Supplementary of "GFPose: Learning 3D Human Pose Prior with Gradient Fields"

1. Details of subVP SDE

In this work, we use the subVP SDE proposed in [5] to perturb the 3D pose data. Formally,

$$d\mathbf{x} = -\frac{1}{2}\beta(t)\mathbf{x}dt + \sqrt{\beta(t)(1 - e^{-2\int_0^t \beta(s)\,ds})}d\mathbf{w}, \quad (1)$$

where $\mathbf{x} \in \mathbb{R}^{J\times 3}$ denotes the 3D human pose, $\beta(t)$ denotes the noise scale at timestep t. The drift coefficient of $\mathbf{x}(t)$ is $\mathbf{f}(\mathbf{x}, t) = -\frac{1}{2}\beta(t)\mathbf{x}$. The diffusion coefficient is $g(t) = \sqrt{\beta(t)(1 - e^{-2\int_0^t \beta(s)\,ds})}$. $t \in [0, 1]$ is a continuous variable. We adopt the linear scheduled noise scales:

$$\beta(t) = \beta(0) + t\left(\beta(1) - \beta(0)\right). \quad (2)$$

We empirically set the minimum and maximum noise scale $\beta(0)$ and $\beta(1)$ to 0.1 and 20.0, respectively.

2. Closed Form of Loss Function

Because the drift coefficient $\mathbf{f}(\mathbf{x}, t)$ is affine, according to [5], the transition kernel $p_{0t}(\mathbf{x}(t) \mid \mathbf{x}(0), \mathbf{c})$ in the loss function (Eq. 6 in the main text) is always a Gaussian distribution, where the mean and variance can be obtained in the closed form:

$$\mathcal{N}\left(\mathbf{x}(t); \mathbf{x}(0)e^{-\frac{1}{2}\int_0^t \beta(s)\,ds}, \left[1 - e^{-\int_0^t \beta(s)\,ds}\right]^2 \mathbf{I}\right). \quad (3)$$

Following [4], we choose the weighting function $\lambda(t) = \sigma(t)^2$. Thus, the loss function can be written as:

$$\begin{aligned}\mathcal{L} &= \mathbb{E}_{\mathcal{U}(t;0,1)}\left[\lambda(t)\left\|\mathbf{s}_\theta\left(\mathbf{x}(t), t, \mathbf{c}\right) + \frac{\mathbf{x}(t) - \mu}{\sigma^2}\right\|_2^2\right] \\ &= \mathbb{E}_{\mathcal{U}(t;0,1)}\left[\left\|\sigma(t)\mathbf{s}_\theta\left(\mathbf{x}(t), t, \mathbf{c}\right) + \mathbf{z}\right\|_2^2\right],\end{aligned} \quad (4)$$

where $\mathbf{z} \sim \mathcal{N}(0, 1)$ is random noise. $\mathbf{s}_\theta\left(\mathbf{x}(t), t, \mathbf{c}\right)$ is the score network we would like to learn. According to Eq. 3, we can get $\sigma(t) = 1 - e^{-\int_0^t \beta(s)\,ds}$.

During training, we uniformly sample the time variable t from $[0, 1]$ and sample a noise vector $\mathbf{z} \in \mathbb{R}^{J\times 3}$ from a standard normal distribution. Then we perturb the ground-truth 3D human pose $\mathbf{x}(0)$ according to Eq. 3 to get the noisy 3D pose $\mathbf{x}(t)$:

$$\mathbf{x}(t) = \mathbf{x}(0)e^{-\frac{1}{2}\int_0^t \beta(s)\,ds} + \mathbf{z} \cdot \left(1 - e^{-\int_0^t \beta(s)\,ds}\right). \quad (5)$$

Then we can compute the loss according to Eq. 4 to train the score network.

3. Network Architecture

Fig. 1 shows the architecture of our score network $\mathbf{s}_\theta$. It is a simple fully-connected network with a structure similar to [3]. Following [3], we set the hidden dimension of all FC layers to 1024. We adopt group normalization [6] with the number of groups set to 32. We use SiLU [1] as nonlinear activation and set the dropout rate to 0.25.

4. Pose Sampling

To sample human poses from the learned pose prior $p_{data}(\mathbf{x}|\mathbf{c})$, we need to solve the reverse-time SDE (Eq. 5 in the main text). Following [5], we simulate the RSDE via the Predictor-Corrector (PC) sampler. We use the Euler-Maruyama solver as the predictor and Identity as the corrector. We set the number of sampling steps $N = 1000$, start time $T = 1.0$ and the end time $eps = 1e-3$. Please refer to Alg. 1 for the detailed sampling process.

5. Detailed Settings for Different Tasks

In this section, we detail how we train and use our model for each task described in the main text. The difference lies in the types of conditions, masking strategy and sampling process.

Monocular 3D Human Pose Estimation During training, GFPose conditions on the detected 2D pose without any condition masking strategy, *i.e.*, HPJ-000. At inference, we sample a noise vector $\mathbf{x}(T) \in \mathbb{R}^{J\times 3}$ from a standard normal distribution $\mathcal{N}(0, 1)$, and gradually adjust it according to the standard sampling process described in Alg. 1.

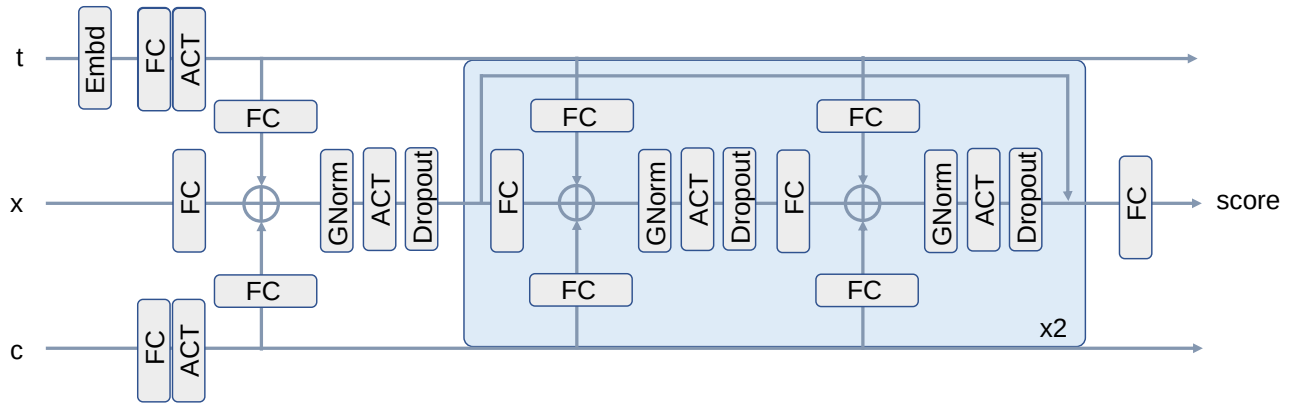

Figure 1. Architecture of the score network $\mathbf{s}_\theta$. It is a plain fully connected network consisting of 2 residual blocks. $\mathbf{x} \in \mathbb{R}^{J\times 3}$ denotes noisy 3D poses. $\mathbf{c}$ denotes different task conditions, *e.g.*detected 2D poses. t denotes the timestep. $\oplus$ denotes the sum operator.

Algorithm 1 Sampling from $p_{data}(\mathbf{x}|\mathbf{c})$

Require: learned $\mathbf{s}_\theta$; sampling step N; condition c; start time T; end time eps

$\mathbf{f}(\mathbf{x}, t) \leftarrow -\frac{1}{2}\beta(t)\mathbf{x}$
$g(t) \leftarrow \sqrt{\beta(t)(1 - e^{-2\int_0^t \beta(s)\,ds})}$
$dt \leftarrow \frac{1}{N}$
$\mathbf{x}_N \sim \mathcal{N}(0, 1)$
for $i \leftarrow N-1$ to 0 **do**
  $t \leftarrow eps + (T - eps) \cdot \frac{i+1}{N}$
  $\mathbf{x}'_i \leftarrow \mathbf{x}_{i+1} - \left[\mathbf{f}(\mathbf{x}_{i+1}, t) - g(t)^2 \mathbf{s}_\theta\left(\mathbf{x}_{i+1}, t, \mathbf{c}\right)\right] dt$
  $\mathbf{z} \sim \mathcal{N}(0, 1)$
  $\mathbf{x}_i \leftarrow \mathbf{x}'_i + g(t)\sqrt{dt}\mathbf{z}$
end for
return $\mathbf{x}_0$

Pose Completion (2D $\rightarrow$ 3D) During training, GFPose conditions on the detected 2D pose with masking strategy HPJ-001 and HPJ-020 for missing joint and part completion, respectively. Here, we explore three sampling approaches. The first is the standard conditional inference process as described in Alg. 1. The second is the imputation approach proposed in [5]. We also try combining these two inference techniques as the third approach. In this task, we adopt the combination approach as it performs best. In Sec. 7, we will further compare different inference approaches.

Pose Completion (3D $\rightarrow$ 3D) During training, GFPose conditions on the gt 3D pose with a masking strategy HPJ-020 for missing body part completion. We also adopt the combination approach during sampling.

Denoising Mocap Data In this task, GFPose is trained with a masking strategy HPJ-T00, *i.e.*, human level mask is always activated and the condition is always zero. To use this model for denoising, we condition the model on $\emptyset$ and start the sampling process from a noisy 3D pose $\mathbf{x}(T) \in \mathbb{R}^{J\times 3}$ instead of a noise vector. We set the start time T of reverse-time SDE to a small value of 0.05 or 0.1 instead of the default value 1.0, as shown in Table 7 in the main text. Intuitively, a smaller start time notifies the score network $\mathbf{s}_\theta$ that we are not starting from pure noise, so a small adjustment is sufficient.

Pose Generation For generation, GFPose is trained with a masking strategy HPJ-T00. During inference, GFPose conditions on $\emptyset$ and gradually adjusts the noise vector $\mathbf{x}(T) \in \mathbb{R}^{J\times 3}$ sampled from a standard normal distribution $\mathcal{N}(0, 1)$ to generate realistic and diverse 3D poses.

6. Train A Unified Model for Various Tasks

Instead of training separate models for different tasks, we can also train one unified model for all tasks. Concretely, we active all three levels of masks during training. There are two different choices to train such a unified model. 1) We only condition on the detected 2D pose, *i.e.* $\mathbf{c} \subset \{\mathbf{x}_{2d}\}$. We call this model "U2D". 2) We condition on the detected 2D pose 0.9 and the 3D pose with a probability of 0.1, *i.e.* $\mathbf{c} \subset \{\mathbf{x}_{2d}, \mathbf{x}_{3d}\}$. We call this model "U3D". Note that the 3D condition shares the same masking strategy as the 2D condition. This helps the model to explicitly establish the relationship between different 3D body parts. We quantitatively study the unified model in the following.

Monocular 3D Pose Estimation We first ablate condition masking strategies on monocular 3D human pose estimation. Table 1 reports the minMPJPE(mm) for different models on H3.6M dataset. We can find that training with human- or part- level masks slightly boosts the per-

p_h	p_p	p_j	2D	3D	minMPJPE ($S = 1/200$)
0.0	0.0	0.0	✓	✗	51.0 / 35.6
0.1	0.0	0.0	✓	✗	50.8 /35.8
0.0	0.1	0.0	✓	✗	**50.5 / 35.1**
0.0	0.0	0.1	✓	✗	51.7 / 36.4
0.1	0.1	0.0	✓	✗	51.0 / 36.0
0.0	0.1	0.1	✓	✗	51.1 / 35.5
0.1	0.1	0.1	✓	✗	52.3 / 36.7
0.1	0.2	0.1	✓	✗	52.4 / 36.5
0.1	0.1	0.1	✓	✓	**51.3 / 35.9**
0.1	0.2	0.1	✓	✓	51.6 / 36.4

Table 1. Effects of condition masking strategies on monocular 3D human pose estimation. We report minMPJPE(mm) on H3.6M dataset under protocol #1. S denotes the number of samples. p_h, p_p and p_j respectively denote the probability of activating human-, part- and joint- level masks.

formance of pose estimation ($\sim$0.5mm). Training with the mixed masking strategy, *i.e*. the unified model "U2D" (3rd and 4th row from last) causes a slight drop ($\sim$1mm) in reconstruction accuracy. Further incorporating 3D pose (1st and 2nd row from last, "U3D") can benefit the pose estimation task.

Pose Completion We compare the two unified models "U2D" and "U3D" on the pose completion task. From Table 2, we can observe that the unified models perform slightly worse than the specifically trained model when recovering occluded body parts. However, they achieve competitive results as the specifically trained model when recovering occluded random joints and significantly outperforms the SOTA [2]. From Table 3, we can find that the unified model "U3D" performs quite well when recovering from partial 3D observations. Note that "U2D" cannot directly apply to this task unless through the imputation technique introduced in the next section.

Pose Denoising Table 4 shows that the unified model performs slightly better on small noise intensities while the specifically trained model (HPJ-T00) performs slightly better on large noise intensities.

Pose Generation We train 3 deterministic pose estimators on the synthetic poses generated by HPJ-T00, "U2D" and "U3D", respectively. Table 5 shows that the specifically trained model HPJ-T00 can generate the best quality poses. Unified models also show competitive results.

Summary We can trade very little performance loss for a versatile unified model "U3D".

Occ. Parts	Sep.	U2D	U3D	Li *et al*. [2]
1 Joint	37.8	**37.5**	37.6	58.8
2 Joints	39.6	39.8	**39.5**	64.6
2 Legs	**53.5**	54.9	53.8	-
2 Arms	**60.0**	62.1	60.4	-
Left Leg + Left arm	**54.6**	56.4	55.2	-
Right leg + Right arm	**53.1**	54.3	53.5	-

Table 2. Recover 3D pose from partial 2D observation. We compare the unified models ("U2D" and "U3D") and the model specifically trained for this task (Sep. here means HPJ-001 and HPJ-020, reported in the main text. Please refer to Section 5). We report minMPJPE(mm) on H3.6M dataset under protocol #1. 200 samples are drawn.

Occ. Parts	Sep.	U3D
Right Leg	**5.2**	5.6
Left Leg	**5.8**	5.9
Left Arm	**9.4**	9.7
Right Arm	**8.9**	9.0

Table 3. Recover 3D pose from partial 3D observation. We compare the unified models ("U2D" and "U3D") and the model specifically trained for this task (Sep., reported in the main text. Please refer to Section 5 for details) on H3.6M dataset and report minMPJPE(mm) under protocol #1. 200 samples are drawn.

Noisy Data	Base	Sep.	U2D	U3D	Start T
GT	0	14.7	**13.8**	14.0	0.05
GT + $\mathcal{N}(0, 25)$	33.1	25.0	25.1	**24.8**	0.05
GT + $\mathcal{N}(0, 100)$	65.5	**42.8**	44.0	43.5	0.05
GT + $\mathcal{N}(0, 400)$	126.0	**64.6**	65.5	65.2	0.1
GT + $\mathcal{U}(25)$	49.1	**32.8**	33.5	33.0	0.05
GT + $\mathcal{U}(50)$	96.2	**50.9**	51.0	51.0	0.1
GT + $\mathcal{U}(100)$	178.2	**89.4**	91.4	91.1	0.1

Table 4. Denoising results on H3.6M dataset. We report MPJPE (mm) under Protocol #2. "Sep." here indicates model "HPJ-T00". (Please refer to Section 5) "Base" represents the MPJPE(mm) of noisy data. $\mathcal{N}$ and $\mathcal{U}$ denote Gaussian and uniform noise, respectively. T denotes the start time of RSDE.

Model	Sep.	U2D	U3D
MPJPE	**58.1**	60.8	59.7

Table 5. Quality comparison of generated poses. We train 3 deterministic pose estimators on the poses generated by 3 different GF-Pose models (Sep. here means "HPJ-T00". Please refer to Section 5). Then we evaluate them on the H3.6M test set. MPJPE(mm) under Protocol #1 is reported.

Model	Cond.	Impt.	Cond.+Impt.
HPJ-010	**35.1**	49.5	36.8
HPJ-T00	-	**42.3**	-
U2D	**36.7**	43.0	37.2
U3D	**35.9**	43.0	37.1

Table 6. Comparison between different sampling approaches on 3D pose estimation. HPJ-T00 means we always activate the human level mask and learn an unconditional score model.

Model	Cond.	Impt.	Cond.+Impt.
HPJ-020	53.9	72.8	**53.5**
HPJ-T00	-	**64.9**	-
U2D	55.8	67.4	**54.9**
U3D	54.6	66.8	**53.8**

Table 7. Comparison between different sampling approaches on pose completion from partial 2D observation (2 legs). HPJ-T00 indicates an unconditional score model.

7. Different Sampling Approaches

Imputation [5] provides a flexible way to use score models for imputation tasks, like image inpainting and colorization. By repeatedly replacing part of the data in $\mathbf{x}(t)$ with the observed data $\mathbf{x}_{obs}$ during the sampling process, imputation allows us to use unconditional score models to impute the missing dimensions of data. Here, we can also view the 3D pose estimation and the completion task as imputation tasks, *i.e.*, we impute the depth of 2D poses or impute the missing body joints. We compare the performance of imputation, conditional inference and the combination of these 2 techniques in Table 6, 7, 8. We can get the following observations: (1) conditional inference generally performs better than imputation. (2) Combining two inference approaches can yield consistently better results than either approach alone on pose completion tasks. (3) Conditional inference performs best on the 3D human pose estimation task. (4) Imputation works better with unconditional models (HPJ-T00).

8. More Qualitative Results

We show more qualitative results for multi-hypothesis 3D pose estimation (Figure 2), pose completion (Figure 3), pose denoising and generation (Figure 4). All poses are sampled from the unified model "U3D". Please find the demo videos and qualitative comparisons with other methods from our project webpage: https://sites.google.com/view/gfpose/.

Model	Cond.	Impt.	Cond.+Impt.
HPJ-020	5.3	8.3	**5.2**
HPJ-T00	-	**6.0**	-
U2D	-	**6.2**	-
U3D	8.6	6.1	**5.6**

Table 8. Comparison between different sampling approaches on pose completion from partial 3D observation (rigth leg). HPJ-T00 indicates an unconditional score model.

References

[1] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 1

[2] Chen Li and Gim Hee Lee. Generating multiple hypotheses for 3d human pose estimation with mixture density network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3

[3] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *ICCV*, pages 2640–2649, 2017. 1

[4] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019. 1

[5] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 1, 2, 4

[6] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 1

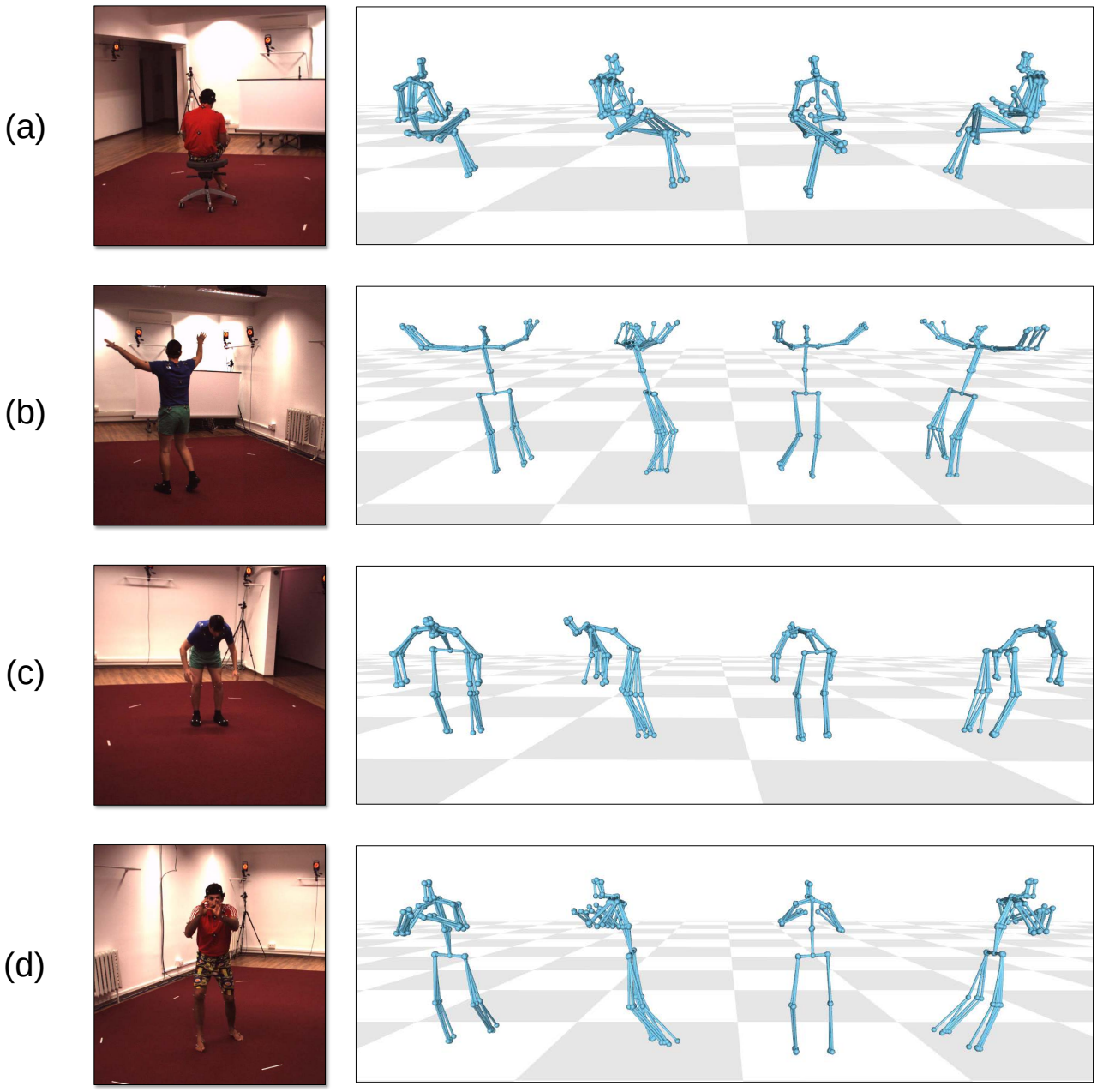

Figure 2. Multi-hypothesis 3D human pose estimation. We randomly sample 5 hypotheses from GFPose and show them from 4 different viewpoints. From left to right, the plausible poses are rotated 90 degrees clockwise around the z-axis.

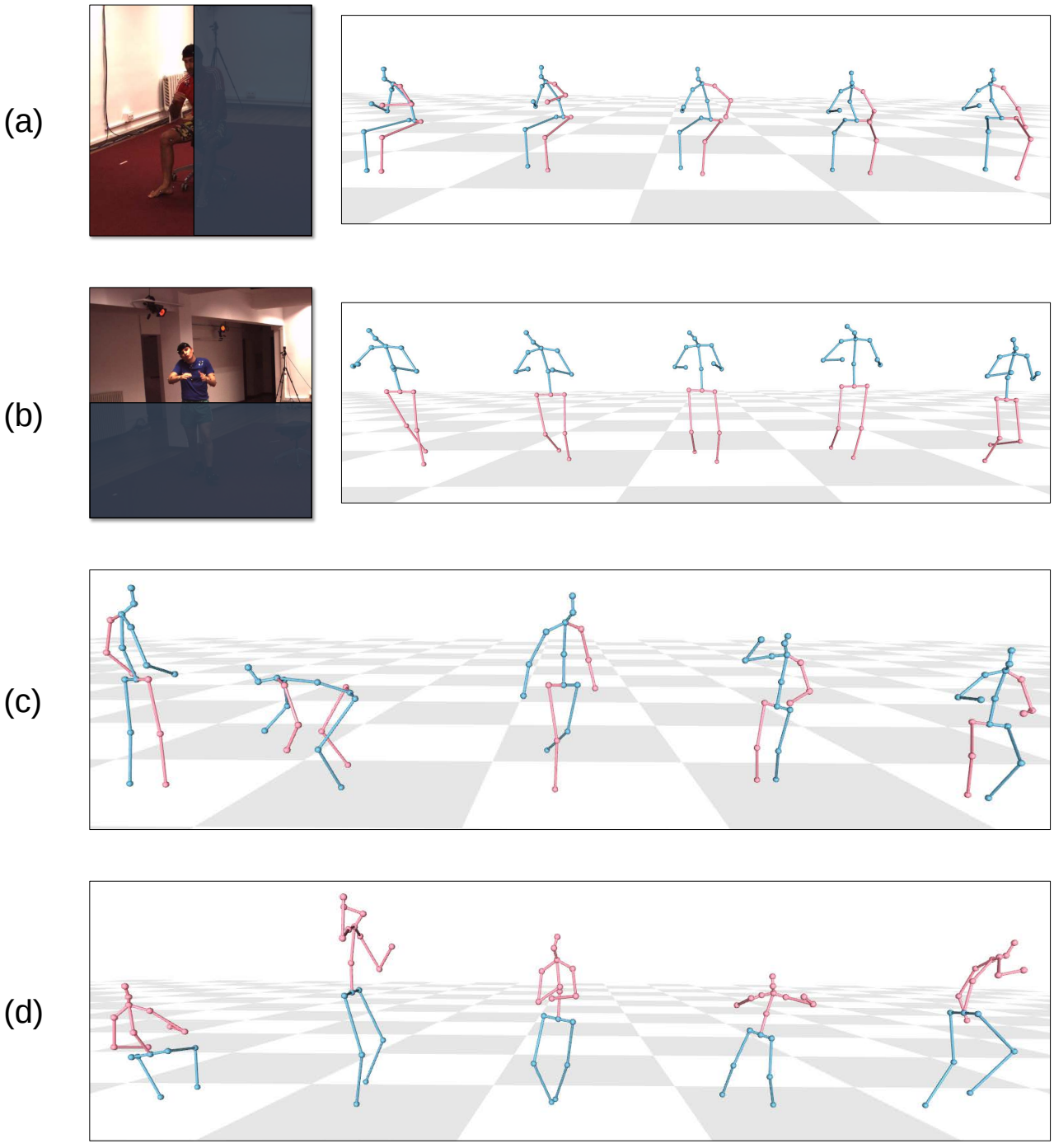

Figure 3. Pose completion. GFPose can recover full 3D human body from partial observations. 3D poses corresponding to the visible observation and missing observation are plotted in different colors for clarity. **(a)(b)** Recover full 3D human body from partial 2D image. We show five plausible poses sampled from GFPose. (a) Left side of the body is invisible. (b) Lower body is invisible. **(c)(d)** Recover full 3D human body from partial 3D poses. For each instance, we sample and plot one plausible completion. (c) Left arm and right leg are missing. They are completed by GFPose. (d) Upper body are missing and completed by GFPose.

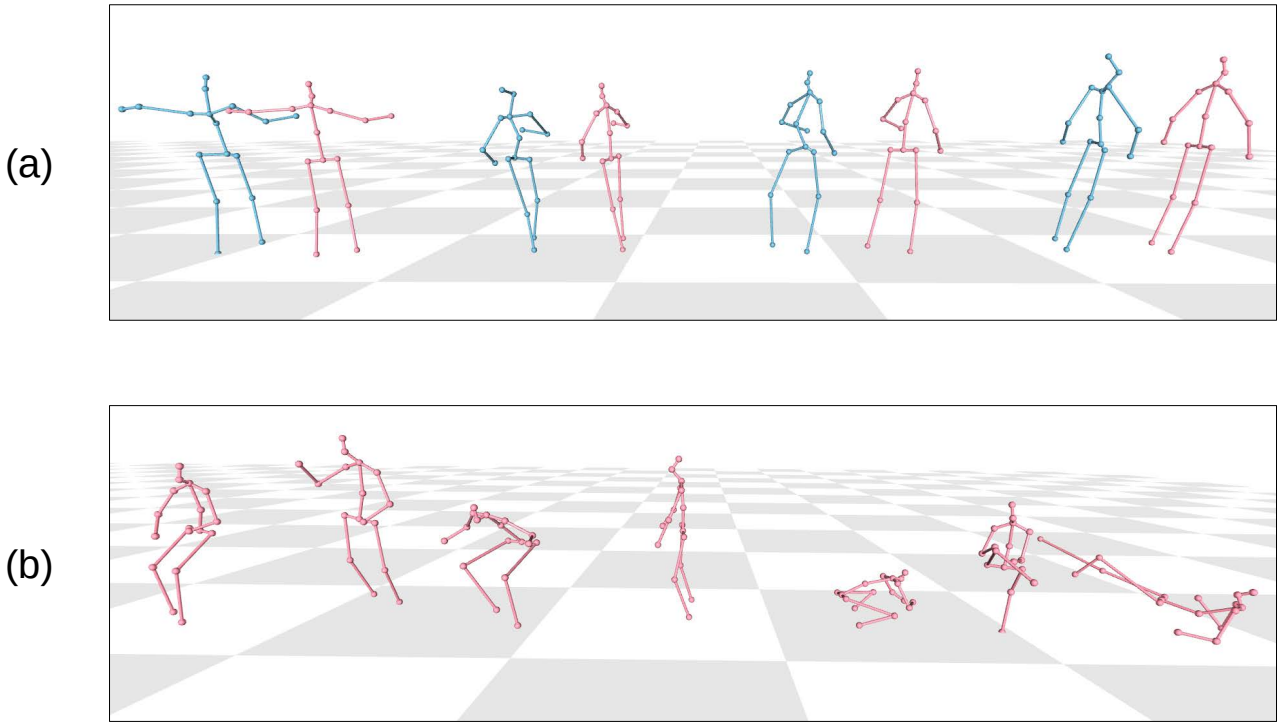

Figure 4. Pose denoising and generation. **(a)** Pose denoising. We add Gaussian noise $\sim \mathcal{N}(0, 100)$ onto GT and denoise it with GFPose. Noisy poses and denoised poses are plotted in different colors. GFPose can effectively correct unreasonable poses. **(b)** Pose generation. GFPose can generate diverse and realistic poses.