Stratified Avatar Generation from Sparse Observations

Supplementary Material

In this supplementary material, we provide additional ablation on our design choice of the SAGE Net and implementation specifics.

A. Extra Ablation Studies

A.1 Input sequence length

Our model adheres to the online inference setting, where it processes sparse tracking signals from the past N frames and predicts the full body motion of the final frame as done in [2, 8]. As indicated in [1, 2, 8], the length of the input sequence is a critical factor affecting the model's performance, involving a balance between efficiency and effectiveness. Therefore, it is essential for our model to effectively tackle shorter sequences, as this not only maintains performance but also significantly reduces computational costs.

We examine AvatarJLM [8] and our method with different input lengths N under setting S1, as presented in Tab. A. The results demonstrate that our proposed SAGE Net is more robust to variations in the input sequence length compared to the baseline method, AvatarJLM [8]. Notably, SAGE Net is able to exceed AvatarJLM's performance even when utilizing just a quarter of their sequence length (10 frames for our method compared to 40 frames for AvatarJLM).

A.2 Predicting noise

Our SAGE Net follows the approach of previous methods [5, 6] by directly predicting the raw data during the diffusion process, specifically the clean latent z_0 in our context. In this subsection, we adapt the diffusion process to predict the residual noise ϵ instead of z_0, while maintaining all other components as they are, to validate the effectiveness of this design choice. Results are detailed in Tab. B. We observe that compared with predicting the noise ϵ, this strategy leads to enhanced performance.

B. Implementation Details

B.1 Disentangled VQ-VAE

The VQ-VAE$_{up}$ and VQ-VAE$_{low}$ follow the architecture in [3], unitizing a 4-layer transformer network [7]. Each of these transformer layers includes a 4-head self-attention module and a feedforward layer with 256 hidden units.

For the training of VQ-VAEs, we employ a set of loss terms including a rotation-level reconstruction loss, a forward kinematic loss as proposed in [2], and a hand loss as proposed in [8] with batch size of 512. Adam optimizer is adapted for training, and we set its Betas parameters to $(0.9, 0.99)$ and the weight decay rate to $1e-4$. The initial learning rate is $1e-4$ and decreases by a factor of 0.2 at the milestone epochs $[25, 35, 50]$.

B.2 Stratified Diffusion

In our transformer-based model for upper-body and lower-body diffusion, we integrate an additional DiT block as described in [4]. Each model features 12 DiT blocks, each with 8 attention heads, and an input embedding dimension of 512. The full-body decoder is structured with 6 transformer layers.

The diffusion process is trained with 1000 sampling steps, employing the "squaredcos_cap_v2" beta schedule. For this schedule, we set the starting beta value at 0.00085 and the ending beta value at 0.012. The training of the upper-body diffusion model, lower-body diffusion model, and the full-body decoder D_{full}, is conducted sequentially. Each component is trained with a batch size of 400, using the Adam optimizer. We set the weight decay at 1e-4 and begin with an initial learning rate of 2e-4. The learning rate undergoes a reduction by a factor of 0.25 at the milestone epochs of 20 and 30.

B.3 Refiner

The refiner is a simple two-layer GRU for smoothing the output sequence with minimal computational expense. During the training stage, the refiner learns to predict the residual error $\hat{\Theta}_{res}$ between the ground truth motion Θ and the predicted motion $\hat{\Theta}$ from the full-body decoder. The final rotation prediction $\hat{\Theta}_{final}$ can be obtained by:

$$\hat{\Theta}_{final} = \hat{\Theta} + \hat{\Theta}_{res} \tag{1}$$

For achieving a balance between smoothness and accuracy in the predicted motion sequences, we adopt various loss terms previously utilized in related research [2, 8]. These include the rotation-level reconstruction loss L_{rec}, the velocity loss L_{vel}, and the forward kinematic loss L_{fk}.

In addition, we design a new loss term jitter loss L_{jitter} to directly control the jitter:

$$L_{jitter} = \frac{f^3}{N-3} \sum_{i=1}^{i=N-3} ||(\hat{v}_{i+2} - \hat{v}_{i+1}) - (\hat{v}_{i+1} - \hat{v}_i)||_2 \tag{2}$$

where $\hat{v}_i,\ i = 1, 2..., N-1$, represents the predicted joint velocity of i^{th} frames, and f represents the fps (frames per second).

Method	Length	MPJRE	MPJPE	MPJVE	Hand PE	Upper PE	Lower PE	Jitter
AvatarJLM	10	3.19	3.76	24.67	1.31	1.84	7.13	11.39
AvatarJLM	20	3.76	3.52	21.69	1.25	1.73	6.65	9.17
AvatarJLM	40	2.90	3.35	20.79	1.24	1.72	6.20	8.39
SAGE (Ours)	10	2.56	3.34	22.45	1.34	1.44	6.08	8.07
SAGE (Ours)	20	2.53	3.28	20.62	1.18	1.39	6.01	6.55
SAGE (Ours)	40	2.51	3.20	19.36	1.39	1.43	5.75	7.28

Table A. Ablation of the input sequence length. The purple background color denotes the motion length used in the original methods. The computational cost is directly proportional to the length of the input sequence, so we select 20 as our choice for the optimal trade-off between performance and computational cost.

Method	MPJRE	MPJPE	MPJVE	Hand PE	Upper PE	Lower PE	Root PE	Jitter
SAGE (pred noise)	3.64	4.43	25.18	3.79	2.41	7.38	3.64	9.00
SAGE (Ours)	2.53	3.28	20.62	1.18	1.39	6.01	2.95	6.55

Table B. Ablation of the diffusion formulation: Predicting original latent z *vs* predicting the residual noise ϵ. Predicting clean latent z achieves superior performance. The purple background color denotes our choice.

The complete loss term for training the refiner can be written as:

$$L = \alpha * L_{rec} + \beta * L_{vel} + \gamma * L_{fk} + \delta * L_{jitter}$$

We set $\alpha, \beta, \gamma, \delta$ to 0.01, 10, 0.05, and 0.01 to force the refiner to focus more on motion smoothness in the training process.

All experiments can be carried out on a single NVIDIA GeForce RTX 3090 GPU card, using the Pytorch framework.

References

[1] Yuming Du, Robin Kips, Albert Pumarola, Sebastian Starke, Ali K. Thabet, and Artsiom Sanakoyeu. Avatars grow legs: Generating smooth human motion from sparse tracking inputs with diffusion model. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 481–490, 2023. 1

[2] Jiaxi Jiang, Paul Streli, Huajian Qiu, Andreas Fender, Larissa Laich, Patrick Snape, and Christian Holz. Avatarposer: Articulated full-body pose tracking from sparse motion sensing. In *European Conference on Computer Vision (ECCV)*, pages 443–460, 2022. 1

[3] Thomas Lucas, Fabien Baradel, Philippe Weinzaepfel, and Grégory Rogez. Posegpt: Quantization-based 3d human motion generation and forecasting. In *European Conference on Computer Vision (ECCV)*, pages 417–435, 2022. 1

[4] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4172–4182, 2023. 1

[5] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *CoRR*, 2022. 1

[6] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit Haim Bermano. Human motion diffusion model. In *International Conference on Learning Representations (ICLR)*, 2023. 1

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 5998–6008, 2017. 1

[8] Xiaozheng Zheng, Zhuo Su, Chao Wen, Zhou Xue, and Xiaojie Jin. Realistic full-body tracking from sparse observations via joint-level modeling. In *IEEE/CVF international conference on computer vision (ICCV)*, 2023. 1