JUMPS: Joints Upsampling Method for Pose Sequences

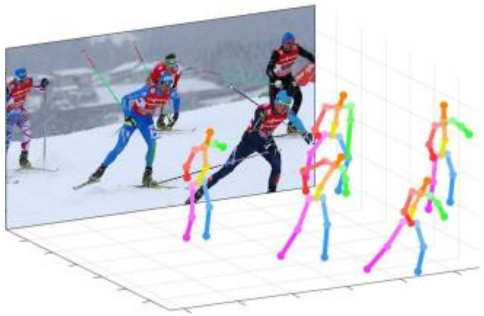
Lucas Mourot, François Le Clerc, Cédric Thébault and Pierre Hellier

Human Pose Estimation

- **Important in many applications s.t.**
  - Markerless motion capture
  - Advanced sports analysis
  - Autonomous driving

[1]

- **Frequent issues in 2D HPE**
  - Frame by frame ⟶ temporal inconsistencies
  - Limited number of joints s.t. 12 or 16
  - (Self-)Occlusions leading to missing keypoints



[2]

- **3D HPE**
  - Generally rely on 2D HPE in a way or another

[3]

Improving 2D Pose Sequences

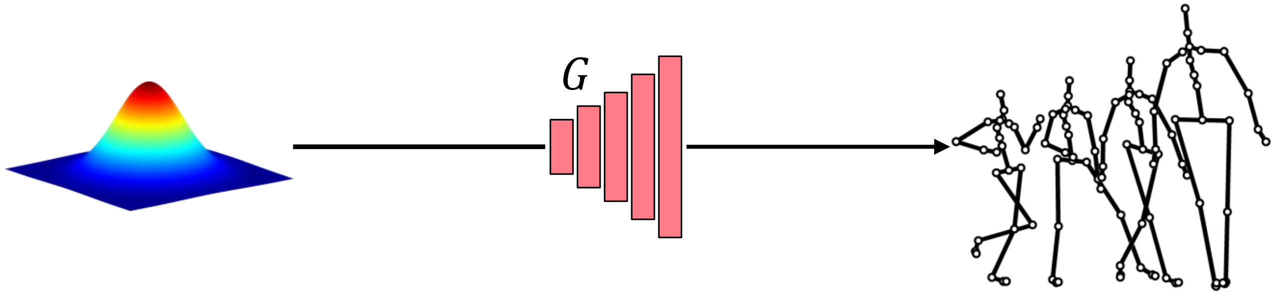
- **Goals:**
  - Increase spatial resolution, i.e. upsample joints
  - Recover occluded joints
  - Increase spatio-temporal consistency

- **Approach**
  - Learn the distribution of human motion at high spatial resolution with a deep generative model
  - Produce a complete high-res. motion with the model learnt through an optimization with the incomplete low-res. input motion as a constraint

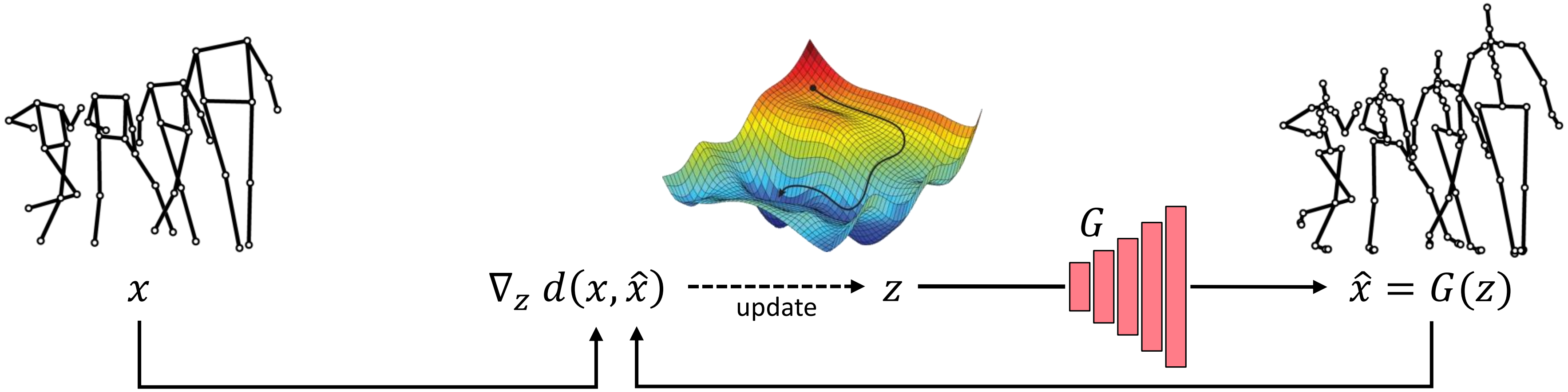
- **Benefits**
  - Enrich motions: details, completeness, consistency
  - Should help to disentangle 3D poses from 2D keypoints in 3D HPE

Method Overview

1. **Learn the distribution of human motions as a mapping from latent variables to pose sequences**
   - The mapping is parametrized by the weights of the generator G.

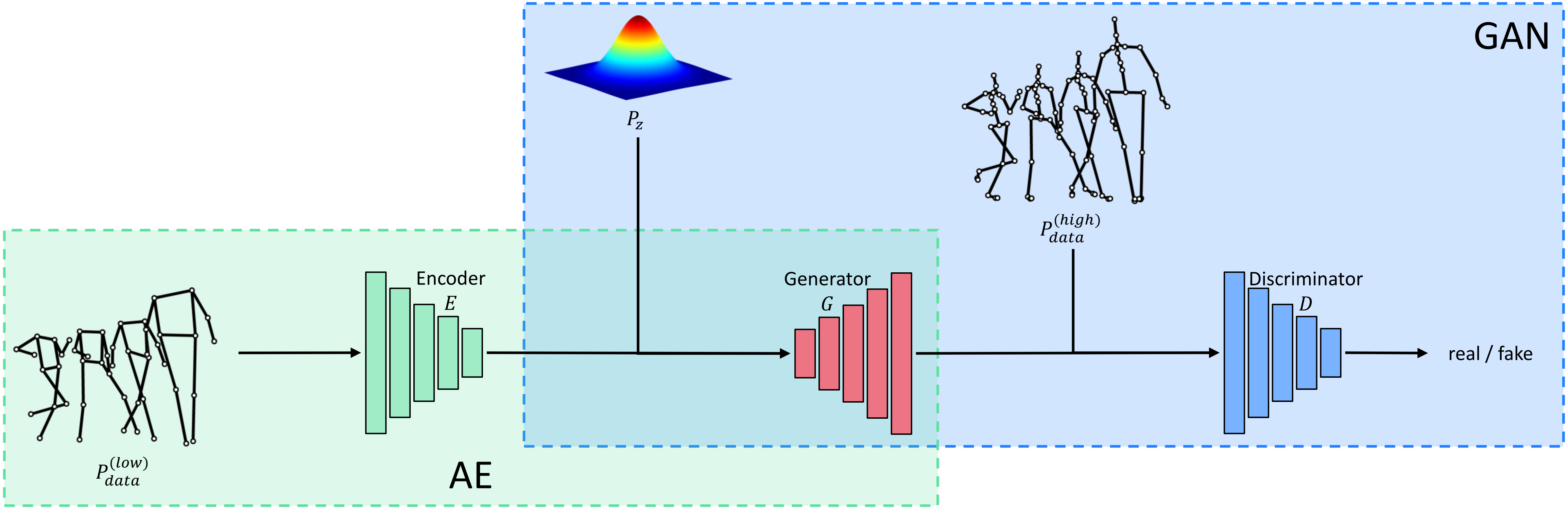


2. **Upsample and complete motions by optimizing latent variables s.t. generated motion matches input**
   - Distance computed over nonmissing joints.



Deep Generative Model

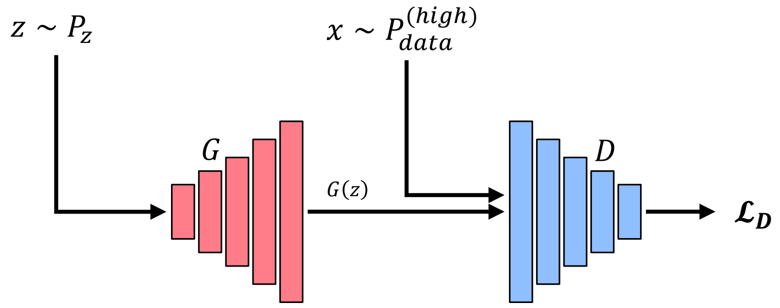
- **Autoencoder (AE) + Generative Adversarial Network (GAN)**
- **Decoder ≡ Generator**



Losses & Training

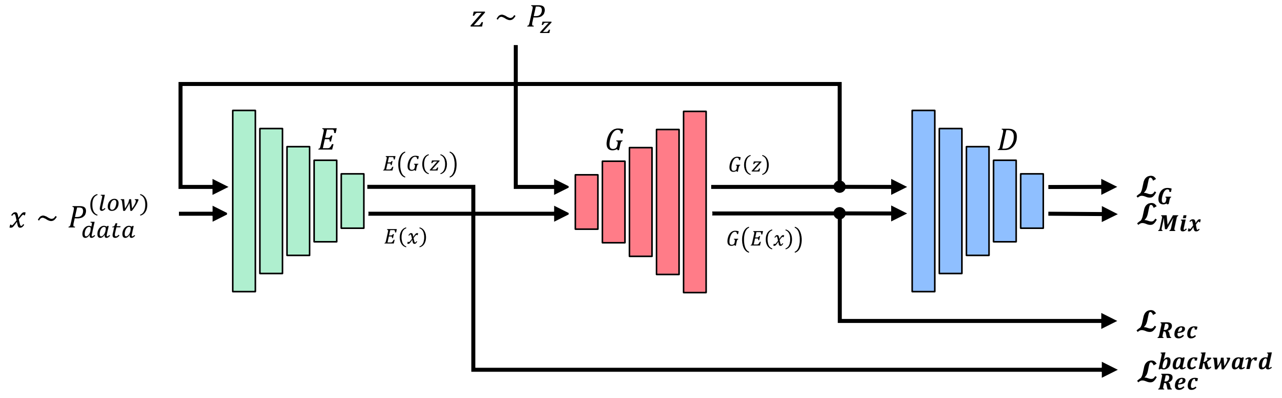
- **Alternating iterations between discriminator and generator / encoder.**
- **GAN framework:**
  - Wassertstein loss used (see $\mathcal{L}_D$ and $\mathcal{L}_G$)
- **AE framework:**
  - both positions and velocities are optimized (see $\mathcal{L}_{Rec}$)
  - Cyclic loss $\mathcal{L}_{Rec}^{backward}$ encourage the latent space learned to match P_z

Discriminator's training iteration



$$\mathcal{L}_D = D(G(z)) - D(x) + \underbrace{\lambda_{gp} \cdot (\|\nabla_{\tilde{x}} D(\tilde{x})\|_2 - 1)^2}_{Wasserstein\ gradient\ penalty}$$

Encoder and generator's training iteration



$$\mathcal{L}_G = -D(G(z))$$

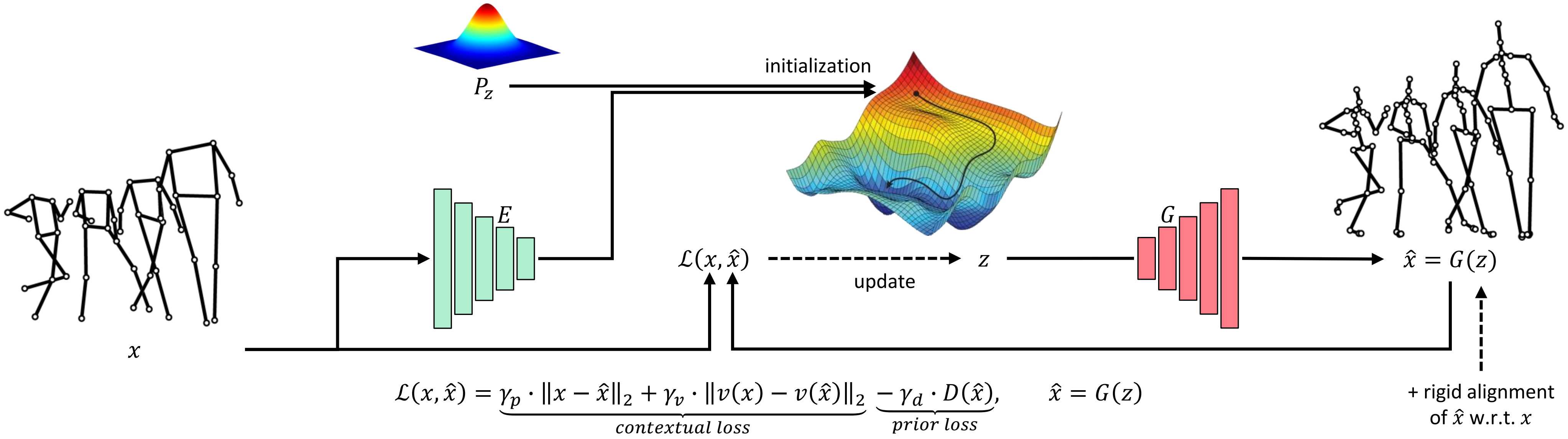
$$\mathcal{L}_{Mix} = -D\left(G(E(x))\right)$$

$$\mathcal{L}_{Rec} = \lambda_p \cdot \|x - \hat{x}\|_2 + \lambda_v \cdot \|v(x) - v(\hat{x})\|_2, \qquad \hat{x} = G(E(x))$$

$$\mathcal{L}_{Rec}^{backward} = \lambda_z \cdot MSE(z, \hat{z}), \qquad \hat{z} = E(G(z))$$

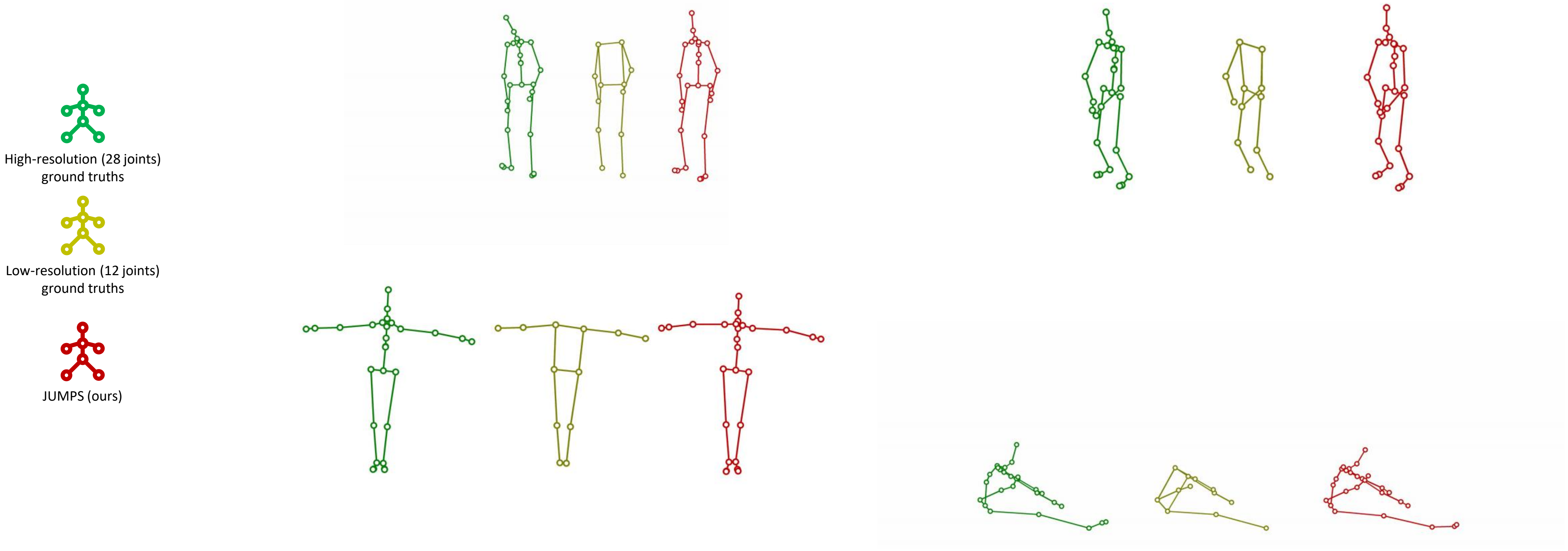
Latent Optimization

- Goal $x^* = G(z^*)$, with $z^* = \arg\min_{z \in P_z} \mathcal{L}(x, G(z))$
- $\mathcal{L} = contextual\ loss + prior\ loss$
- Several optimization in parallel starting from different z_0, including $E(x)$
- Rigid alignment of $G(z)$ w.r.t. input motion x



Results: Upsampling

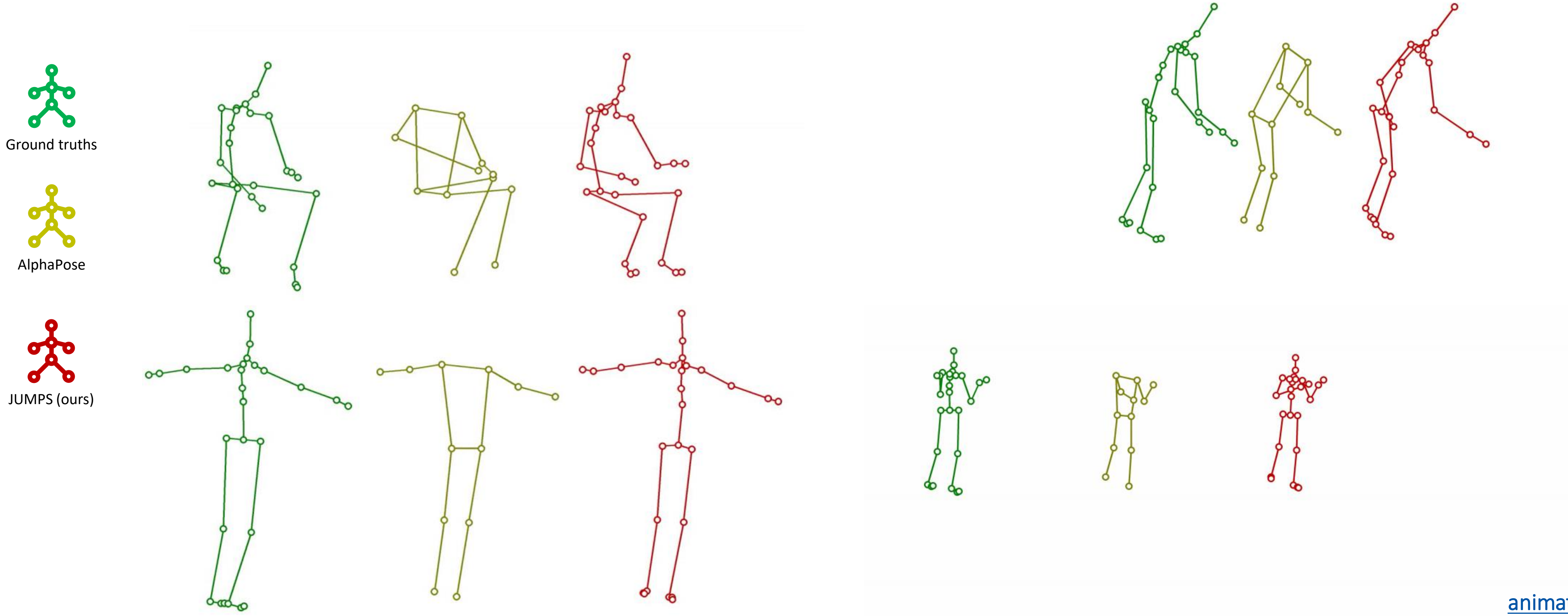
Method	PCKh@0.1	PCKh@0.5	PCKh@1.0	AUC over [0, 1]
JUMPS w/o alignment	0.0368	0.4384	0.6814	0.3912
JUMPS w/o encoder	0.1701	0.8259	0.9678	0.7005
JUMPS (ours)	**0.6096**	**0.9674**	**0.9965**	**0.8803**



animated results

Results: Human Pose Estimation Post-Processing

Method	PCKh@0.1	PCKh@0.5	PCKh@1.0	AUC over [0, 1]
JUMPS w/o alignment	0.0207	0.3423	0.6304	0.3249
JUMPS w/o encoder	0.0537	0.6801	0.9059	0.5692
AlphaPose	**0.0941**	0.7659	0.9157	0.6310
JUMPS (ours)	0.0842	**0.7723**	**0.9276**	**0.6341**



animated results