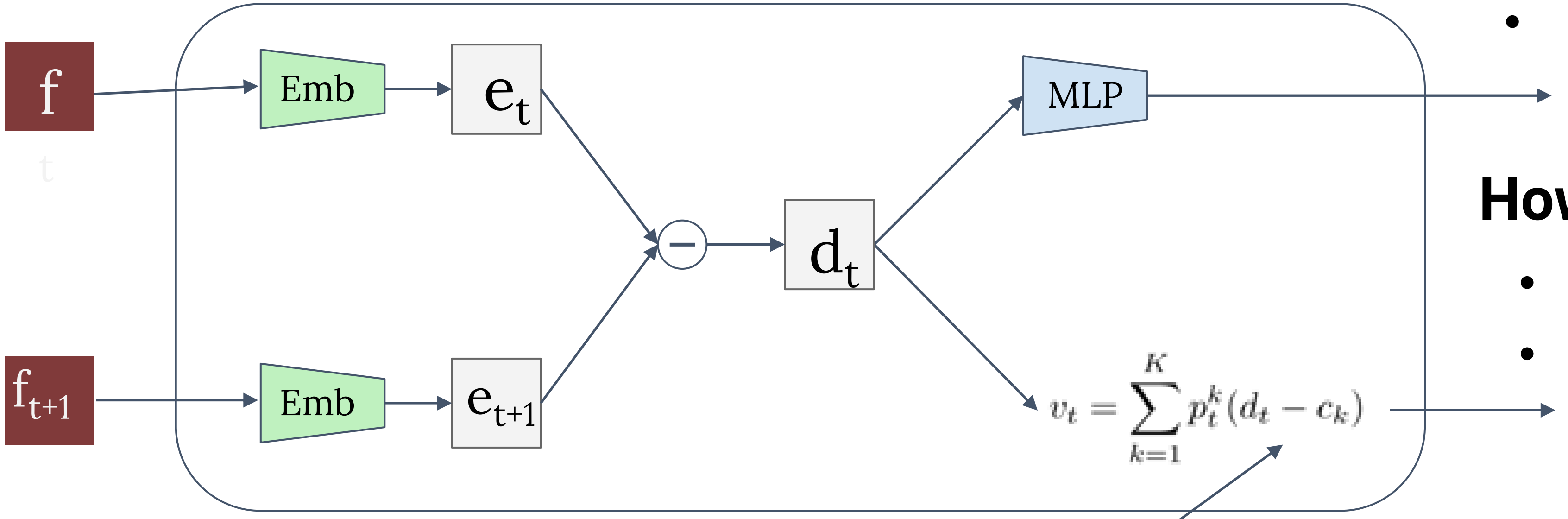


t-SNE plot of d_t



- Use an MLP to assign a label to each point d_t: the high-level action associated to the current frame
- Use of action variability embeddings to ensure a well-posed reconstruction loss on the frames

Action Network



Which action is done

- **Left**
- **Right**

How the action is done

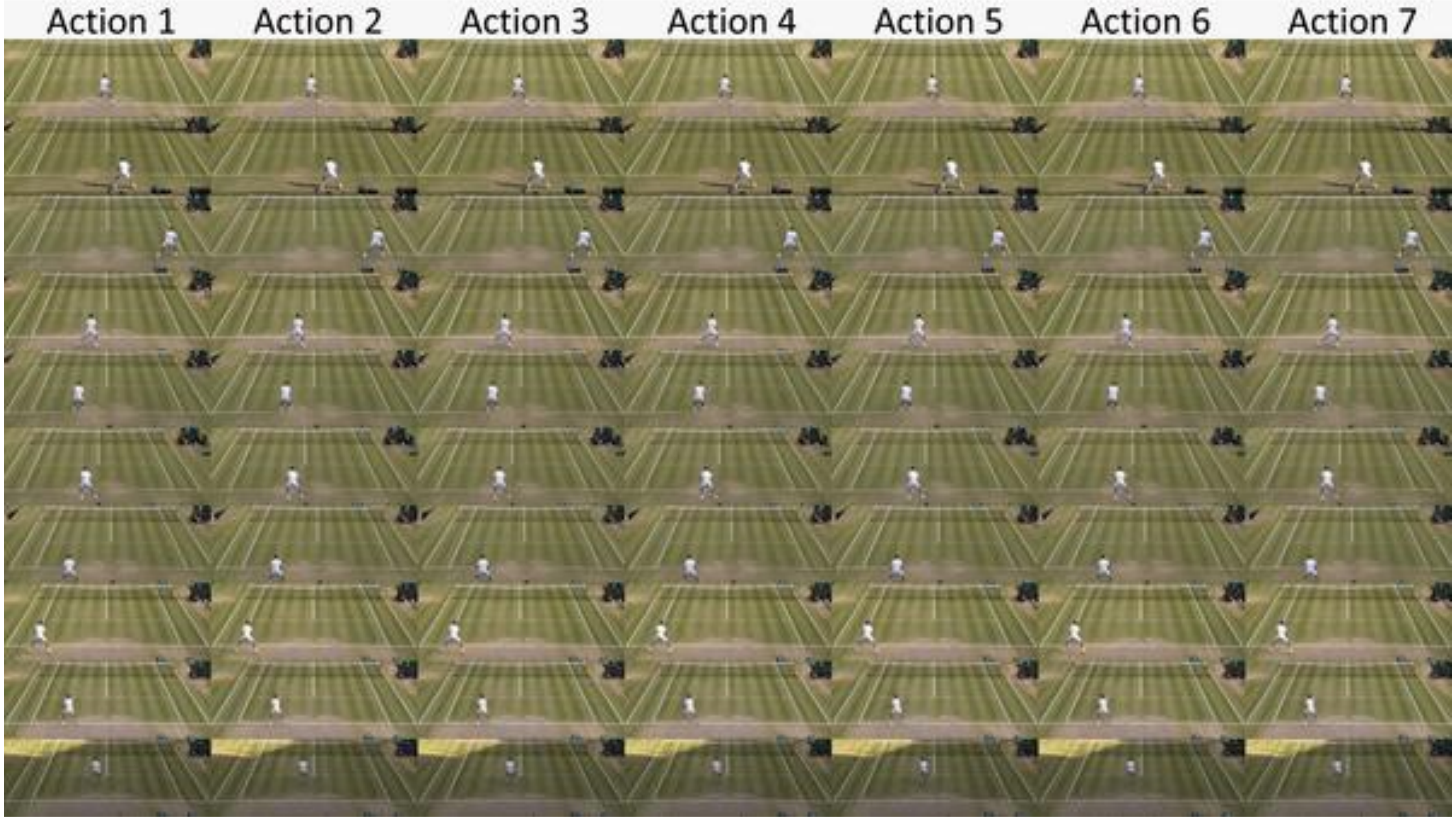
- **Speed**
- **Limb movement**

Expectation of distance from cluster centroids

t-SNE plot of d_t

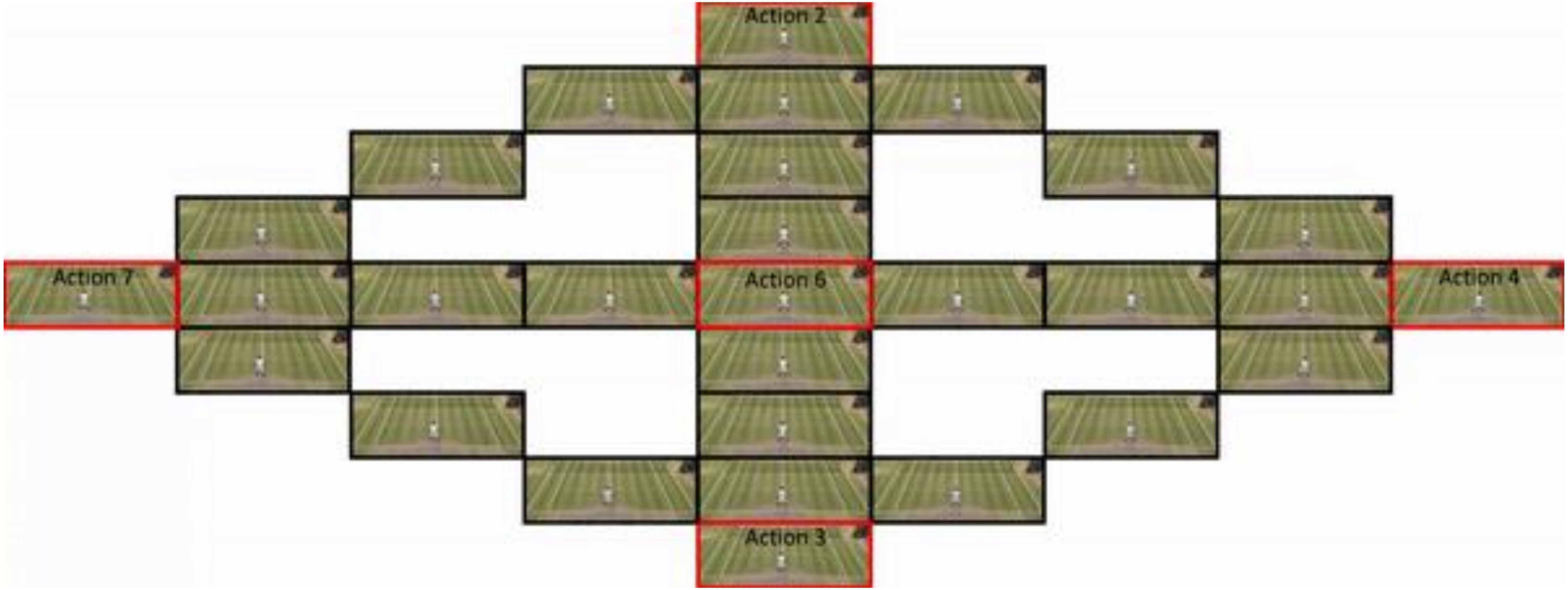
- For each d_t compute the expectation of its distance from the cluster centroids: variability embedding v_t => the specific way in which an action is performed

Results



- We learn a wide range of actions. The meaning of actions is consistent, independently from the starting frame the action is applied to

Action Interpolation



- At inference, we typically pose $v_t = 0$ and let the user specify actions a_t at each time step
- v_t can also be obtained from an action direction d_t that moves between the centroids of different actions: it is possible to generate a variety of different movement directions, eg. diagonal movements

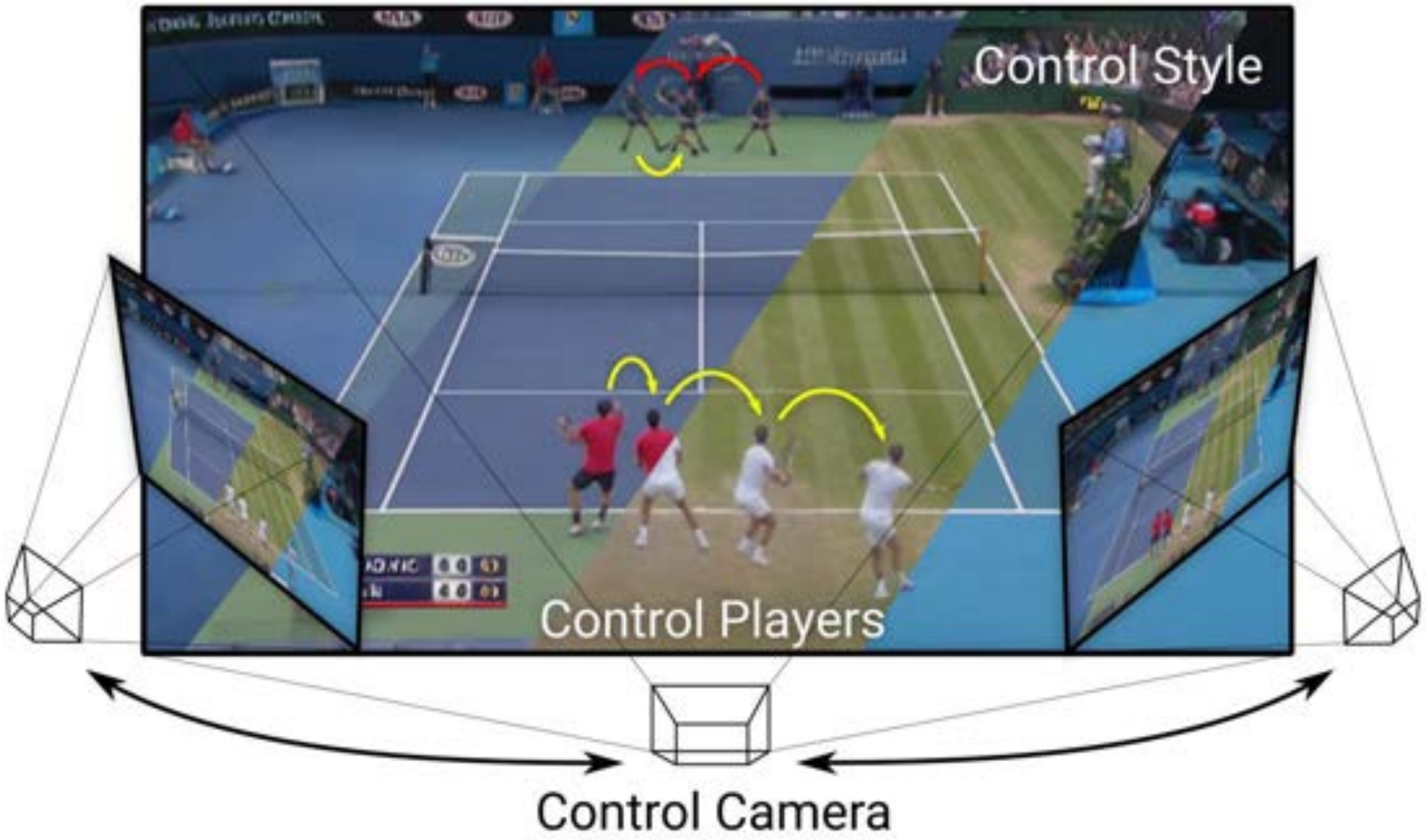

Action

Playable Environments

- Menapace, et al., “Playable Environments: Video Manipulation in Space and Time”, CVPR22

https://github.com/willi-menapace/PlayableEnvironments

Playable Environments



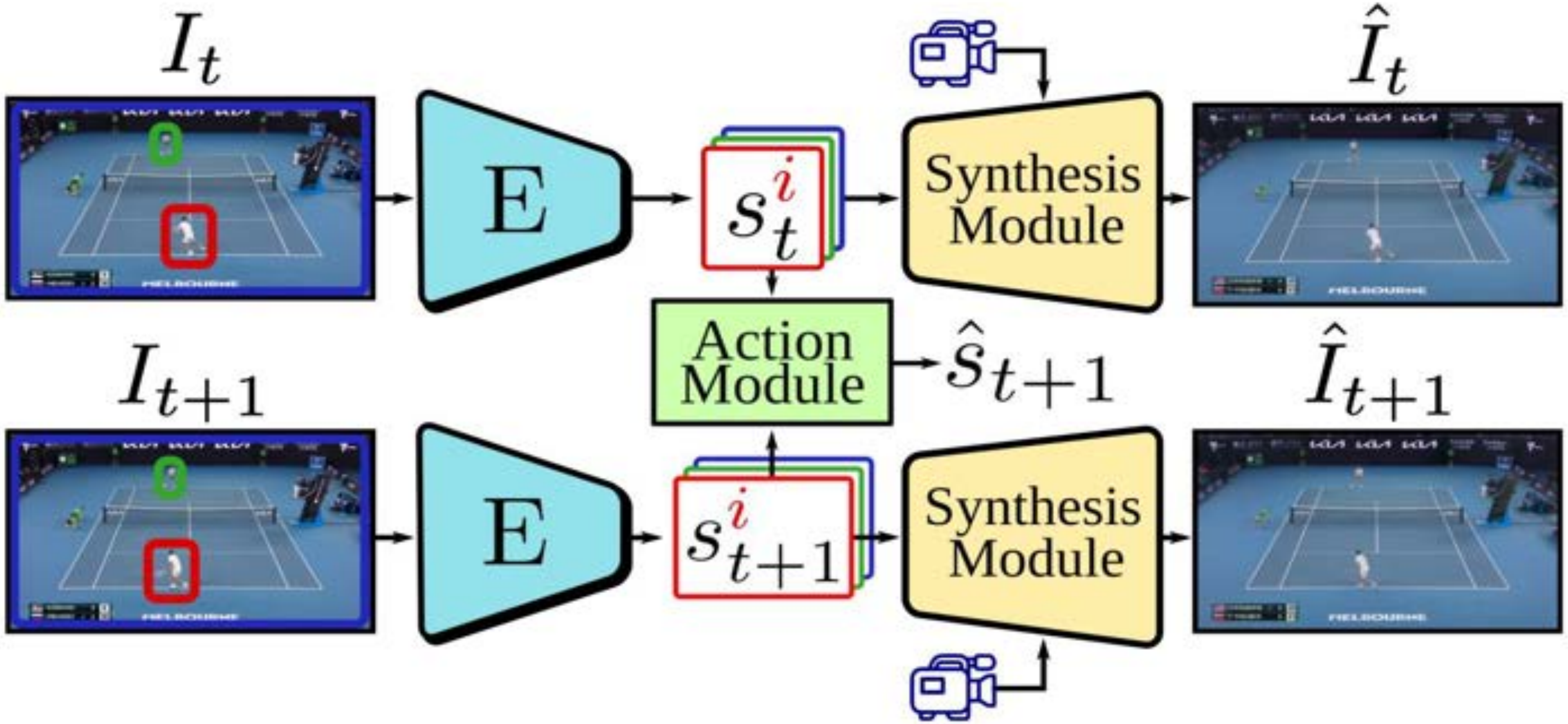
- Learn a model that represents the observed environment
- Allow the user to input actions to the model through a controller at test time

Playable Environments

Playable Environments

Playable Environments

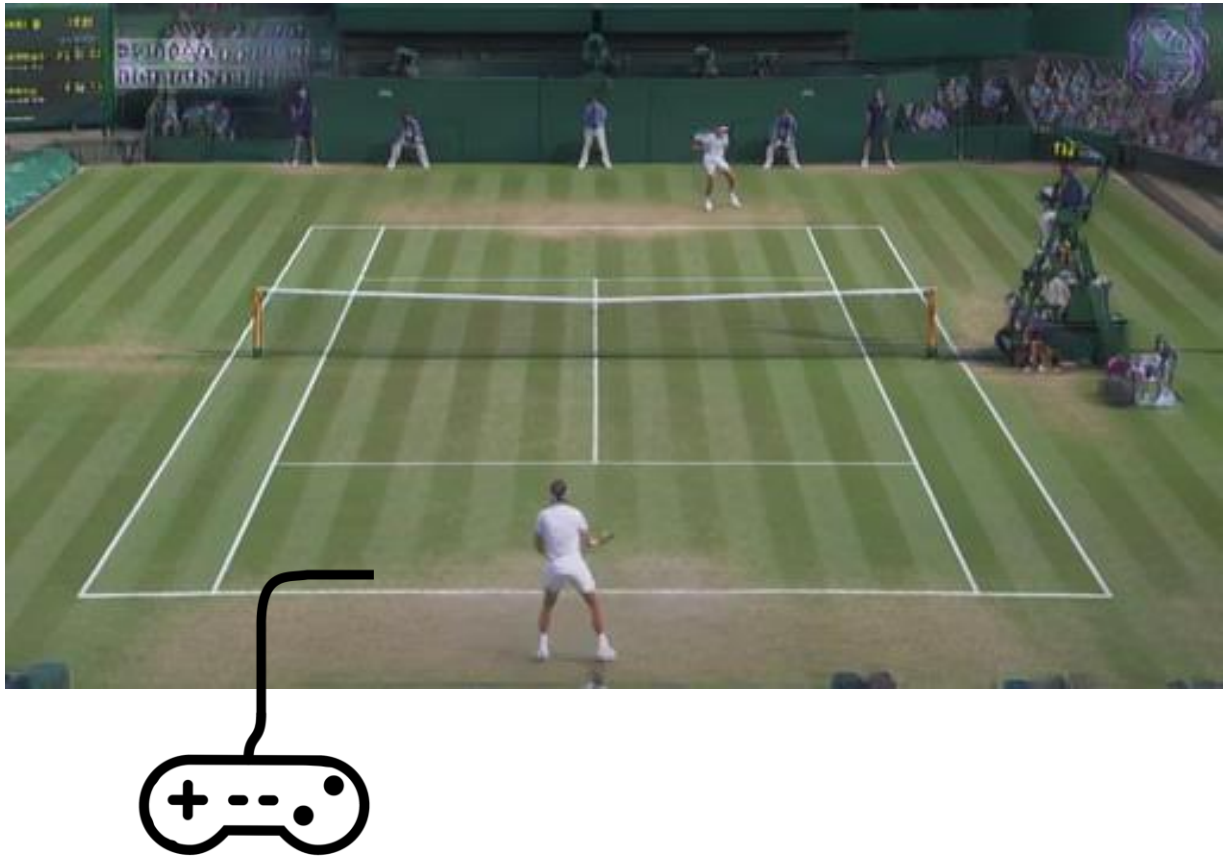
Framework



I_t
E
s_t^i
Synthesis Module
$\hat{I}_t$
Action Module
$\hat{s}_{t+1}$
I_{t+1}
E
s_{t+1}^i
Synthesis Module
$\hat{I}_{t+1}$

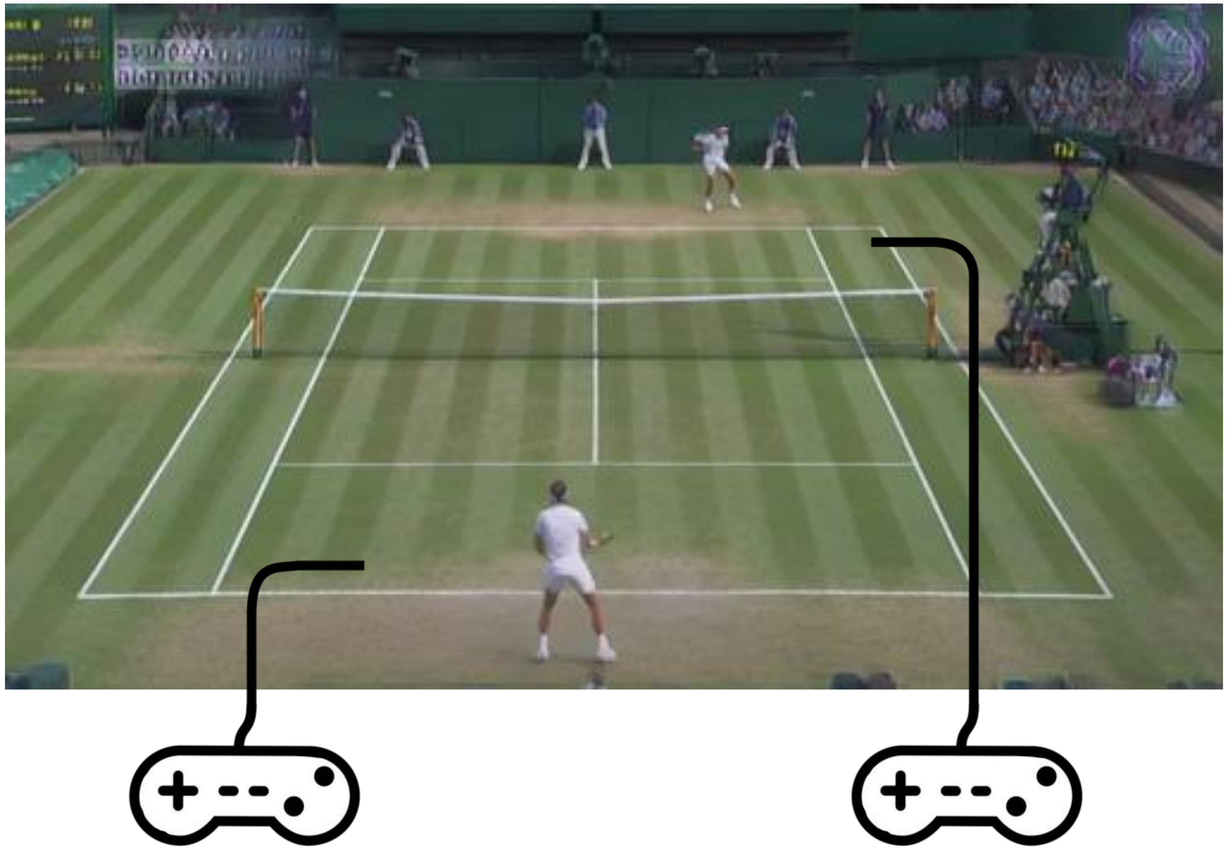
Framework Characteristics

1. Playability



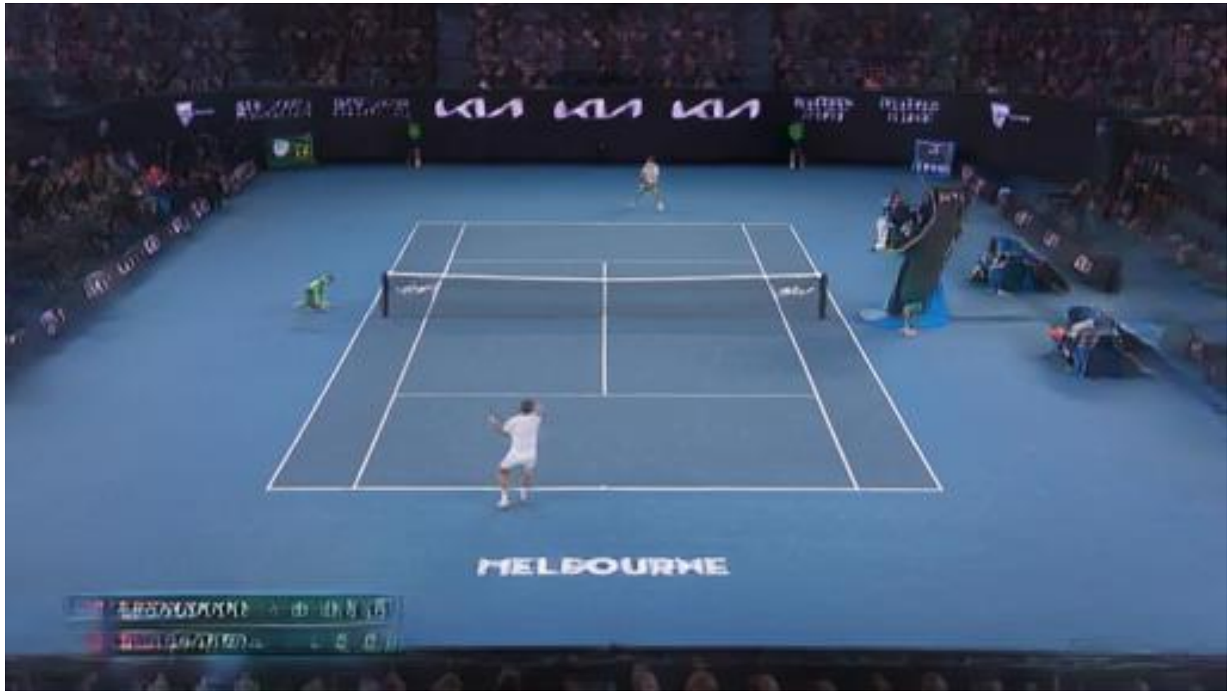
Framework Characteristics

1. Playability
2. Multi Object
3. Deformable Objects



Framework Characteristics

1. Playability
2. Multi Object
3. Deformable Objects
4. Camera Control



Framework Characteristics

1. Playability
2. Multi Object
3. Deformable Objects
4. Camera Control
5. Style Control
6. Robustness

Learned Actions

Learnable Game Engines (LGEs)

- Menapace, et al., “Plotting Behind the Scenes: Towards Learnable Game Engines”, arxiv 2023
- Menapace, et al., “Promptable Game Models: Text-guided Game Simulation via Masked Diffusion Models”, ACM ToG 2024

https://learnable-game-engines.github.io/lge-website/

Related Work

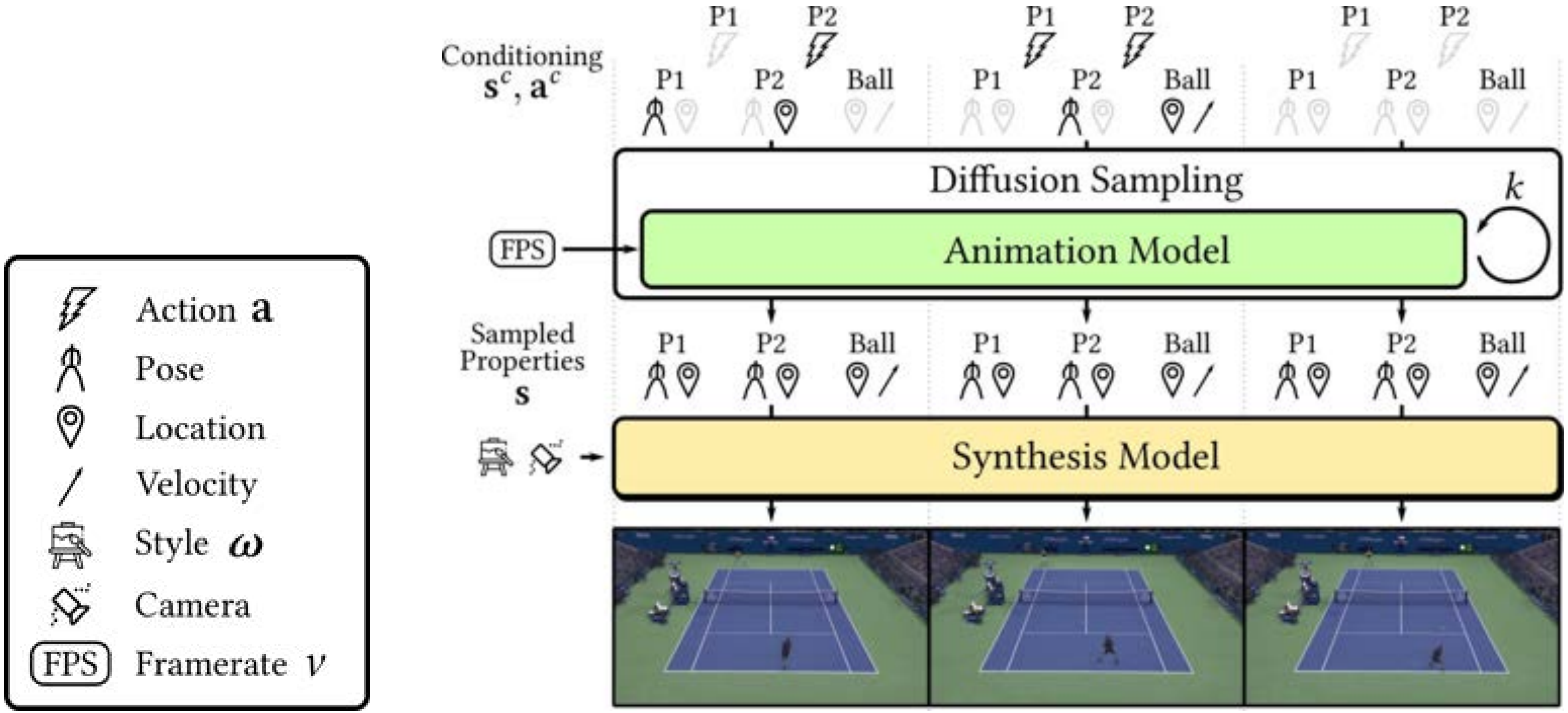
GameGAN
[Kim et. al, CVPR 2020]

Playable Video Generation
[Menapace et. al, CVPR 2021]

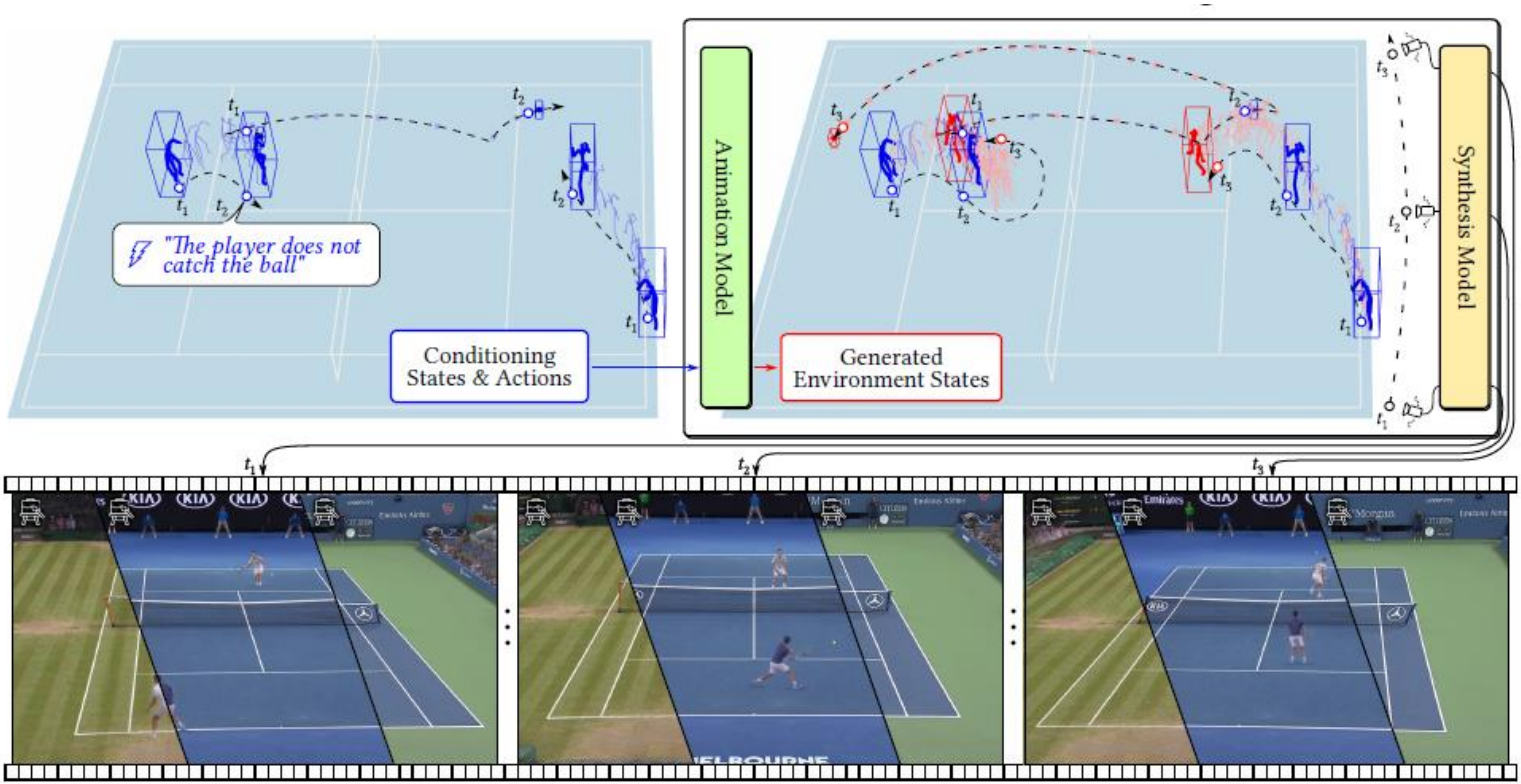
Playable Environments
[Menapace et. al, CVPR 2022]

Method

Two separately trained components:

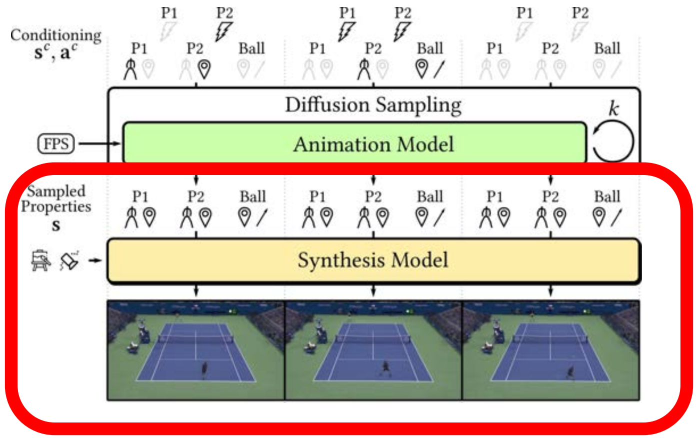


Method

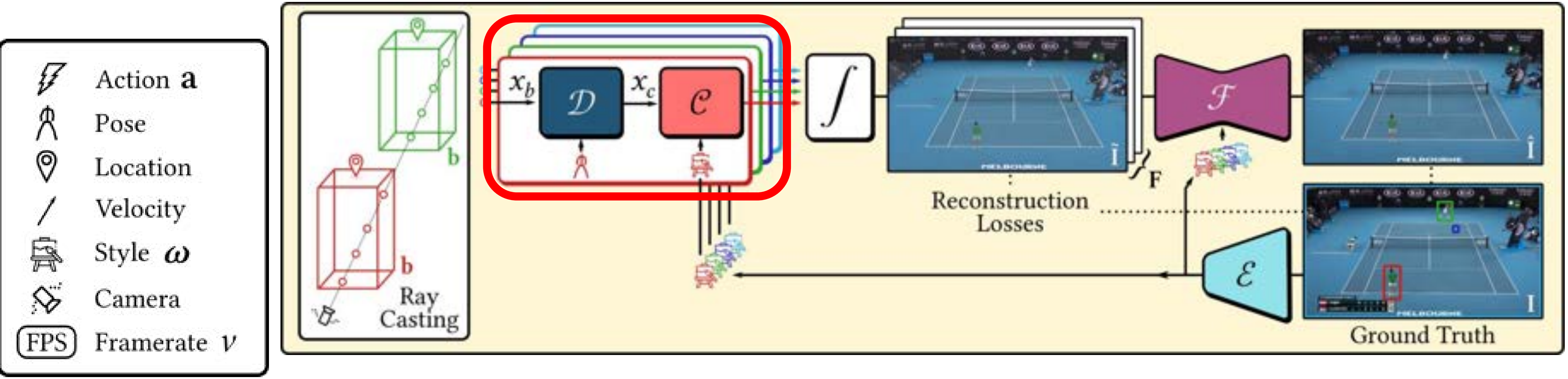


"The player does not catch the ball"
Conditioning States & Actions
Animation Model
Generated Environment States
Synthesis Model

Synthesis Module

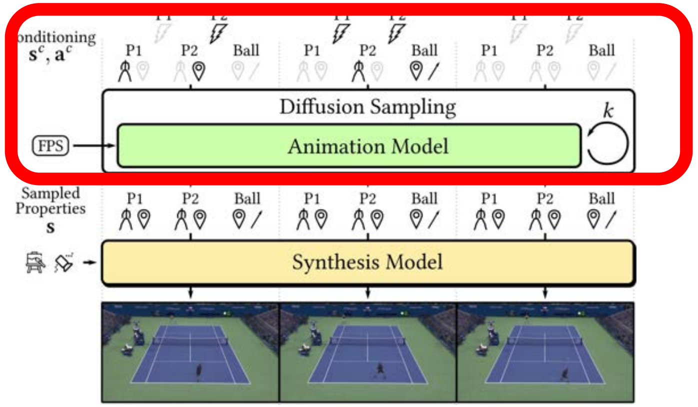


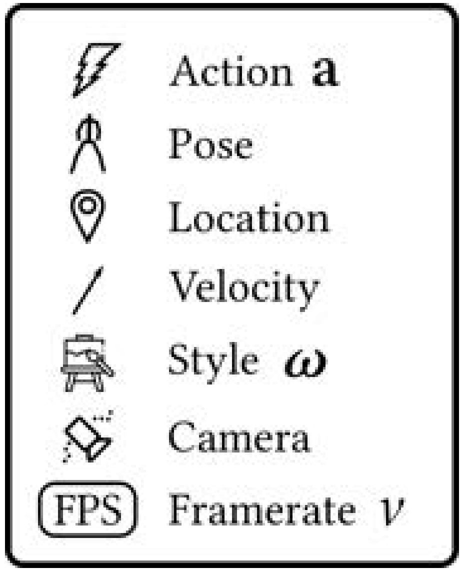
- NERF-based: renders the state of the environment from a given viewpoint
- A composition of NERFS, one for each object
- The model is trained using L2 and perception reconstruction losses

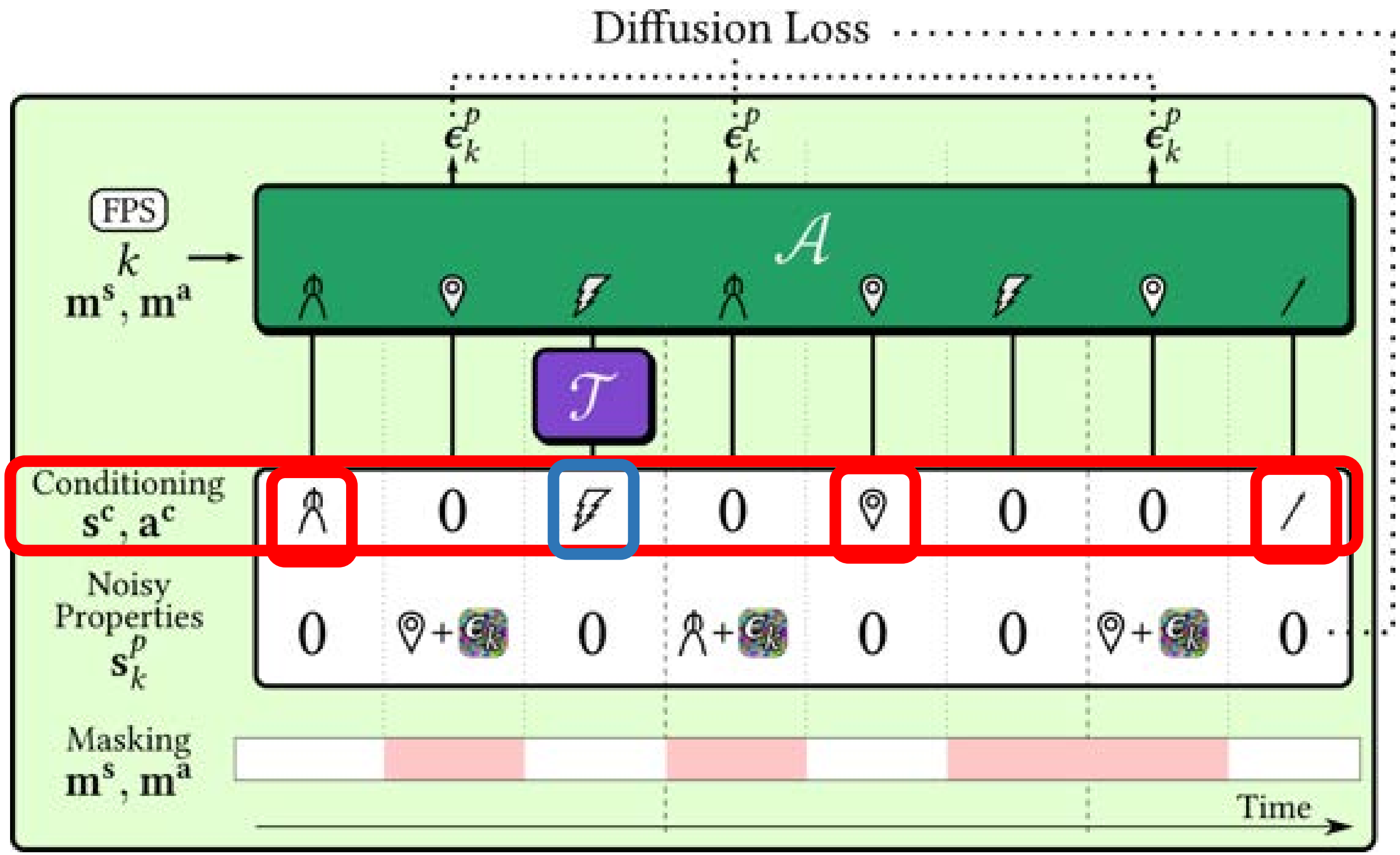


Animation Module

- Diffusion-based: produces sequences of states based on conditioning signals
  - Values: pose, location, velocity of a player or the ball
  - Natural language: what a player is doing

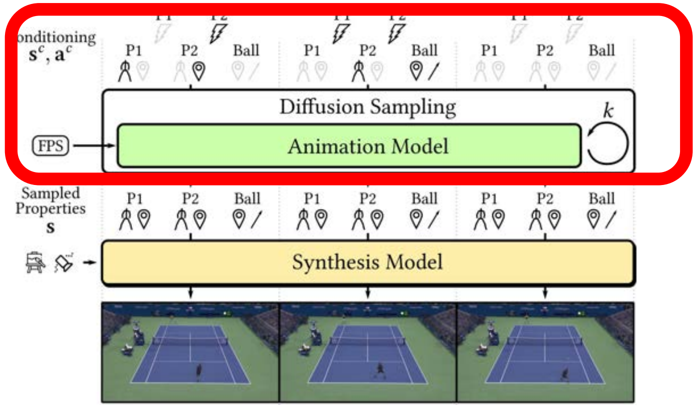


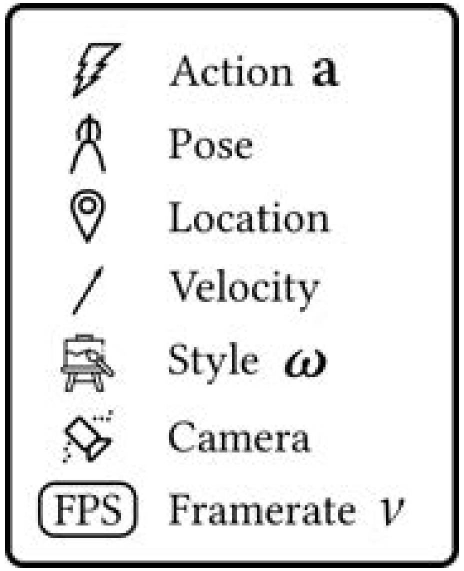


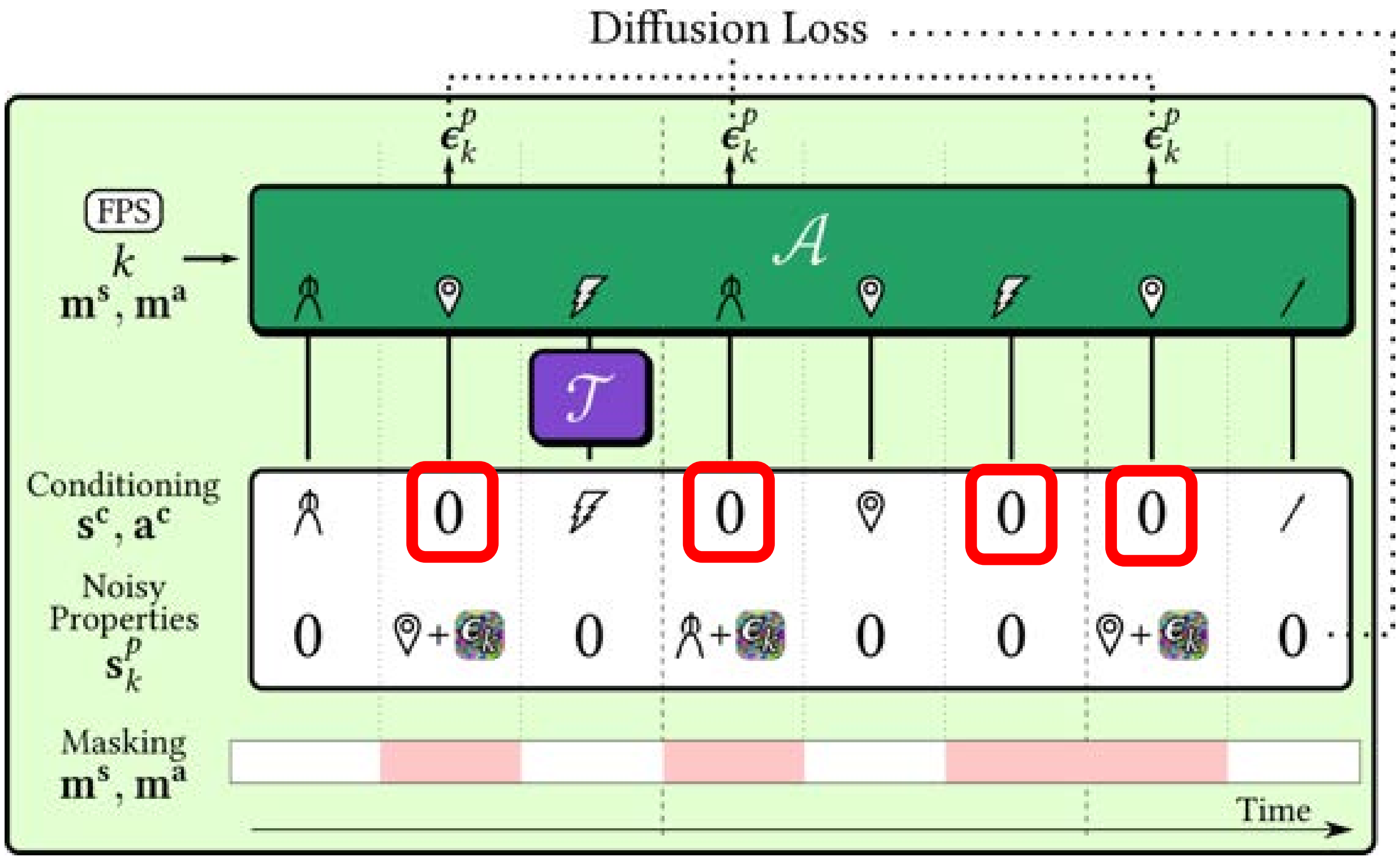


Animation Module

- The conditions are optional: the model can be used at inference time for different task by changing the structure of the conditioning

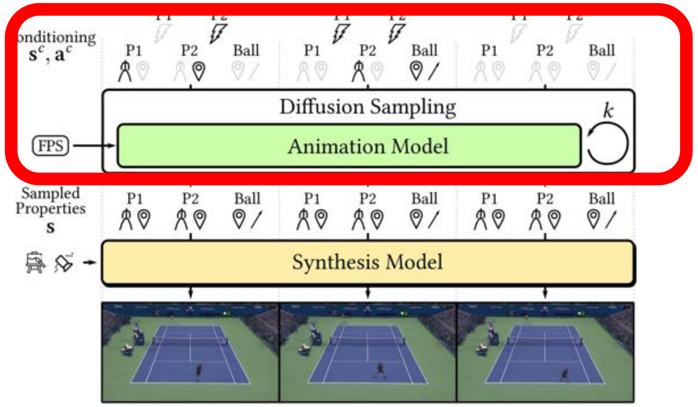


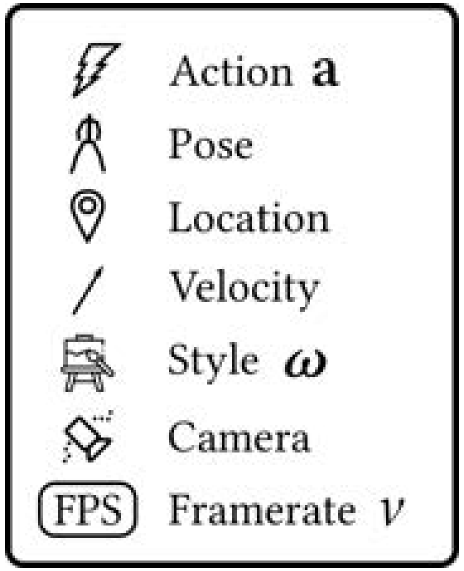


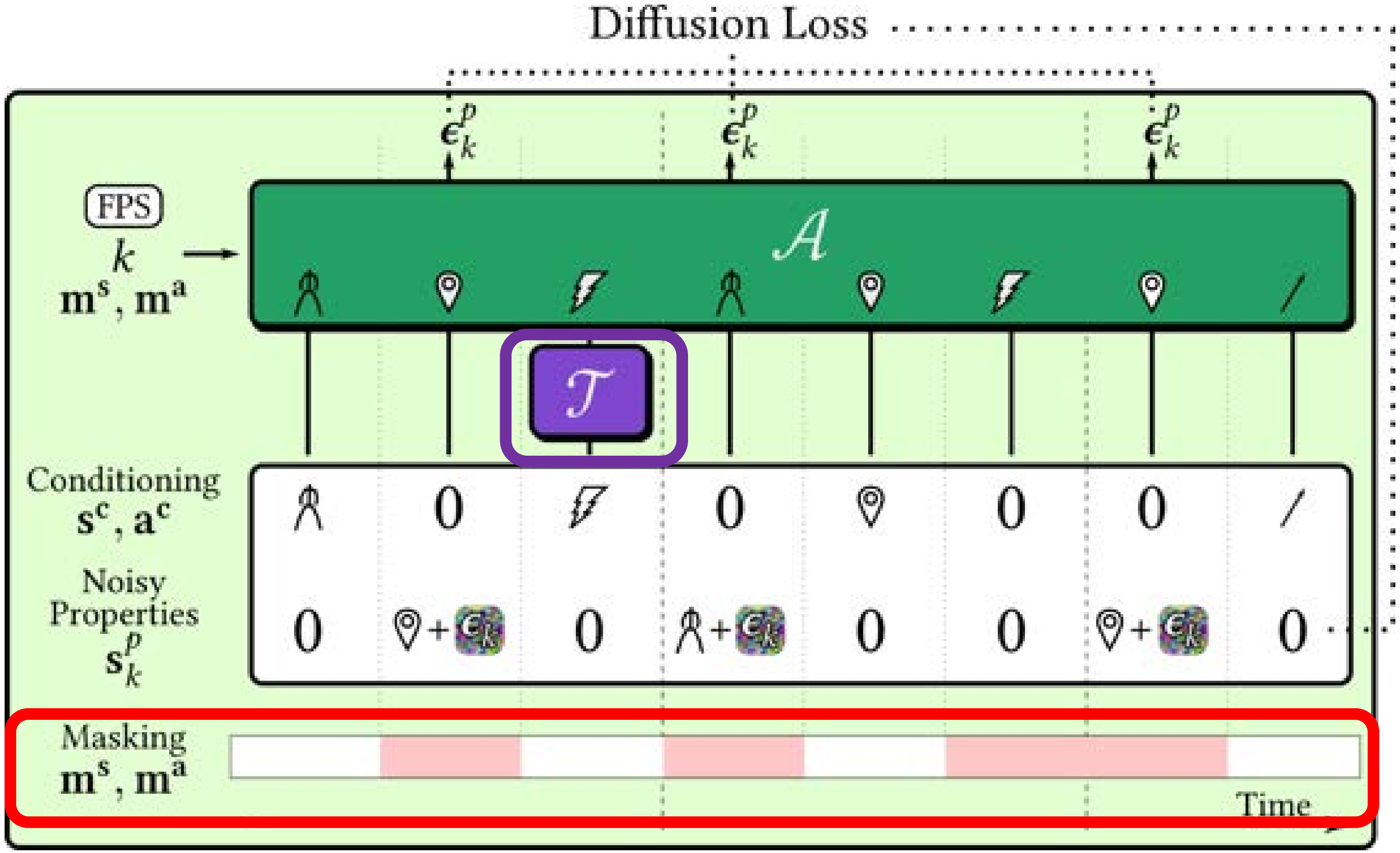


Animation Module

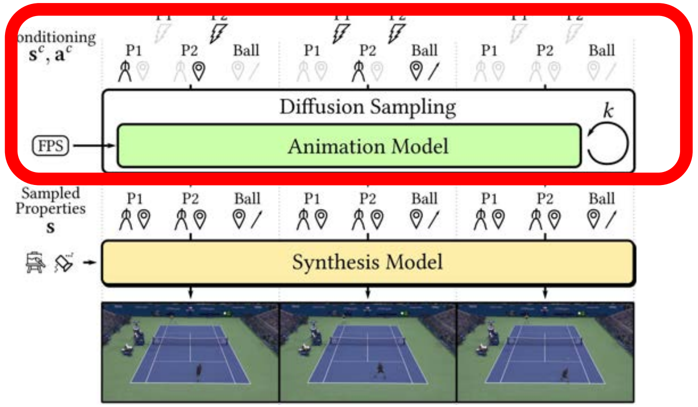
- The model is based on a transformer architecture where a frozen T5 encodes the natural language conditioning
- A mask specifies which part of the input serves as conditioning and which needs to be predicted

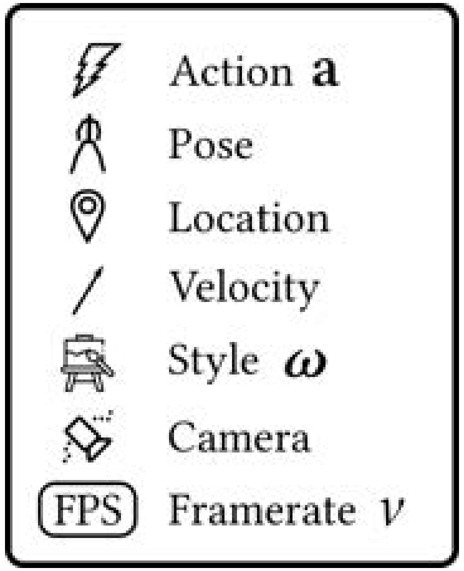




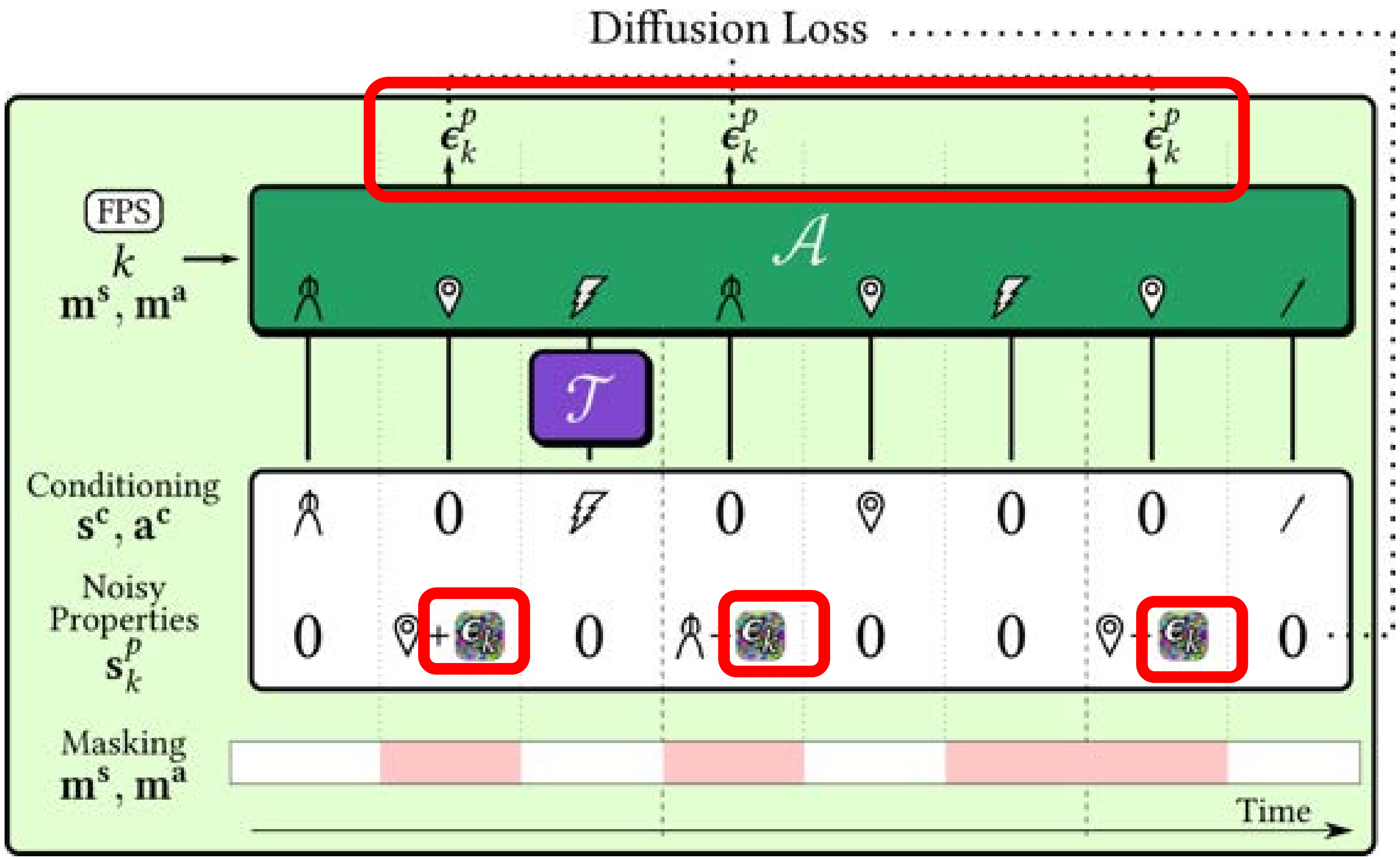


Animation Module

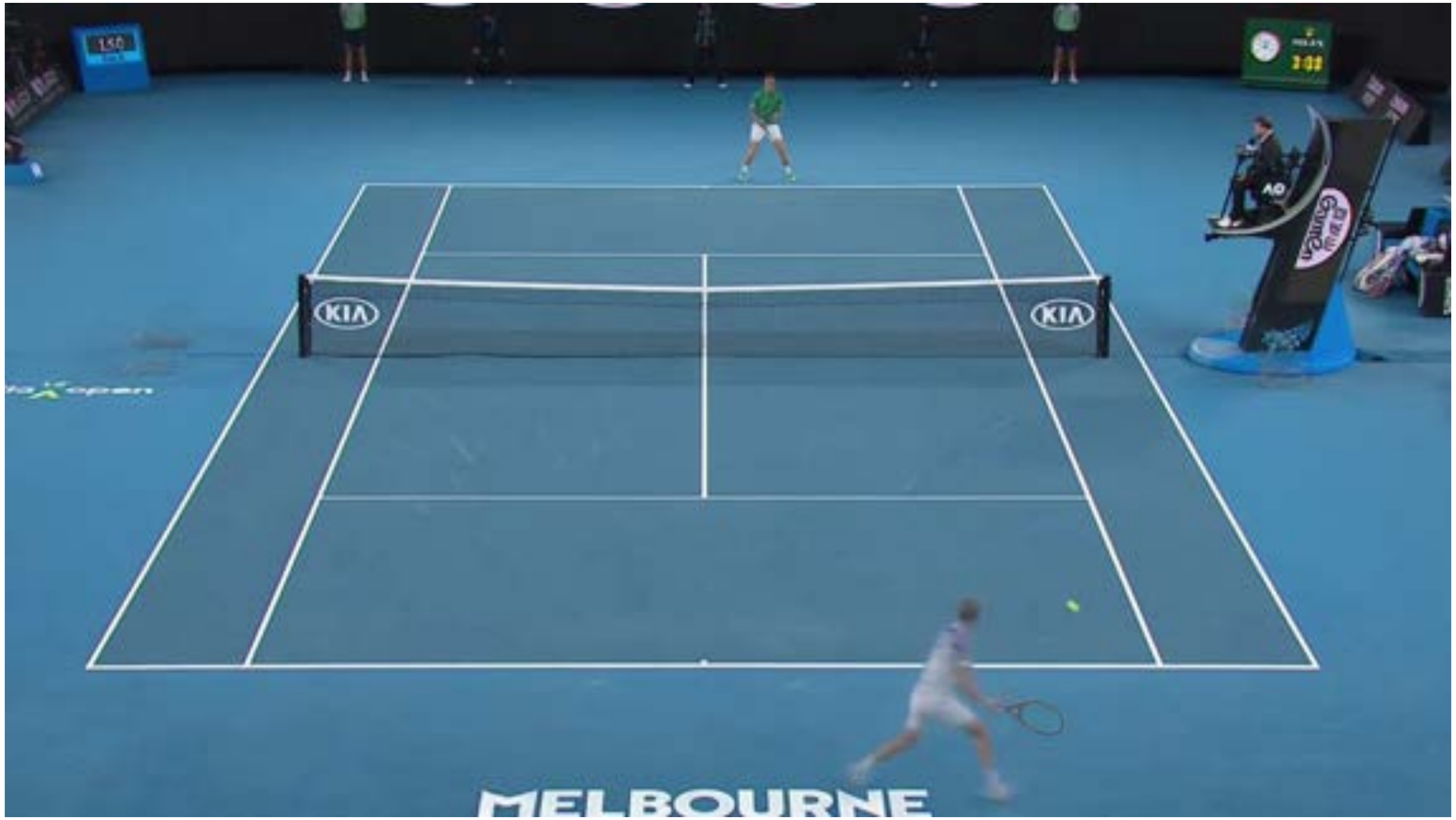




- Finally, the model is trained to predict noise applied to the sequence



Controllable Synthesis



Text-Controllable Animation

Learnable Game Engines:

- Understand physics and game logic
- Can receive action inputs expressed with natural language

Text-Controllable Animation

Designing Game Strategy

Designing Game Strategy

Making the player win:

- Reconstruct the scene
- Devise winning actions
- Animate players
- Render the results

Designing Game Strategy

Designing Game Strategy

the player serves and sends the ball to the right service box

The player stands still waiting for a serve

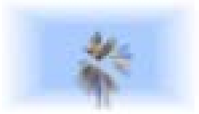
Original video = Bottom player loses

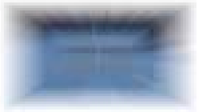
the player serves and sends the ball to the right service box

The player stands still waiting for a serve

½ Original video + “The [TOP] player doesn’t catch the ball”= Bottom player wins

Play LGEs as Videogames





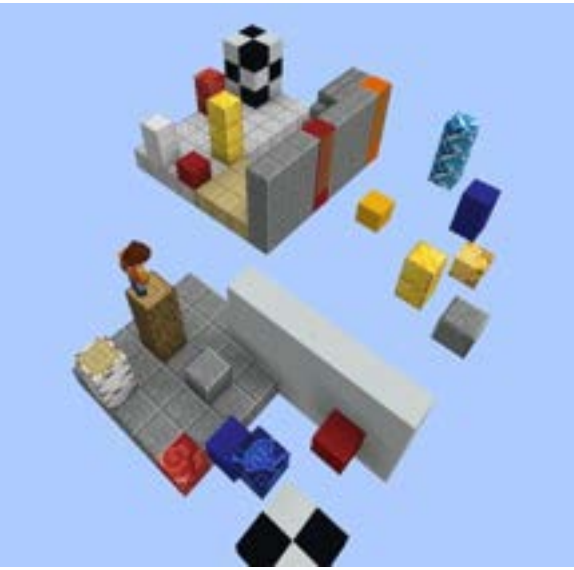
Director’s Mode

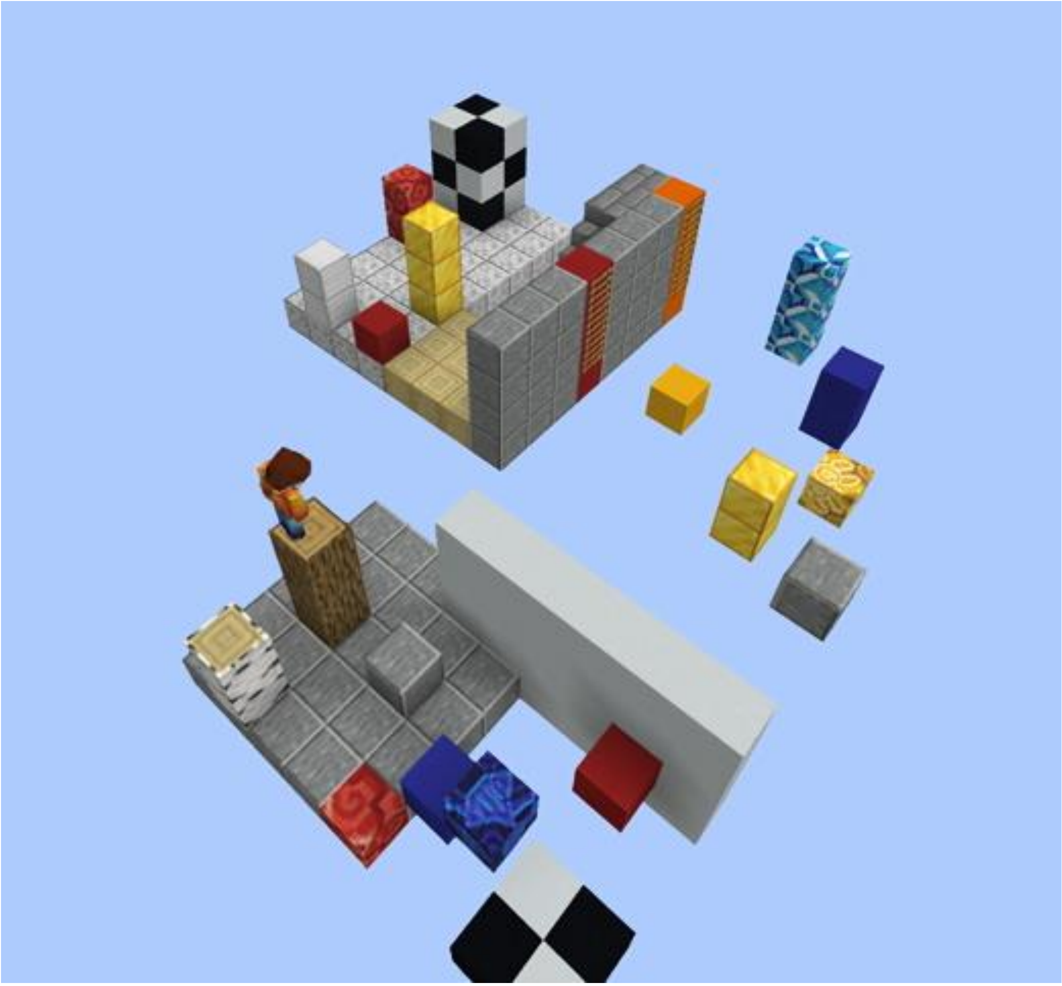
Constrain generation using:

- Desired values of the environment states
- Actions expressed with natural language

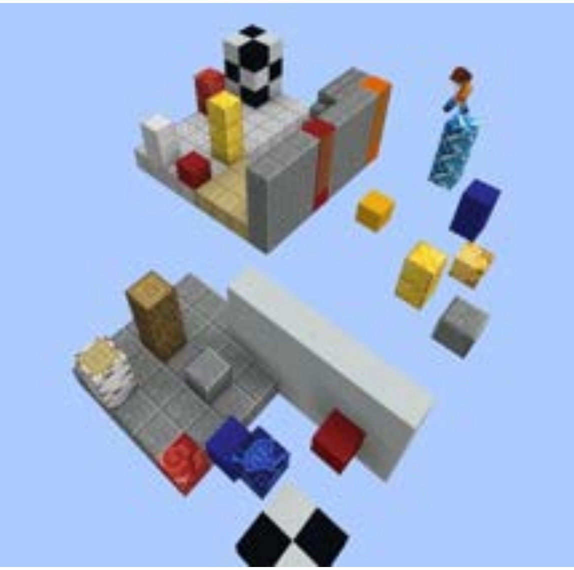
Director's Mode

First Frame

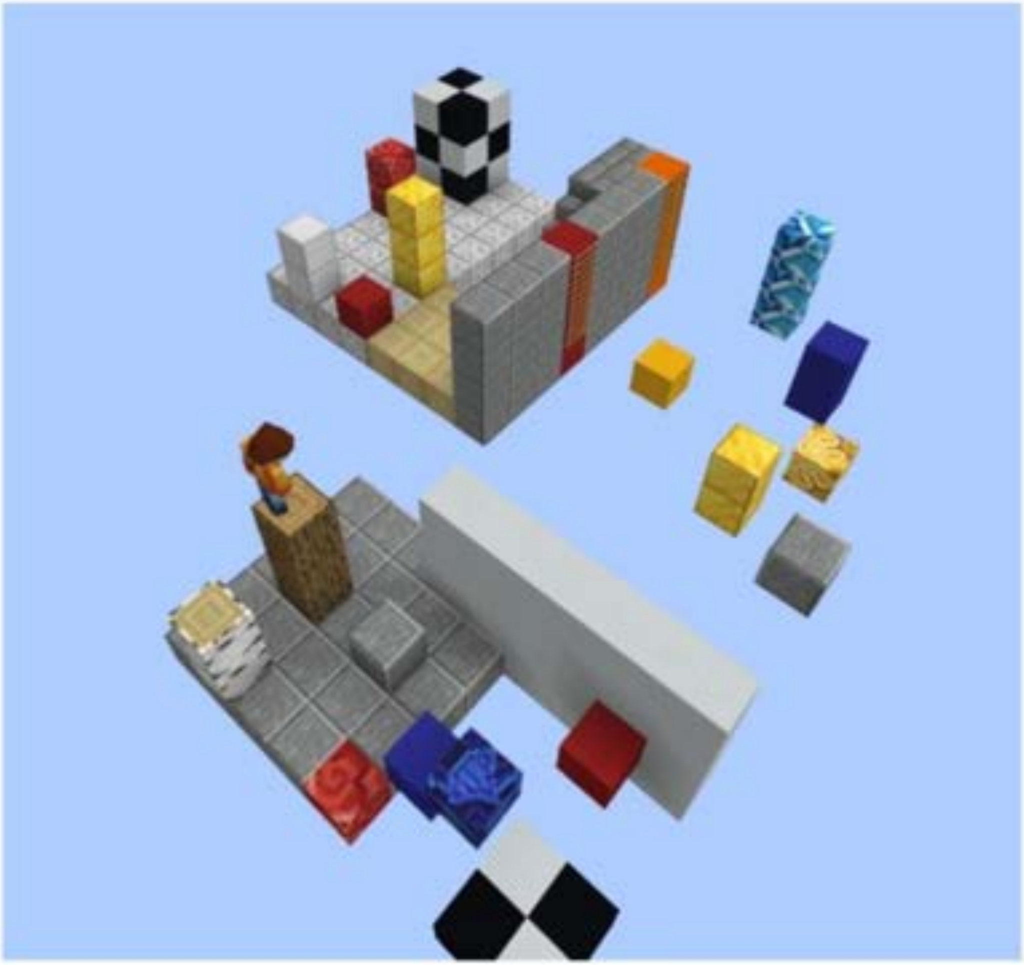


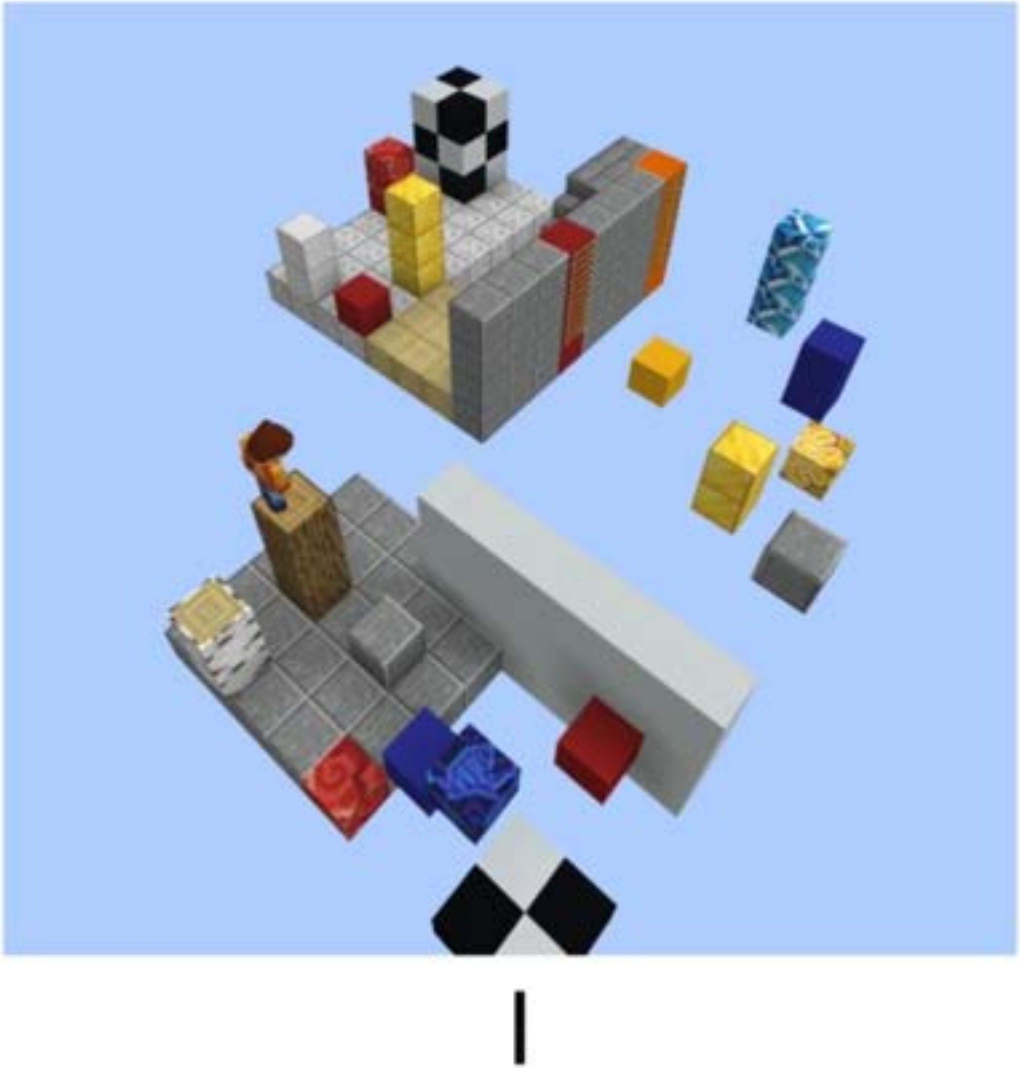


Last Frame



Director's Mode





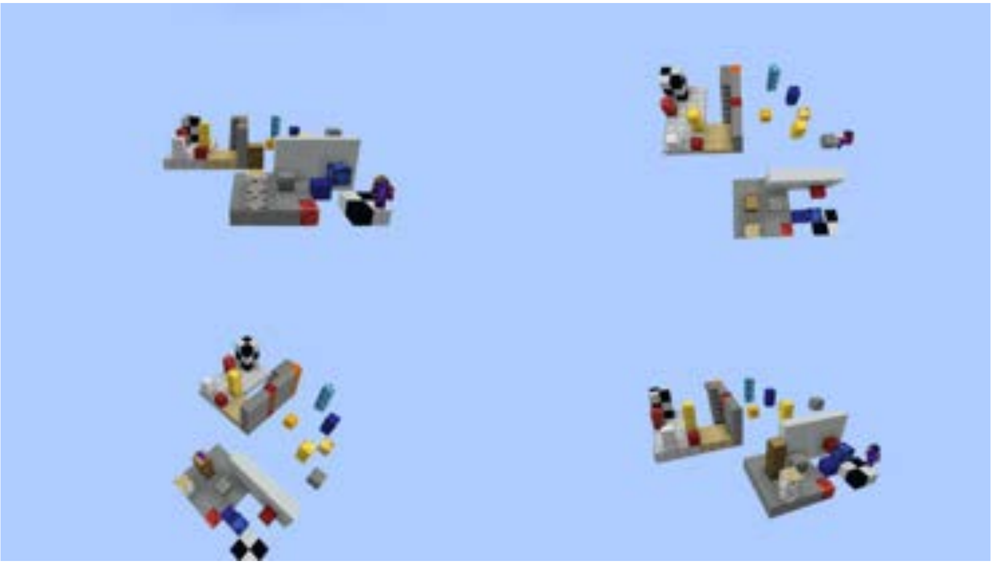
Director’s Mode

The conditioning is flexible, e.g., give multiple actions to constrain the solution

LGE Datasets

Tennis

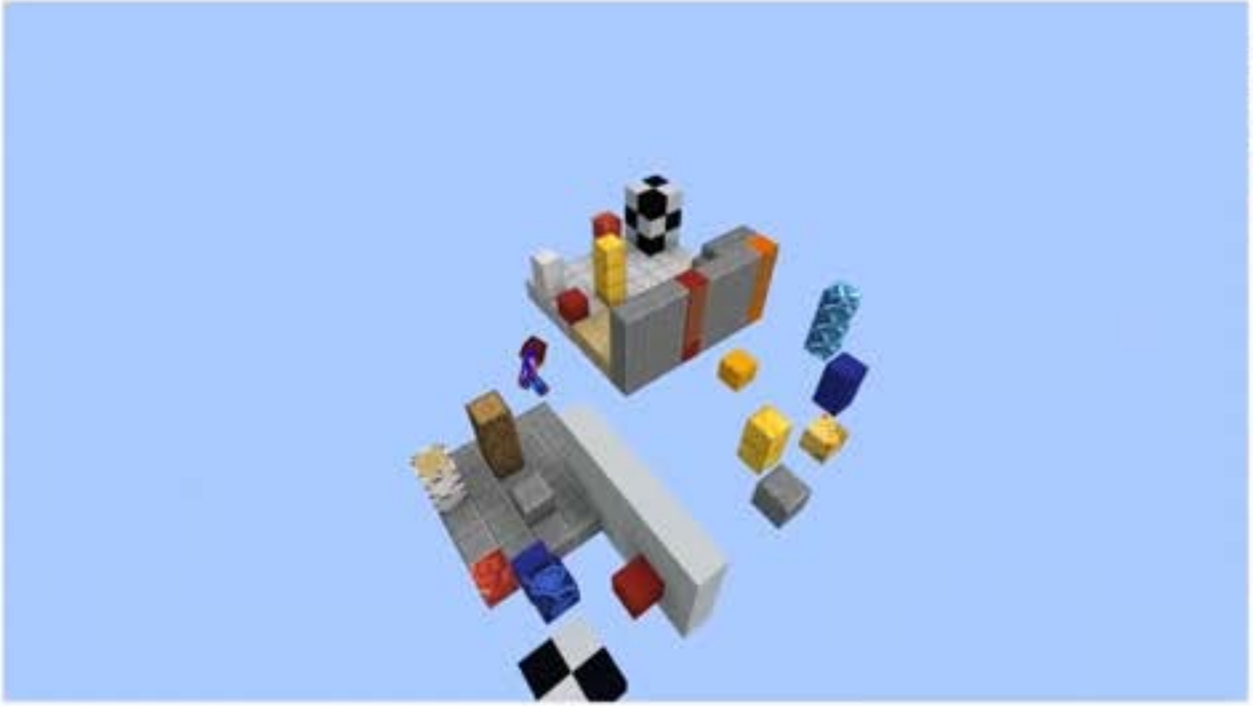
- 7112 video sequences at 1920x1080@25fps
- 15.5 hours of videos
- 1.12M fully annotated frames
- 25.5k unique captions



Minecraft

- 61 video sequences at 1024x567@20fps
- 1.2 hours of videos
- 68.5k fully annotated frames
- 1.24k unique captions

LGE Datasets



Minecraft

Tennis

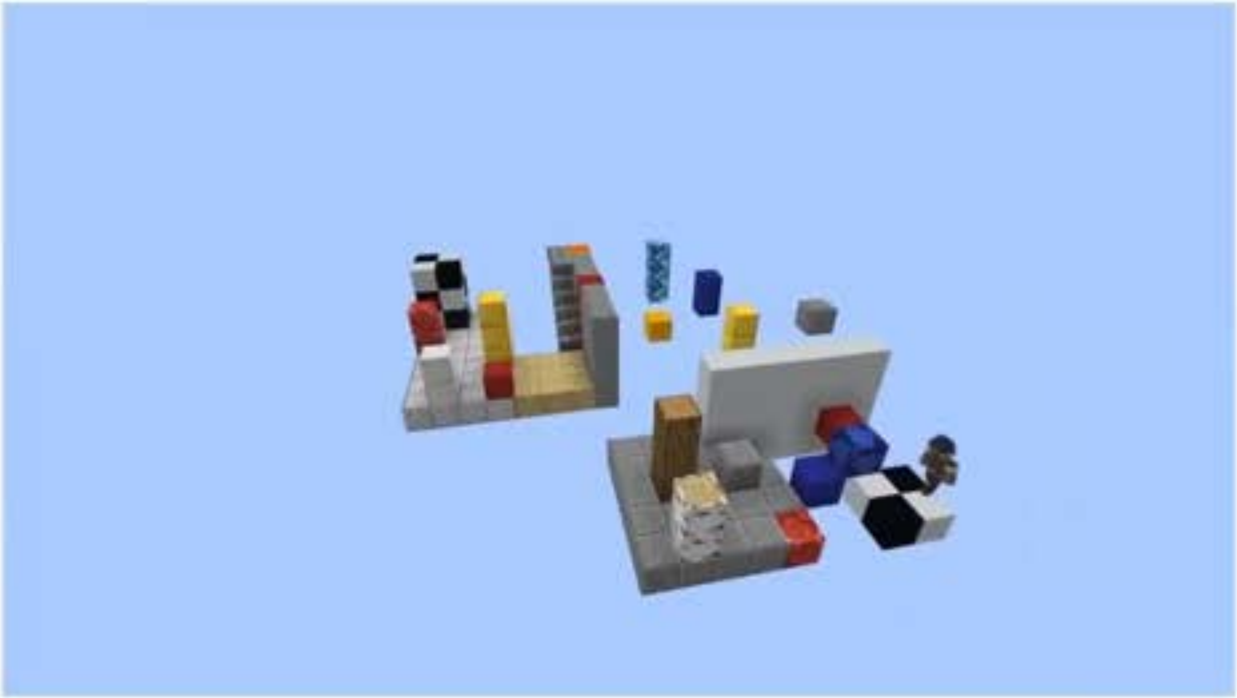
Synthesis Model Evaluation

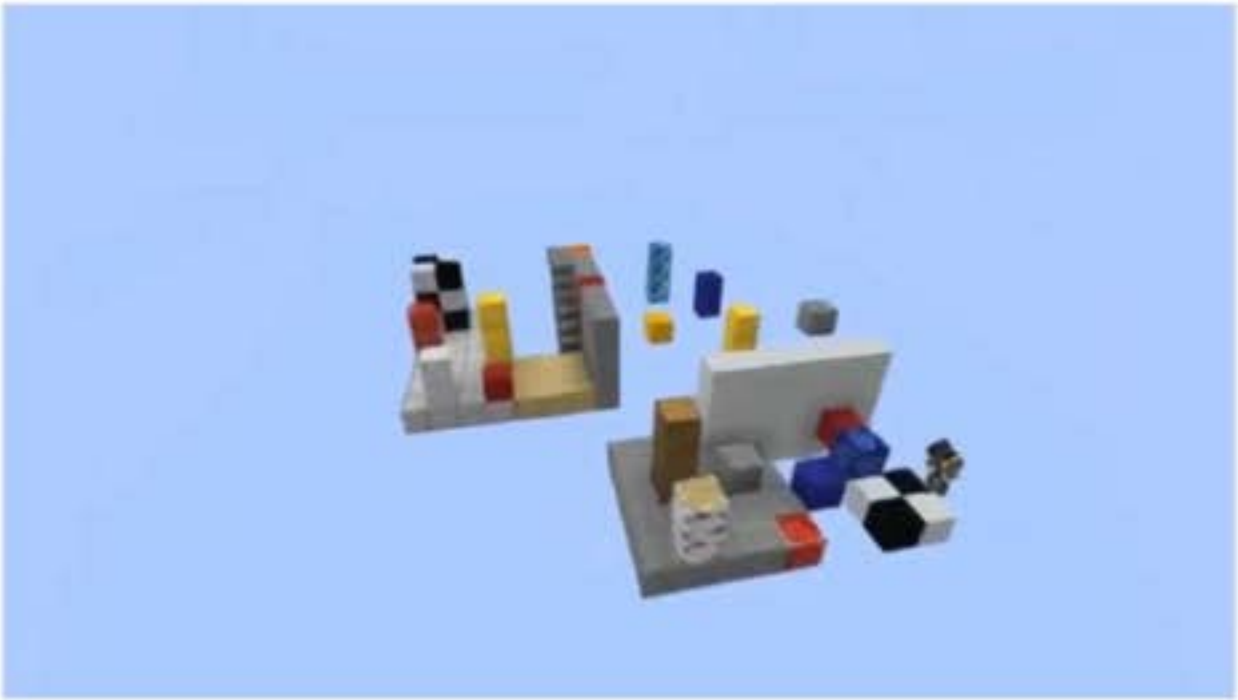
Learnable Game Engines

- Increased resolution
- No checkerboard artifacts

Playable Environments

Synthesis Model Evaluation





Learnable Game Engines

Playable Environments

- Increased resolution
- No checkerboard artifacts

Animation Model Evaluation

Learnable Game Engines

Playable Environments

- Higher quality and higher frame rate sequences
- Better scene dynamics

Beyond Playable Environments

- Can we generate large scenes with manipulable objects inside?
- Can we do that without object localization and camera calibration?
- This environment representation can be used to model complex games with many objects and large environment

Beyond Playable Environments

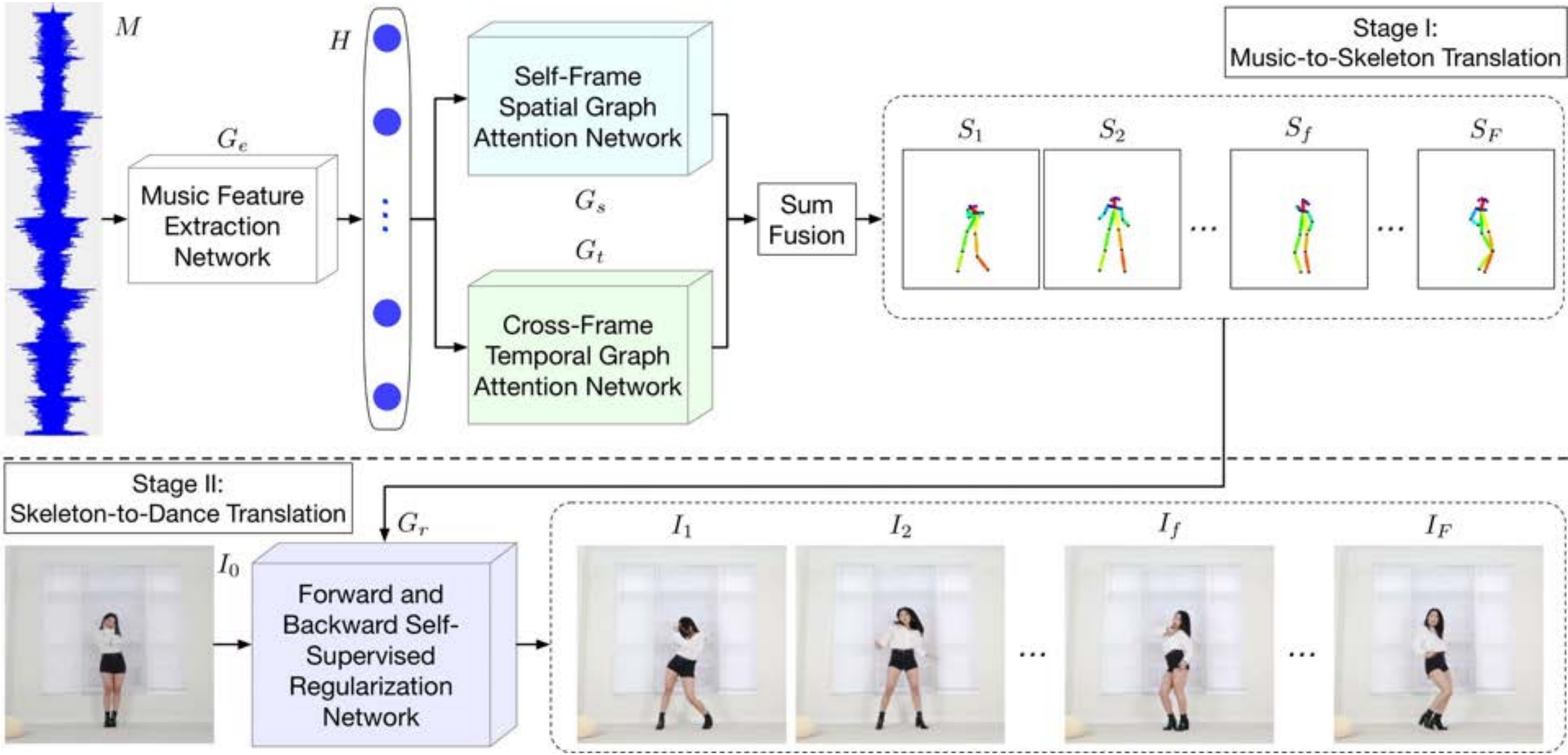
A Corgi dog riding a bike in Times Square wearing sunglasses and a beach hat

A cowboy panda riding on the back of a lion, hand-held camera

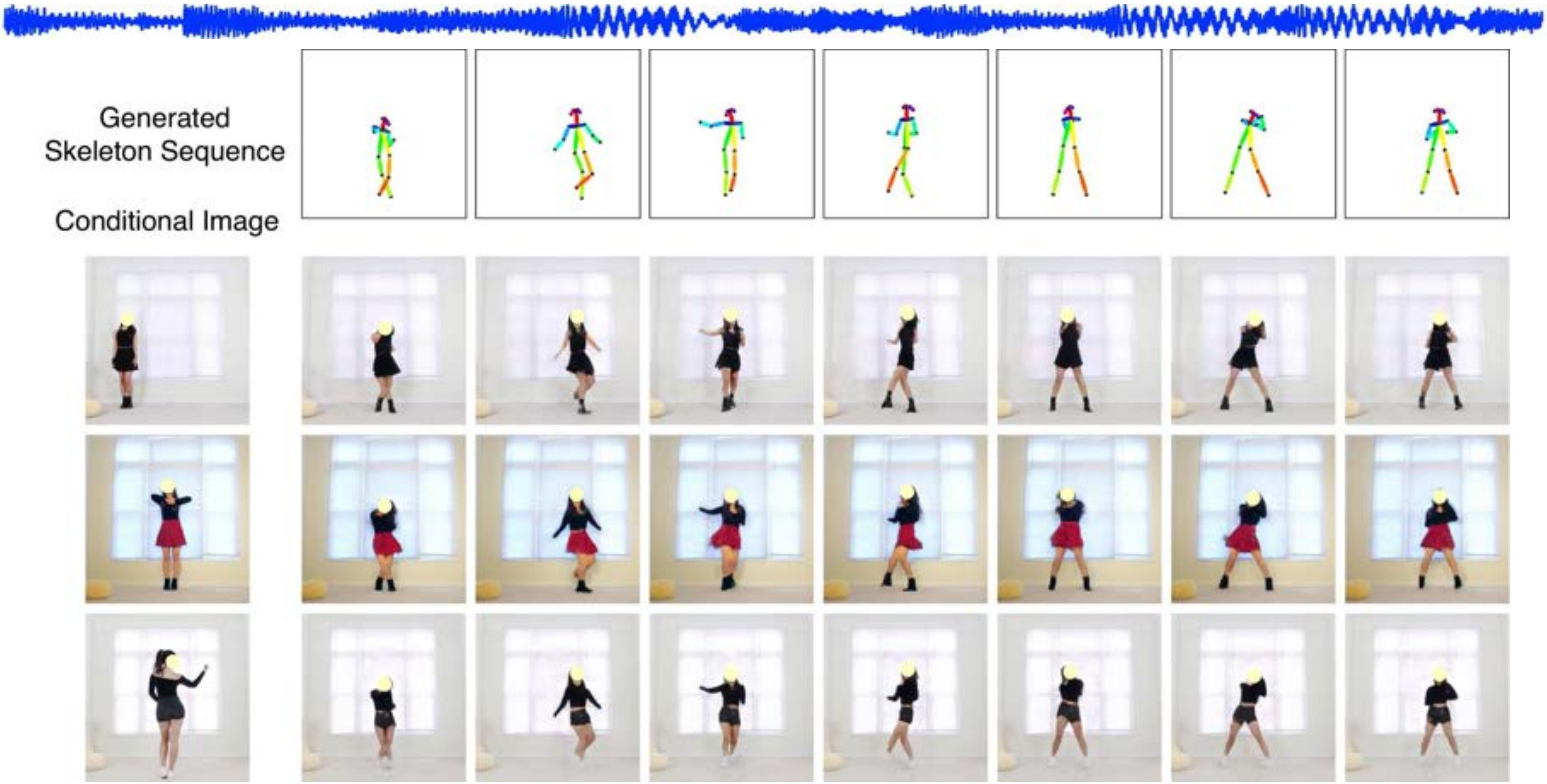
Menapace, et al., "Snap Video: Scaled Spatiotemporal Transformers for Text-to-Video Synthesis", CVPR24

Music-Guided Dance Video Synthesis

DanceGAN


M
G_e
Music Feature Extraction Network
H
Self-Frame Spatial Graph Attention Network
G_s
G_t
Cross-Frame Temporal Graph Attention Network
Sum Fusion
Stage I: Music-to-Skeleton Translation
S_1
S_2
S_f
S_F
Stage II: Skeleton-to-Dance Translation
G_r
I_0
Forward and Backward Self-Supervised Regularization Network
I_1
I_2
I_f
I_F

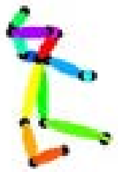
Music-Guided Dance Video Synthesis



Demo

Ballet

Real



Ours

K-pop

Real

Ours

Popping

Real

Ours

Where Are We Going Now …

- Incorporating 3D information
- Modeling complex interactions between actors and between actors and the scene
- Cross-modal seamless integration between text, audio, and visual information
- More attention to bias, privacy, and deep fakes detection
- …

Bias in Text-to-Image Models

A picture of a person in the kitchen

Stable Diffusion XL

Bias in Text-to-Image Models

A picture of a ~~person~~ in the kitchen
chef

Stable Diffusion XL

Bias in Text-to-Image Models

Text-to-image generative models may exhibit unexpected biases

- Given an attribute agnostic prompt
- The model may generate images with specific attributes (low diversity)

Fairness in AI

The increase usage of AI models raises **ethical** and **fairness** concerns

- Is the model performing well regardless of specific protected characteristics?
  - e.g., Age, Skin Color, Gender…

What is fairness in AI?

- The behavior of a deep learning model may exhibit biases against specific minority groups
  - The bias may be directly inherited from the training data
- We refer to fairness as the ability of the model to perform equally regardless of the protected characteristic

Bias in Face Attribute Classification

Task description:

- Given an image of a face
- Classify specific facial attributes
  - e.g., Straight Hair, Big Nose, etc.

The nature of the facial attributes may lead to unbalanced training sets:

- e.g., specific facial features may be more prone for specific protected characteristics

A classifier trained on such data will exhibit or amplify the training set bias [1,2,3,4]

[1] S. Jung, et al. Learning fair classifiers with partially annotated group labels, CVPR22
[2] P. Stock, M. Cisse. Convnets and imagenet beyond accuracy: Understanding mistakes and uncovering biases, ECCV18
[3] L. A. Hendricks, et al. Women also snowboard: Overcoming bias in captioning models, ECCV18
[4] Z. Wang, et al. Towards fairness in visual recognition: Effective strategies for bias mitigation, CVPR20

Bias Mitigation - Use Pre-trained Generative Models

Existing generative bias mitigation methods train generators from scratch[5,6,7]

- Requires domain specific data
- Hard to train (low quality)

Explore the usage of pre-trained generative models[8]

- Balance the original training-set
- Training-free method
- Data-collection free method

Main challenge:

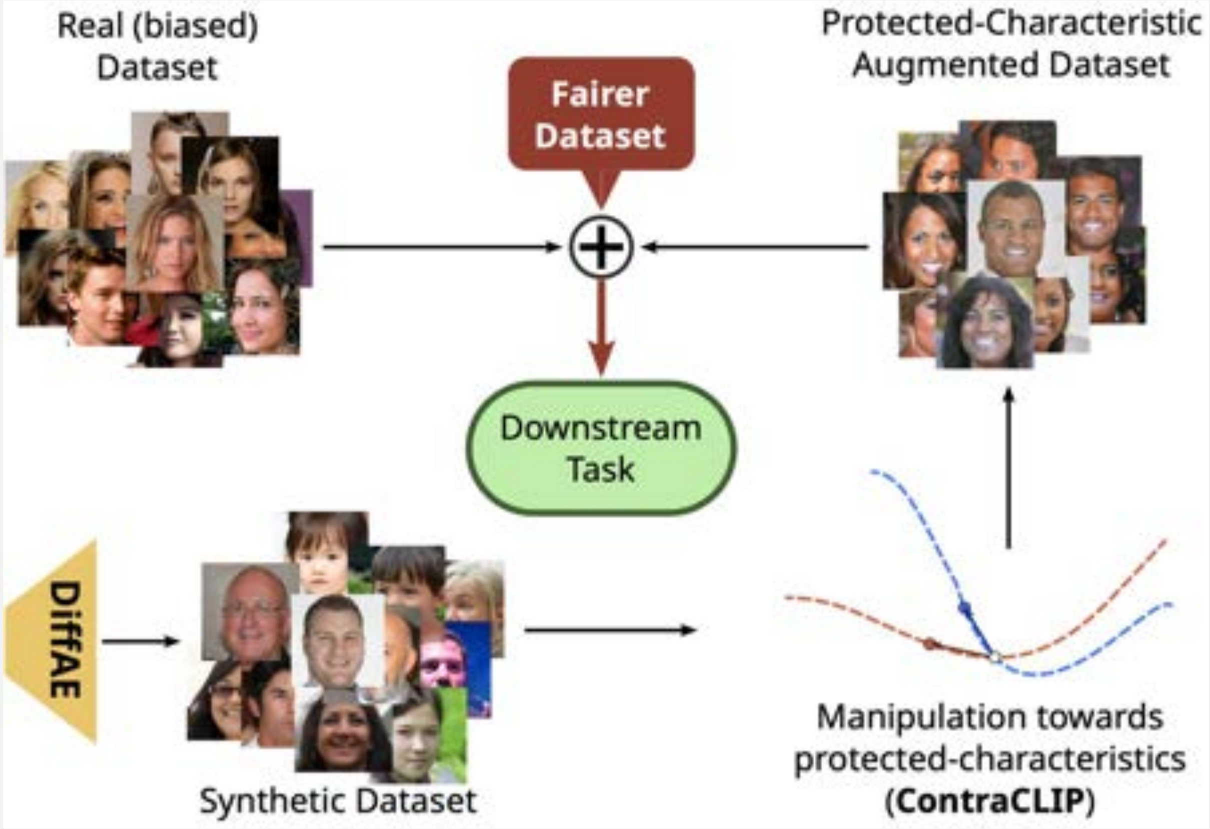
- The generator is itself biased
    - May not capture minority groups

[5] D. Xu, et al. FairGAN: Fairness-aware generative adversarial networks, 2018
[6] S. Dash, et al. Evaluating and mitigating bias in image classifiers: A causal perspective using counterfactuals, WACV22
[7] F. Zhang, et al. Fairness-aware contrastive learning with partially annotated sensitive attributes, ICLR23.
[8] M. D'Incà, et al. Improving Fairness using Vision-Language Driven Image Augmentation, WACV24

Bias Mitigation - Use Pre-trained Generative Models

Make a biased dataset fairer by augmenting it with generated images[9]:

- These images depict the desired protected characteristic (e.g., dark skinned people)
- They could be manipulated by a text-driven augmentation module (ContraCLIP [10])



[9] K. Preechakul, et al. Diffusion autoencoders: Toward a meaningful and decodable representation, CVPR22
[10] C. Tzelepis, et al., ContraCLIP: Interpretable GAN generation driven by pairs of contrasting sentences, 2022

Overcome the Generator Bias

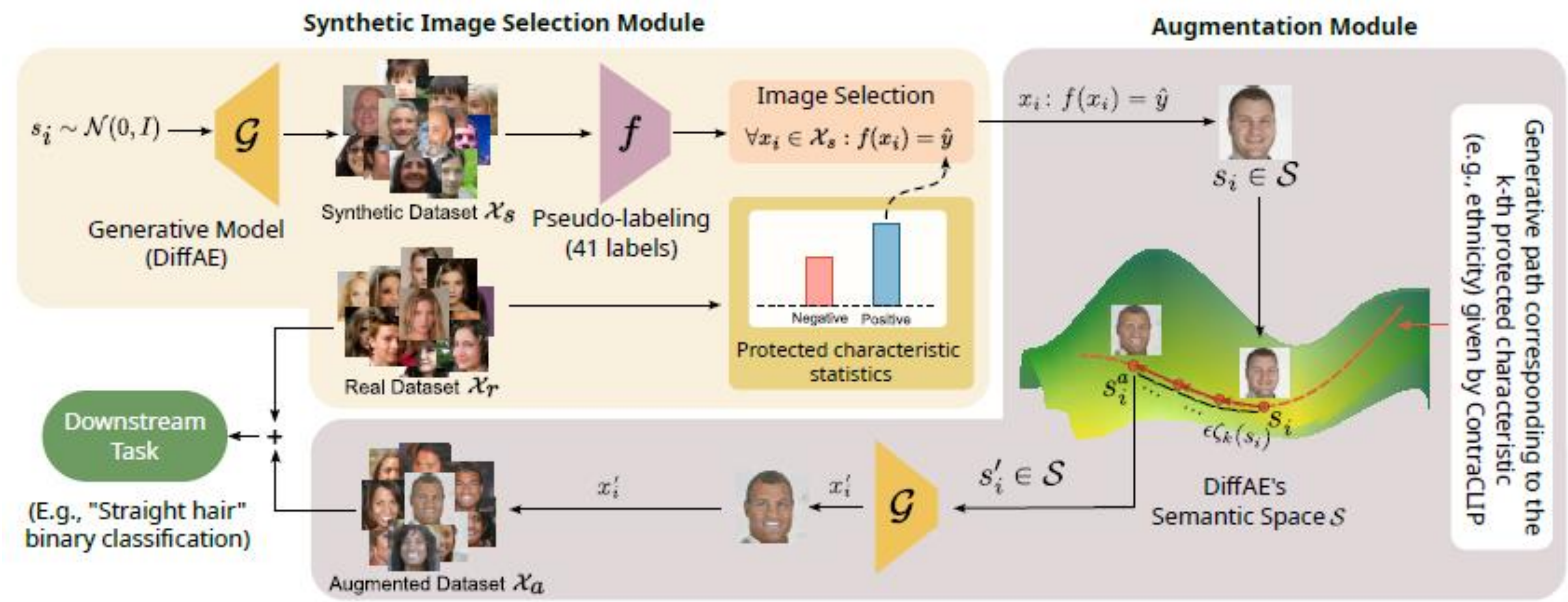
The generator bias may be overcome by:

- Augmenting the generated images towards the desired protected characteristic (e.g., old)

Pipeline:

- Compute statistics on the biased training set
- Identify the minority protected characteristic (e.g., dark skin tone)
- Augment generated images towards the desired protected characteristic
- The classifier is made fairer by fine-tuning on original and augmented synthetic data

Overcome the Generator Bias



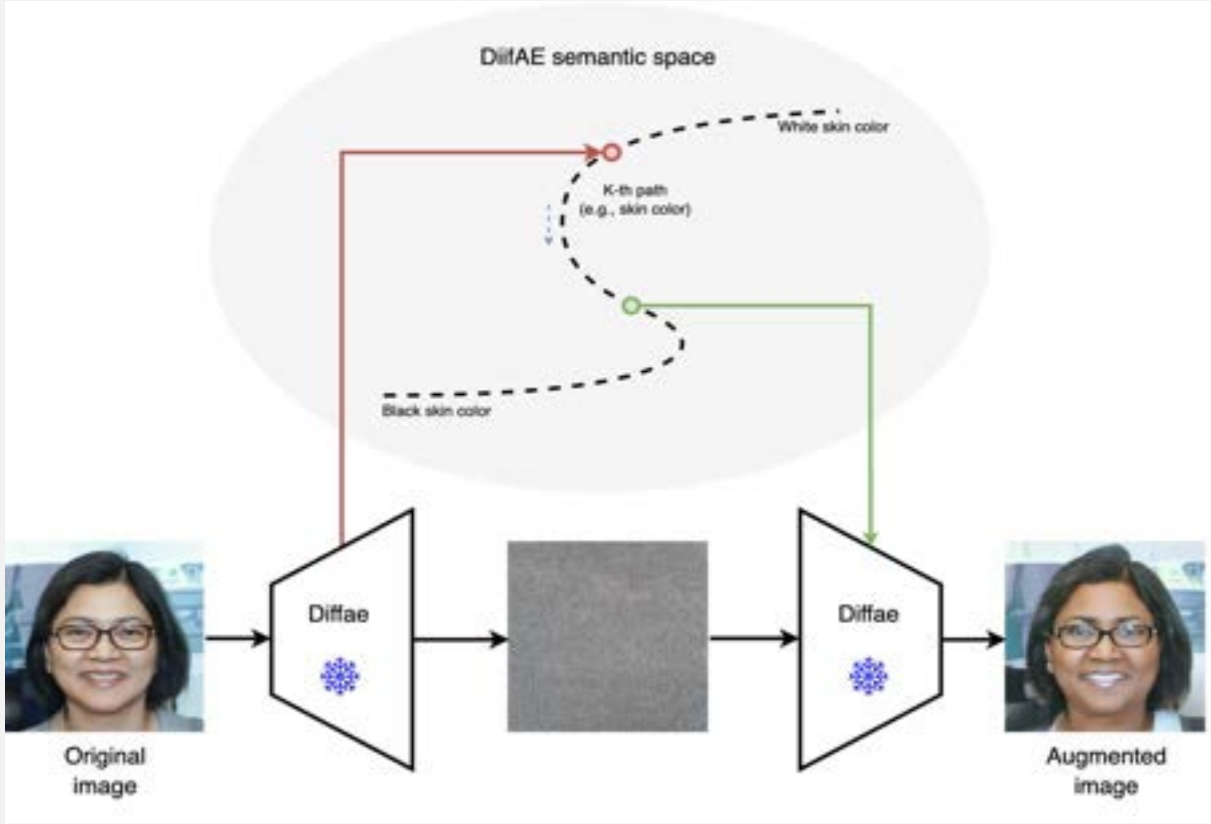
Augmentation Module

Find paths lying in the semantic space

- By leveraging natural language

Paths characteristics:

- Describe one protected characteristic
- When traversed convey the desired augmentation
- Edit only the specific facial attribute
    - Path disentanglement



[9] K. Preechakul, et al. Diffusion autoencoders: Toward a meaningful and decodable representation, CVPR22
[10] C. Tzelepis, et al., ContraCLIP: Interpretable GAN generation driven by pairs of contrasting sentences, 2022

Qualitative Results

Young Age Old

White Skin Color Black

Discussion

Assumptions and limitations:

- The learnt latent paths convey the desired manipulation while preserving the downstream attribute (disentanglement)
  - We attempt to impose the orthogonality of the paths by employing a contrastive loss which improves their disentanglement
- A good pseudo-labelling module is employed
  - Accuracy remains stable across different settings, suggesting the method is robust even when using a simple pseudo-labelling module
- Our method requires a generator with an editable space, pre-trained on data where the attributes to be manipulated are well-represented

Bias Detection via Foundation Models

Foundation models are becoming increasingly popular:

- Trained on high volume data
  - Capable of SOTA performance on multiple tasks
- They cover natural language (e.g., ChatGPT) and multimodal (e.g., LLaVA) domains

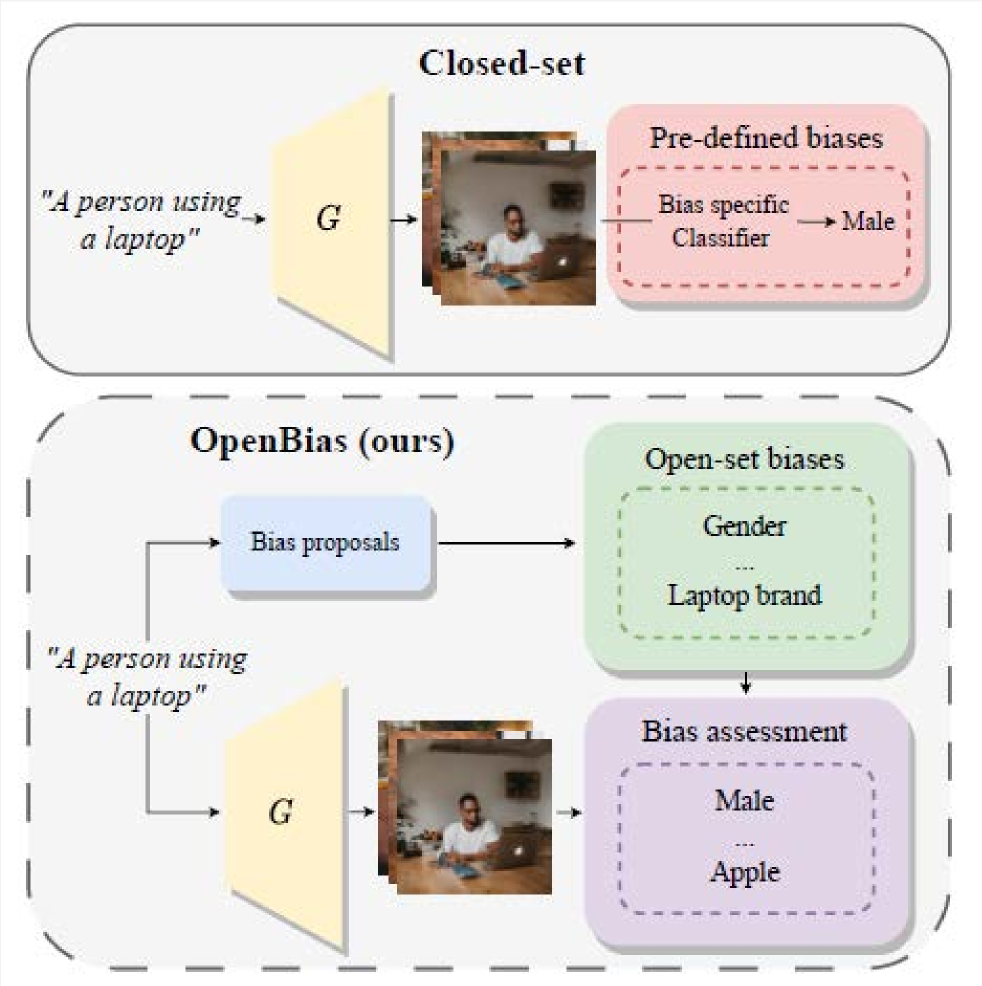
Bias detection in text-to-Image is still an open question:

- So far, we focused on **closed-set of biases**
- The models may exhibit novel biases previously uncovered

Can we use foundation models to **propose** and **detect** biases?

Bias Detection via Foundation Models

- OpenBias: discovering biases of T2I generative models in an open-set setting
- We do not require a predefined list of biases but propose a set of novel domain-specific biases



M. D'Inca, et al. OpenBias: Open-set Bias Detection in Generative Models, CVPR24

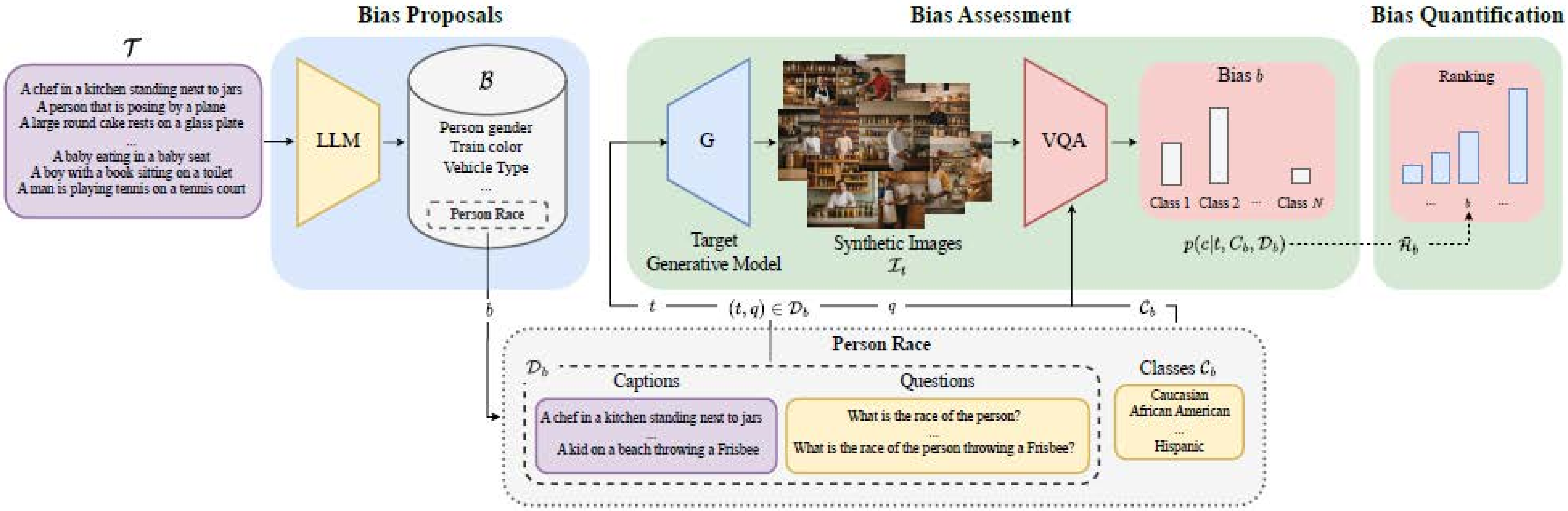
Key Ideas

Three stage pipeline:

Given a set of captions

- **Propose** biases via in-context learning on a Large Language Model (LLM)
- **Generate** the synthetic images with the target generative model G and the given captions
- **Check** the proposed biases via Vision Question Answering (VQA) on the synthetic dataset

OpenBias



Bias Proposals
Bias Assessment
Bias Quantification
$\mathcal{T}$
A chef in a kitchen standing next to jars
A person that is posing by a plane
A large round cake rests on a glass plate
...
A baby eating in a baby seat
A boy with a book sitting on a toilet
A man is playing tennis on a tennis court
LLM
$\mathcal{B}$
Person gender
Train color
Vehicle Type
...
Person Race
G
Target Generative Model
Synthetic Images $\mathcal{I}_t$
VQA
Bias b
Class 1
Class 2
Class N
$p(c|t, \mathcal{C}_b, \mathcal{D}_b)$
Ranking
$\mathcal{R}_b$
b
t
$(t, q) \in \mathcal{D}_b$
q
$\mathcal{C}_b$
Person Race
$\mathcal{D}_b$
Captions
A chef in a kitchen standing next to jars
...
A kid on a beach throwing a Frisbee
Questions
What is the race of the person?
...
What is the race of the person throwing a Frisbee?
Classes $\mathcal{C}_b$
Caucasian
African American
...
Hispanic

Results

Novel discovered biases:

- Person-related biases
- Object-related biases
- Animal-related biases

Train color — "A train zips down the railway in the sun"

Laptop brand — "A photo of a person on a laptop in a coffee shop"

Horse breed — "A cop riding a horse through a city neighborhood"