

Towards Generalizable 3D Human Pose Estimation via Ensembles on Flat Loss Landscapes - Appendix

A Implementation Details

There are several possible strategies for predicting the scale value (e.g., modifying the final layer or using a separate network). We selected the prediction method based on the characteristics of each network architecture. For MLPs, CNNs, and GCNs, we increased the output dimension of the final layer by one, enabling the network to estimate the scale value jointly with the 3D poses. For Transformer-based architectures, we instead used a separate lightweight MLP consisting of one residual block and a hidden dimension of 1024 to predict the scale. This design choice was motivated by the fact that the attention mechanism in Transformers inherently captures global information, which could cause gradient interference if scale prediction were integrated into the main network.

All our models, except those reported in Table 3 and Table 4 of the main paper, were trained on the H36M dataset [2] for 50 epochs with a batch size of 1024. To stabilize the training procedure and avoid overfitting due to the fast convergence with our adaptive scaling mechanism, a linear decay schedule was applied every two epochs throughout training. The models in Table 3 and Table 4 of the main paper were trained for 60 epochs on the same dataset due to the slow convergence of SAM [1], also with a batch size of 1024 and linear decay. We used the Adam optimizer [4] for all experiments. The initial learning rate was set to 0.02 for GCN model and 0.001 for all other models. A dropout rate of 0.25 was used for all our models except GCN and Transformer models. For GCN and Transformer models, we didn't apply dropout. All models were trained using Mean Squared Error (MSE) loss. The number of regression heads M was set to 3. To stabilize training in multi-head architectures, we applied max-norm gradient clipping with a threshold of 1.0. All models were trained under single-frame setting. The same training settings described above were used for our analysis. For estimating top-1 eigenvalue of Hessian matrix, we follow the power iteration method [8] with maximum 100 steps and a tolerance of 0.001. For the binning procedure, we set the bin edges as $b_k = u_{min} + k \frac{u_{max} - u_{min}}{K}$ for the experiments related to the eigenvalues of the Hessian matrix in order to maximize differences in DAR across subsets for clear visualization. For all other experiments, we defined the bin edges by $b_k = u_{(\lceil \frac{kN}{K} \rceil)}$ with sorted u in ascending order. Lastly, All experiments were conducted on a single NVIDIA A6000 GPU paired with an AMD EPYC 74F3 CPU.

B Loss Landscape Visualizations of Other Representative Networks

We present the local and global loss landscapes visualizations for representative network architectures (CNN, GCN, and Transformer) of 3D HPE when K is 3 on H36M [2] in Figure 2. Our results show that, for all representative models, the global loss landscape is complicated and contains multiple local minima that are disconnected. This demonstrates the inherent optimization difficulty of the 3D HPE task. Furthermore, we visualize the global loss landscape of VPose [6] with our adaptive scaling mechanism. As shown in Figure 1, the proposed adaptive scaling mechanism smooths the loss landscape, effectively alleviating the challenges posed by its complex structure.

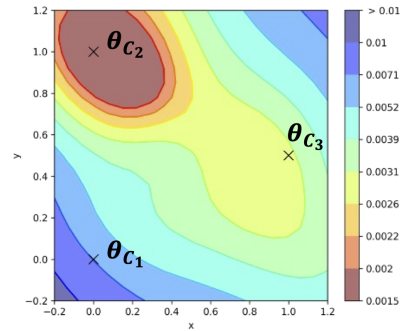


Figure 1: Global loss landscape of VPose [6] with adaptive scaling mechanism on H36M [2].

C Results for various K values

To further expand the visualization of loss landscape of the Section 2.4 in the main paper, we investigate the training loss along the line between each local minimum when K is 4,5,6,7,8, similar to the Figure 5(c) in the main paper. There exist multiple high-loss barriers between each local minimum as shown in the Figure 3. This means that our analysis is not sensitive to the value of K . Considering the inherent redundancy and imbalance in 3D HPE datasets, this behavior is likely to generalize across different values of K .

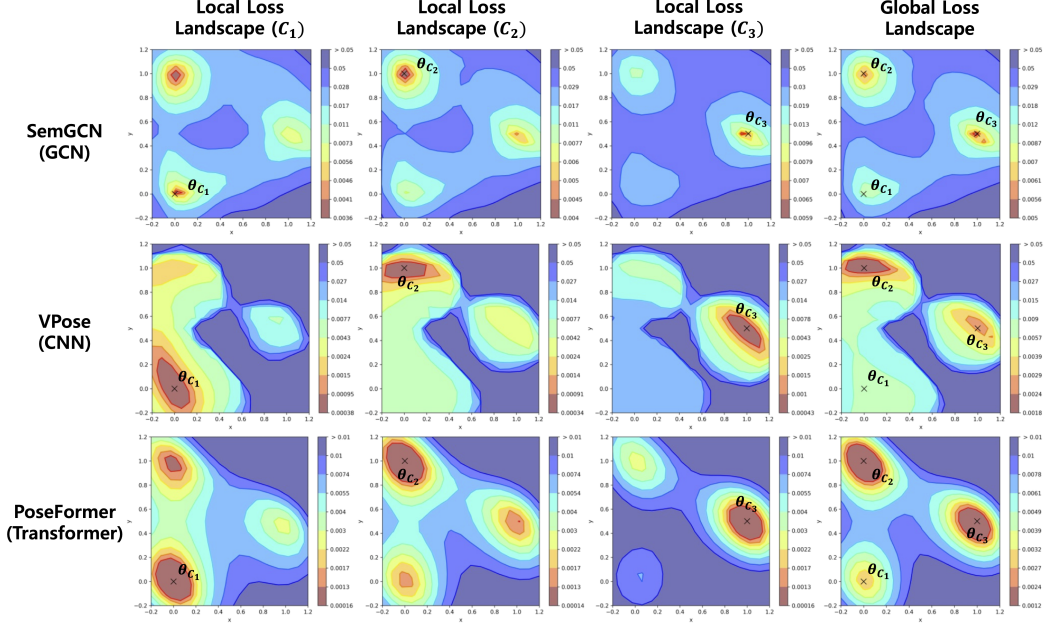


Figure 2: The local and global loss landscape of SemGCN [9], VPoser [6], and PoseFormer [10] when K is 3. θ_{C_1} , θ_{C_2} , and θ_{C_3} are model parameter around local minimum of each local loss landscape. Note that the local loss landscape of C_k is a result when the model trained with only C_k .

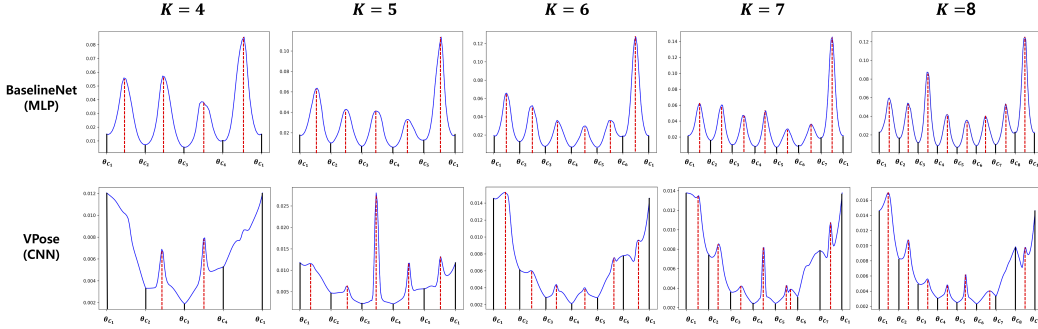


Figure 3: Training loss along the line between local minimum ($\theta_{C_1}, \dots, \theta_{C_K}$) of local loss landscape for various values of K . Each red dashed line represents loss barrier on the interpolated line between each local minimum. We utilized train set of H36M [2] for this experiment.

D Effect of Number of Regression Heads

We analyze the impact of the number of regression heads in our method on the BaselineNet [5] and SemGCN [9], as reported in Table 1. The results suggest that while using multiple heads generally improves performance compared to a single-head setup, the gains do not scale proportionally with the number of heads. This indicates that the number of heads is not a dominant factor in the overall effectiveness of our approach. Therefore, our method remains practical and efficient, as it does not rely heavily on tuning the number of heads to achieve strong performance.

E Weight Initialization for Each Head

To encourage each head to explore distinct regions of the loss landscape, we initialized their weights independently. Specifically, we applied orthogonal weight initialization [7] to each head. To further promote diversity in weight scales, we added linearly increasing offsets sampled from a

Table 1: The MPJPE depending on the number of heads on H36M [2].

Method	#1	#2	#3	#5	#10
BaselineNet [5]	54.2	53.7	53.6	53.8	53.7
SemGCN [9]	61.6	58.2	59.9	58.2	58.3

standard Gaussian distribution to each head in sequence. The overall weight initialization strategy is summarized as follows:

$$\text{orthogonal_init}(W_i) + 0.1 \times i \times d_i, \quad (1)$$

where W_i is weight of i -th head and d_i is a value sampled from standard Gaussian distribution $N(0, I)$. This head-specific weight initialization facilitates diverse exploration by encouraging each head to navigate a distinct region of the loss landscape.

F Computational Efficiency

We present the computational efficiency of our method in Table 2. For our method, we set the number of heads to 3. The size of input 2D pose is (1,32) for 16 joints with batch size 1 for measuring inference time and FLOPs.

Table 2: Comparison on the number of parameters and FLOPs and inference time.

Method	Params.	FLOPs	Inference Time (per example)
BaselineNet [5]	4.29M	4.29M	0.89ms
+ Ours	4.39M	4.39M	1.18ms
VPose [6]	8.49M	8.49M	1.23ms
+ Ours	8.59M	8.59M	1.68ms
SemGCN [9]	0.27M	4.84M	5.51ms
+ Ours	0.28M	4.88M	6.78ms
PoseFormer [10]	8.47M	9.06M	4.07ms
+ Ours	9.14M	9.73M	5.73ms

G Comparison with Other Ensembling Methods

To further validate the effectiveness of our ensemble method, we compare it with Stochastic Weight Averaging (SWA) [3], a technique designed to explore flatter regions of the loss landscape by averaging model weights. We conduct experiments on both BaselineNet [5] and PoseFormer [10] for 60 epochs, as summarized in Table 3. The results demonstrate that our method consistently outperforms SWA. Notably, SWA fails to improve performance for the transformer-based PoseFormer. We hypothesize that this is due to the sensitivity of attention mechanisms within transformer architectures because averaging the weights of attention modules likely disrupts their finely tuned dynamics, leading to degraded performance.

H Various settings for SAM

It is well-known that the effectiveness of SAM [1] is sensitive to its hyperparameters. Specifically, the perturbation radius often requires careful tuning based on the specific model and task. Therefore, we present the results of applying SAM to 3D HPE, both with and without an ensemble, using various perturbation radii. As shown in Table 4, applying SAM alone to 3D HPE does not yield a significant performance gain. On the other hand, the SAM + ensemble strategy shows performance gains over SAM alone in all cases. This indicates that the ensemble strategy is complementary to the loss landscape flattening method.

Table 3: Results of other ensembling method on H36M [2]. Note that Ensemble only represents multi-head ensemble without adaptive scaling mechanism. The number of heads for our method was set to 3.

Method	MPJPE	PA-MPJPE
BaselineNet [5]	54.8	43.8
+ Ensemble only	73.3	58.1
+ SWA [3]	58.3	45.1
+ Ours	53.6	42.7
PoseFormer [10]	56.2	44.3
+ Ensemble only	55.9	44.7
+ SWA [3]	89.4	66.8
+ Ours	54.5	43.9

Table 4: The MPJPE depending on the perturbation radius of SAM [1] on H36M [2].

Method	SAM Perturbation Radius				
	0.01	0.02	0.03	0.05	0.07
BaselineNet [5] + SAM	55.0	55.5	55.8	56.1	56.5
BaselineNet [5] + SAM + Ens.	54.2	54.4	54.5	54.5	54.6
VPose [6] + SAM	56.3	56.7	57.1	57.3	57.8
VPose [6] + SAM + Ens.	55.0	55.1	55.3	55.3	55.5

I Limitations

Our method consists of adaptive scaling mechanism and ensembling strategy. While our adaptive scaling mechanism is motivated by the connection between redundancy and flat loss regions, future work could explore a more rigorous mathematical understanding of the flatness induced by input-dependent scaling. For ensemble strategy, it is relatively simple based on multiple regression heads. While this enables efficient training and inference, more advanced ensemble strategies (*e.g.*, diversity-aware training or Bayesian ensembling) could potentially offer additional gains. Generally, while our method is relatively simple, it introduces a slight increase in inference time, which may be a limitation in latency-sensitive applications.

References

- [1] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*. International Conference on Learning Representations, 2020.
- [2] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, 2013.
- [3] Pavel Izmailov, Dmitrii Podoprikin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.
- [4] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [5] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2640–2649, 2017.
- [6] Dario Pavlo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7753–7762, 2019.

- [7] A Saxe, J McClelland, and S Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *Proceedings of the International Conference on Learning Representations*. International Conference on Learning Representations, 2014.
- [8] Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael W Mahoney. Pyhessian: Neural networks through the lens of the hessian. In *IEEE International Conference on Big Data*, pages 581–590. IEEE, 2020.
- [9] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3425–3435, 2019.
- [10] Ce Zheng, Sijie Zhu, Matias Mendieta, Taojiannan Yang, Chen Chen, and Zhengming Ding. 3d human pose estimation with spatial and temporal transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11656–11665, 2021.