Mining Attribute Subspaces for Efficient Fine-tuning of 3D Foundation Models

Supplementary Material

1. Implementation Details

During subspace extraction and fine-tuning for downstream applications, we fine-tune the model from the pretrained VGGT-1B checkpoint and freeze the DINO encoder to save memory. For the aggregator, depth head, and camera head, we use a cosine learning rate schedule with warm-up. During the warm-up phase, the learning rate linearly increases from 1×10^{-8} to 1×10^{-5} over the first 5% of the total training steps. Following the warm-up, the learning rate then decays following a cosine schedule, dropping to 1×10^{-8} over the remaining training steps. To stabilize training, we apply gradient norm clipping at 0.5.

During the fine-tuning phase, we randomly sample 4-16 views per scene. The network is trained for a total of 24,000 steps, with each step processing 32 images as input. The entire training process requires approximately 20 hours using a single NVIDIA H200 GPU.

The dataset used for subspace extraction consists of 200 generated scenes, each rendered as a sequence of 100 images.

2. Additional Results

2.1. Qualitative Results

We present additional qualitative results of point cloud reconstruction on the synthetic test dataset in Figure 1, which further demonstrate the robustness of our method across different scenes. Our method produces the most accurate reconstructions with noticeably fewer artifacts, showing its transferability to out-of-distribution data.

We present more visualizations of clothed human reconstruction in Figure 2. The first two rows are from the THuman dataset [4], and the last two rows are from the 2K2K dataset [1]. The comparison shows that our method exhibits strong robustness.

2.2. Different Rank Allocation Strategies

Table 1. **Comparison Between Different Rank Allocation Strategies.** The overall trend is the same, with minimal performance differences.

Method	**THuman** (In-domain)			**2K2K** (Cross-domain)		
	Acc ↓	Comp ↓	NC ↑	Acc ↓	Comp ↓	NC ↑
Uniform (d=16)	3.392	2.138	90.91	3.019	1.887	91.70
Uniform (d=32)	3.332	2.220	91.56	2.825	1.878	93.00
Uniform (d=64)	**2.745**	**1.882**	**91.82**	**2.513**	**1.754**	**93.56**
Importance (d=16)	4.088	3.783	88.81	3.822	5.900	90.29
Importance (d=32)	3.778	2.444	91.23	3.165	2.041	92.41
Importance (d=64)	**2.766**	**1.912**	**92.03**	**2.481**	**1.762**	**93.54**

In the Experiment section, we report the results of the method that applied the same d to different layers. Another popular importance rank allocation strategy is based on the effective rank. The effective rank of a matrix W is defined using its Frobenius and spectral norms as:

$$\text{EffectiveRank}(W) = \left(\frac{\|W\|_{\mathcal{F}}}{\|W\|_2}\right)^2.$$

Therefore, the target subspace dimension d for each layer can be dynamically allocated based on this measure. Specifically, the layer-wise subspace size d_l is determined by:

$$d_l = d \times \left\lfloor \frac{\text{EffectiveRank}(W_l)}{\text{AverageEffectiveRank}} \right\rfloor,$$

where d is the global budget for the subspace size.

We present the performance comparison between Uniform and Importance allocation strategies in Table 1. Uniform refers to the allocation strategy reported in the main text, while Importance is based on effective rank-based importance allocation. We observe that the overall trend is the same, and the performance differences are negligible.

3. Details on Synthetic Dataset Generation

In this section, we first briefly introduce the previous work, Megasynth [2], and then describe the generation process of our synthetic datasets.

Megasynth is a pipeline designed for generating synthetic non-semantic datasets. Using scalability and controllability, we can synthesize datasets tailored to exhibit target 3D attribute variations. The generation process begins with creating the layout of indoor scenes by filling the space with boxes of varying sizes. Next, it generates the scene geometry and samples the textures. The geometry is constructed from primitives (such as ellipsoids, cubes, and cylinders) instantiated within each box. To maximize geometric variation, a height field is randomly assigned to each surface. Textures are sampled from the MatSynth texture dataset [3]. During the rendering phase, the light sources are randomized, followed by a random sampling of both the camera distribution and the intrinsic camera parameters.

For the first experiment, 2D Face Anti-Spoofing, we utilized two distinct subspaces: texture and geometry. Each subspace was extracted from five different LoRA adapters. These ten datasets (for subspace extraction) and the datasets employed for fine-tuning were rendered under micro-baseline settings.

The micro-baseline setting emphasizes that camera movements were minimal. This was achieved by interpolating the camera's translation and rotation across control

Figure 1. More visual comparison of 2D Face Anti-Spoofing Tasks.

points. By ensuring that both the translational displacement and the angular differences between these control points remained within a predefined range, the overall movement of the camera trajectory was kept minimal.

In the second experiment, Clothed Human Reconstruction, we used four subspaces: texture, geometry, camera motion, and lighting. Each of these four subspaces was extracted from ten different LoRA adapters. These datasets were object-centered. To achieve this, we made slight modifications to the standard layout sampling strategy: we removed the walls, ceiling, and floor of the room, leaving only the synthetic boxes centrally arranged.

Next, we explain how we customized the dataset generation with respect to these four specific variations. For texture variation, we minimized geometry changes: the scene file was fixed entirely (in the first experiment), or the number of boxes and primitives was limited to introduce only slight geometry variance (in the second experiment), while textures were sampled broadly from the entire texture dataset. For geometry variation, we allowed texture sampling to repeatedly use a subset of the full texture dataset, with different subsets used across different datasets, while varying the box and primitive counts to maximize geometry diversity. To isolate camera movement, we uniformly sampled camera azimuths and elevations on a sphere and randomized distances to define control points. Then, spline interpolation was performed between these points to generate the camera's movement trajectory; different datasets used varying ranges for these orientation targets and distances. Finally, for the lighting variation, we place some sunlight sources in the Blender environment and randomly assign their color and strength for each instance. All images were rendered at a resolution of 518×518 to align with DINO2's patchify process.

4. Singular Values of Matrix C

In this section, we will present the distribution of the singular values of the matrices C during the first iteration of the subspace extraction process. Note that a logarithmic scale has been applied. The significant drop in these curves indicates the existence of shared subspaces.

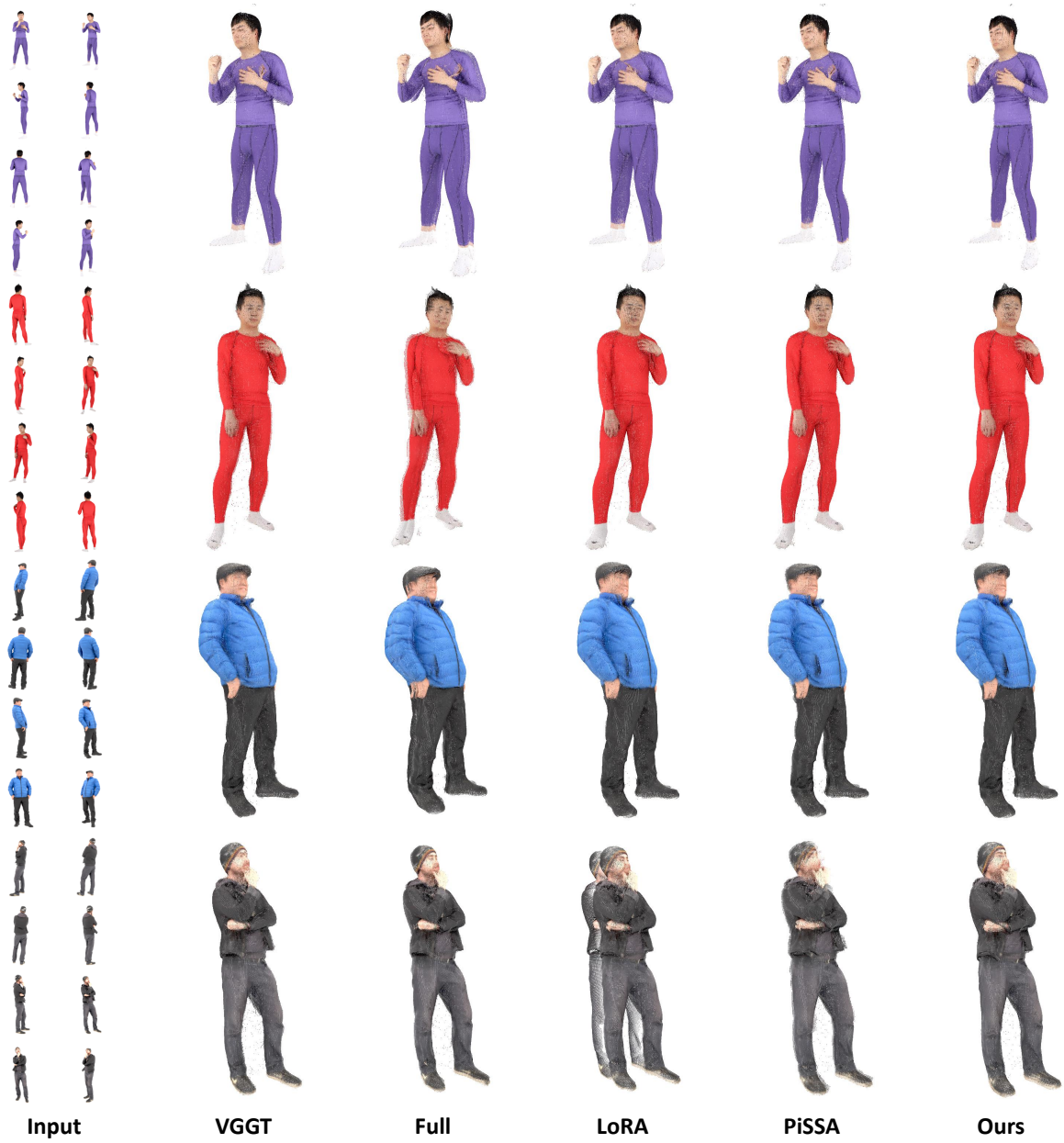

Figure 2. More visual comparison of Clothed Human Reconstruction Tasks.

4.1. Texture Subspace

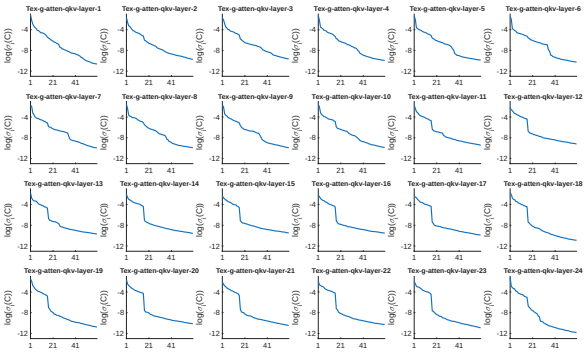

Figure 3. Singular values of the QKV matrix in the global attention layer with respect to texture variations.

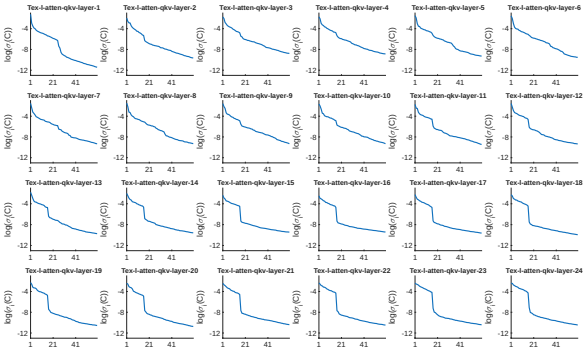

Figure 4. Singular values of the QKV matrix in the frame attention layer with respect to texture variations.

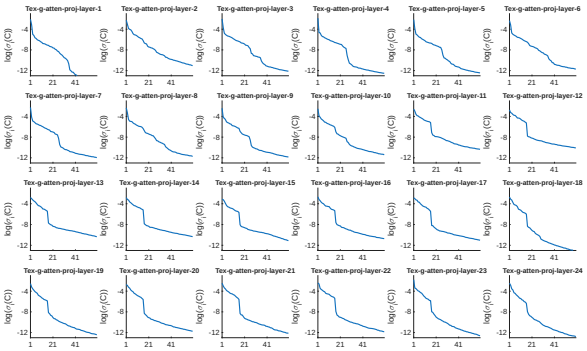

Figure 5. Singular values of the projection matrix in the global attention layer with respect to texture variations.

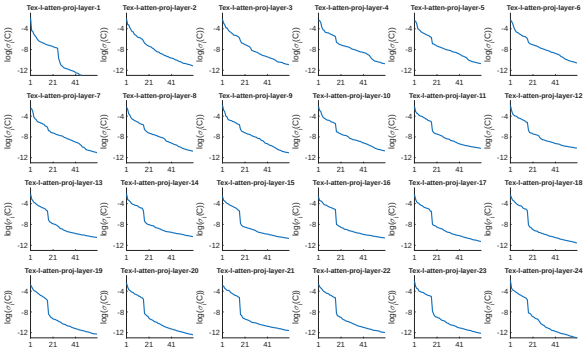

Figure 6. Singular values of the projection matrix in the frame attention layer with respect to texture variations.

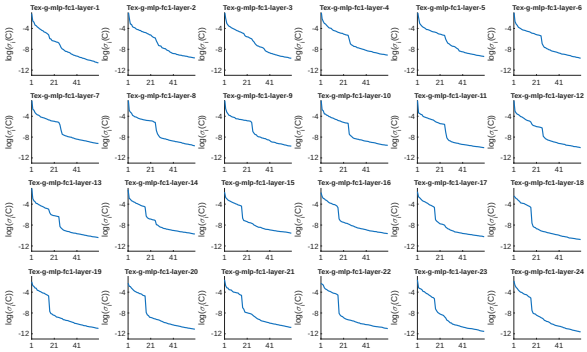

Figure 7. Singular values of the first fully connected matrix in the global attention layer with respect to texture variations.

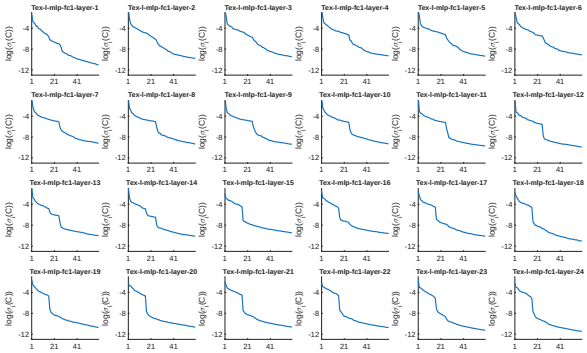

Figure 8. Singular values of the first fully connected matrix in the frame attention layer with respect to texture variations.

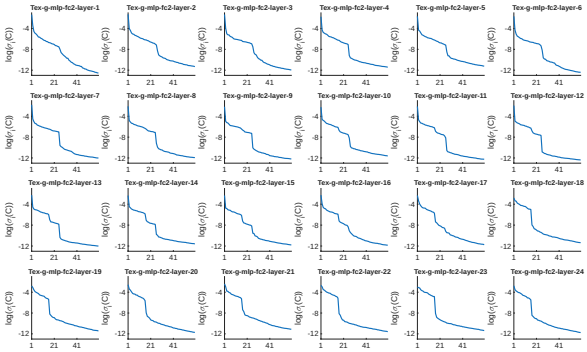

Figure 9. Singular values of the second fully connected matrix in the global attention layer with respect to texture variations.

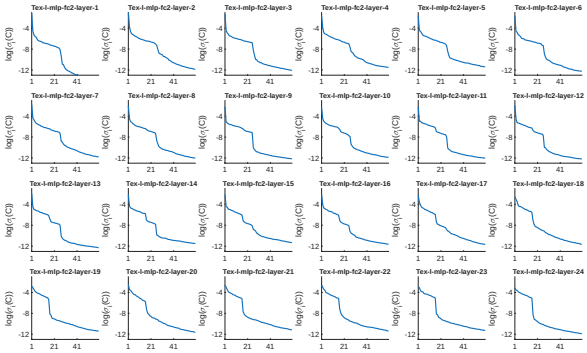

Figure 10. Singular values of the second fully connected matrix in the frame attention layer with respect to texture variations.

4.2. Geometry Subspace

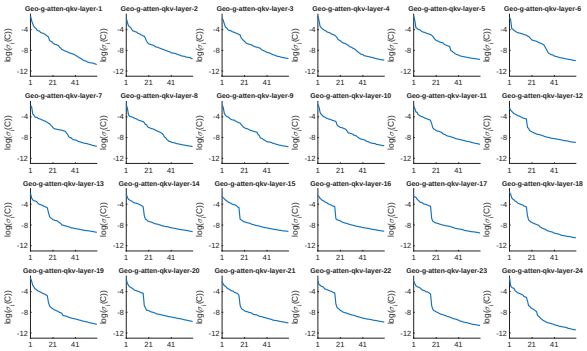

Figure 11. Singular values of the QKV matrix in the global attention layer with respect to geometry variations.

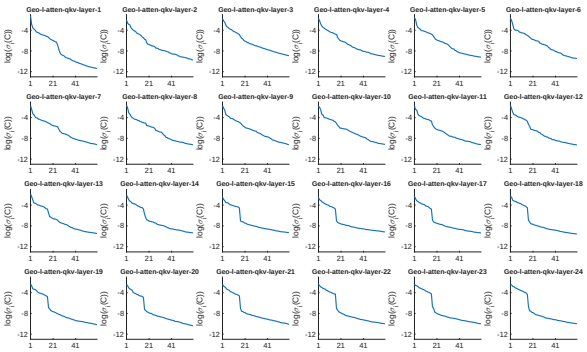

Figure 12. Singular values of the QKV matrix in the frame attention layer with respect to geometry variations.

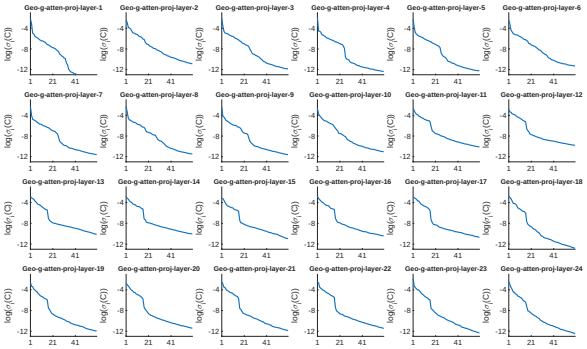

Figure 13. Singular values of the projection matrix in the global attention layer with respect to geometry variations.

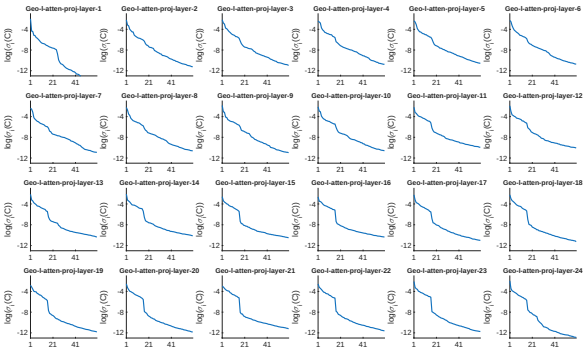

Figure 14. Singular values of the projection matrix in the frame attention layer with respect to geometry variations.

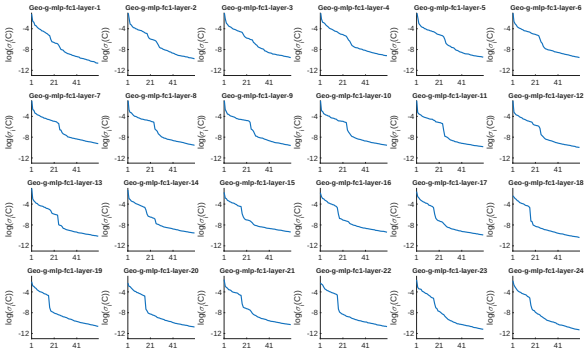

Figure 15. Singular values of the first fully connected matrix in the global attention layer with respect to geometry variations.

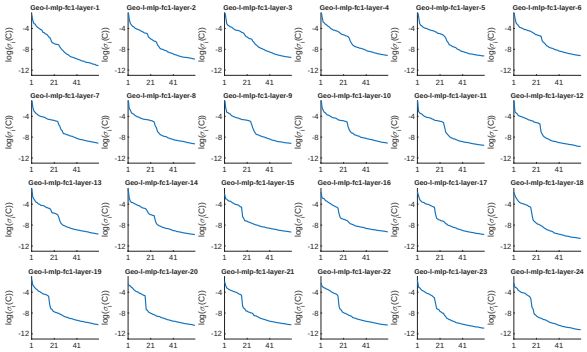

Figure 16. Singular values of the first fully connected matrix in the frame attention layer with respect to geometry variations.

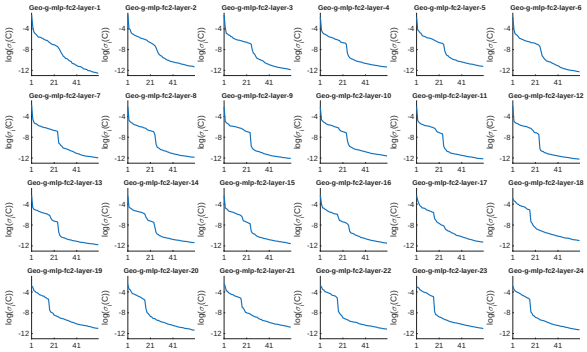

Figure 17. Singular values of the second fully connected matrix in the global attention layer with respect to geometry variations.

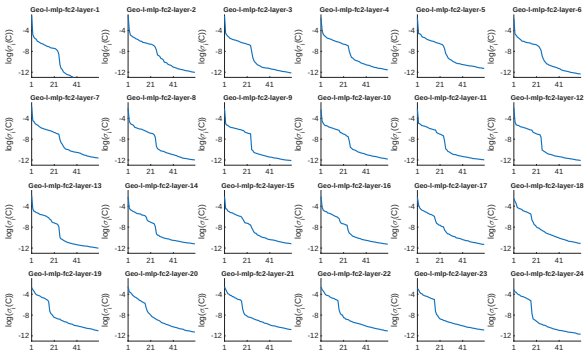

Figure 18. Singular values of the second fully connected matrix in the frame attention layer with respect to geometry variations.

4.3. Camera Motion Subspace

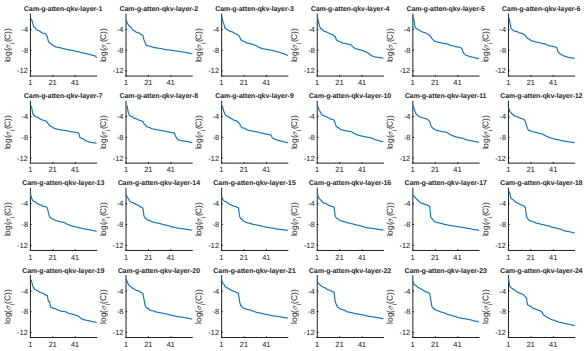

Figure 19. Singular values of the QKV matrix in the global attention layer with respect to camera variations.

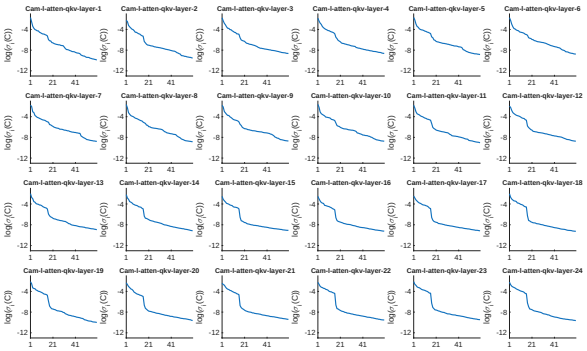

Figure 20. Singular values of the QKV matrix in the frame attention layer with respect to camera variations.

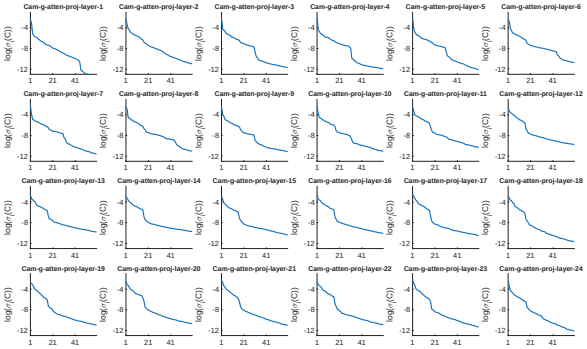

Figure 21. Singular values of the projection matrix in the global attention layer with respect to camera variations.

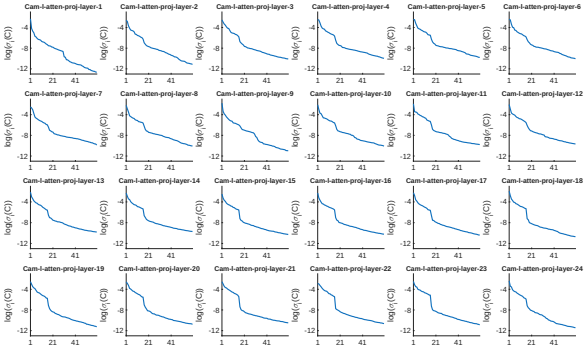

Figure 22. Singular values of the projection matrix in the frame attention layer with respect to camera variations.

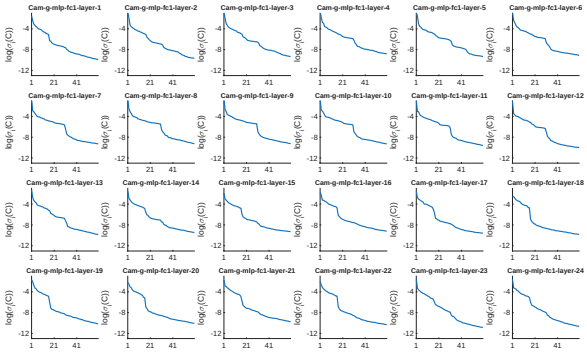

Figure 23. Singular values of the first fully connected matrix in the global attention layer with respect to camera variations.

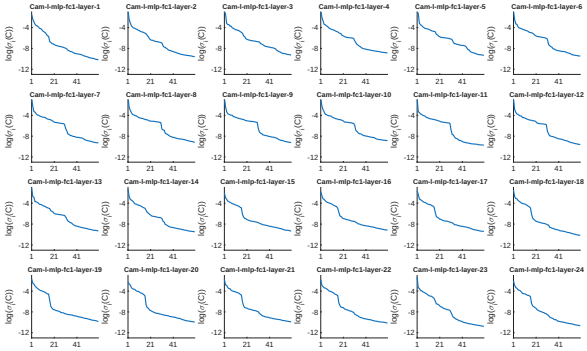

Figure 24. Singular values of the first fully connected matrix in the frame attention layer with respect to camera variations.

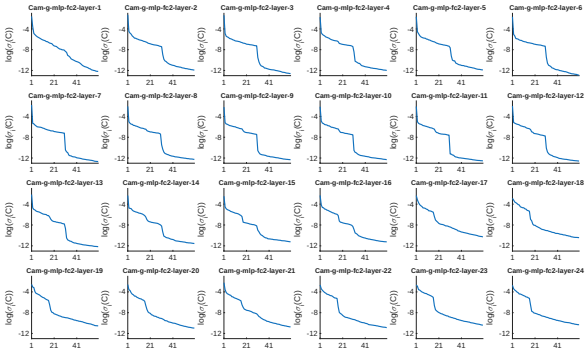

Figure 25. Singular values of the second fully connected matrix in the global attention layer with respect to camera variations.

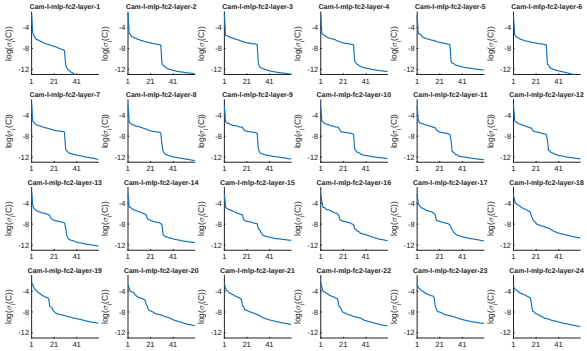

Figure 26. Singular values of the second fully connected matrix in the frame attention layer with respect to camera variations.

4.4. Lighting Subspace

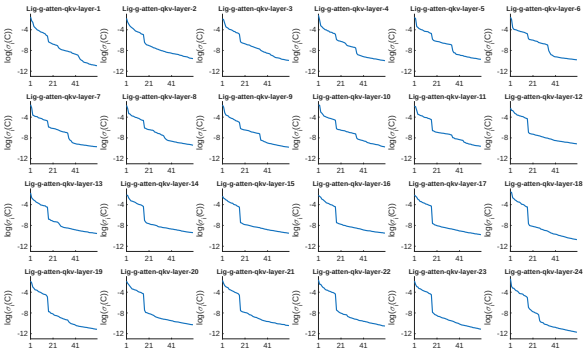

Figure 27. Singular values of the QKV matrix in the global attention layer with respect to lighting variations.

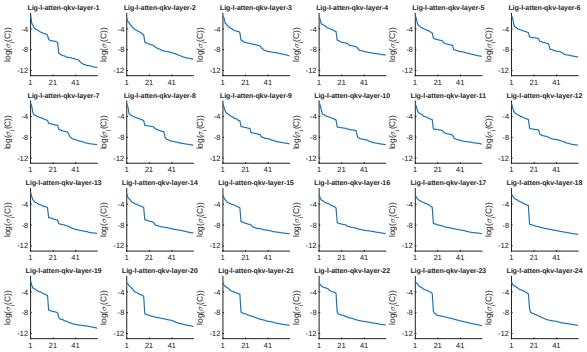

Figure 28. Singular values of the QKV matrix in the frame attention layer with respect to lighting variations.

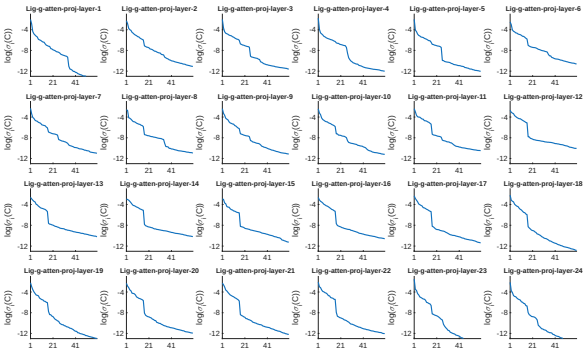

Figure 29. Singular values of the projection matrix in the global attention layer with respect to lighting variations.

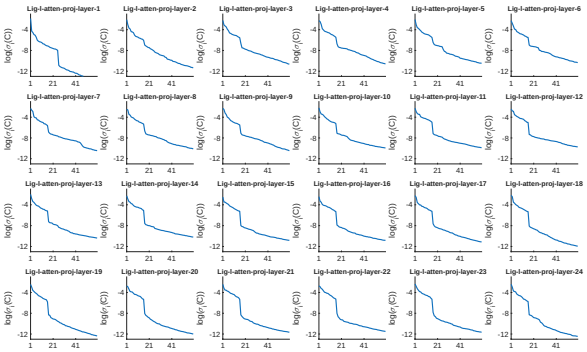

Figure 30. Singular values of the projection matrix in the frame attention layer with respect to lighting variations.

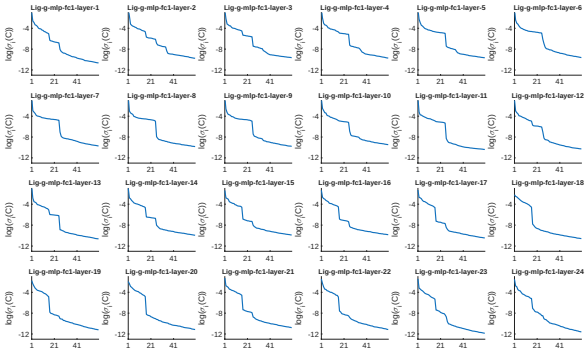

Figure 31. Singular values of the first fully connected matrix in the global attention layer with respect to lighting variations.

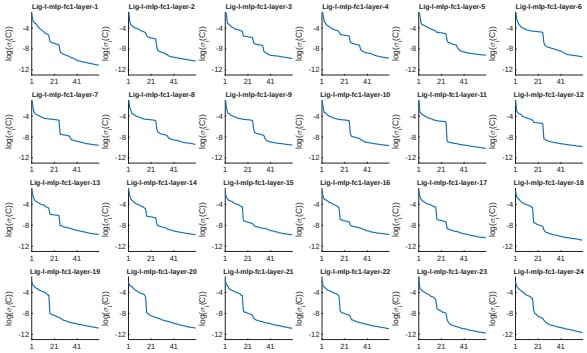

Figure 32. Singular values of the first fully connected matrix in the frame attention layer with respect to lighting variations.

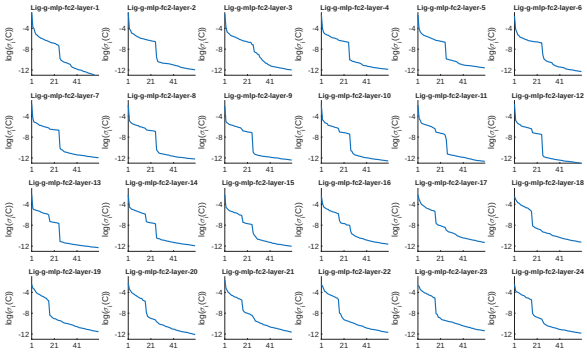

Figure 33. Singular values of the second fully connected matrix in the global attention layer with respect to lighting variations.

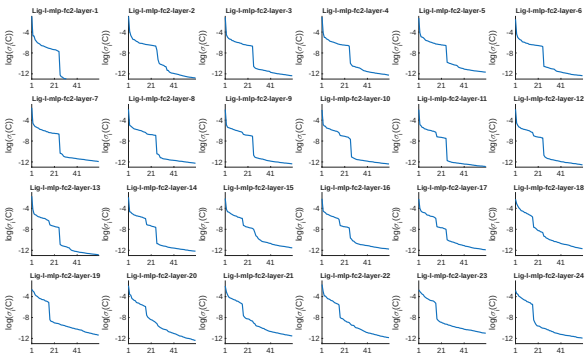

Figure 34. Singular values of the second fully connected matrix in the frame attention layer with respect to lighting variations.

5. Subspace Orthogonality Analysis

In this section, we present the distribution of the generalized eigenvalues λ used in our subspace orthogonality analysis. Results are shown for all six pairs of subspaces. The eigenvalue acts as a reprojection error, where values closer to 1 indicate greater orthogonality between two subspaces. The curves show that these six pairs of subspaces are approximately orthogonal to each other.

5.1. Geometry vs Texture

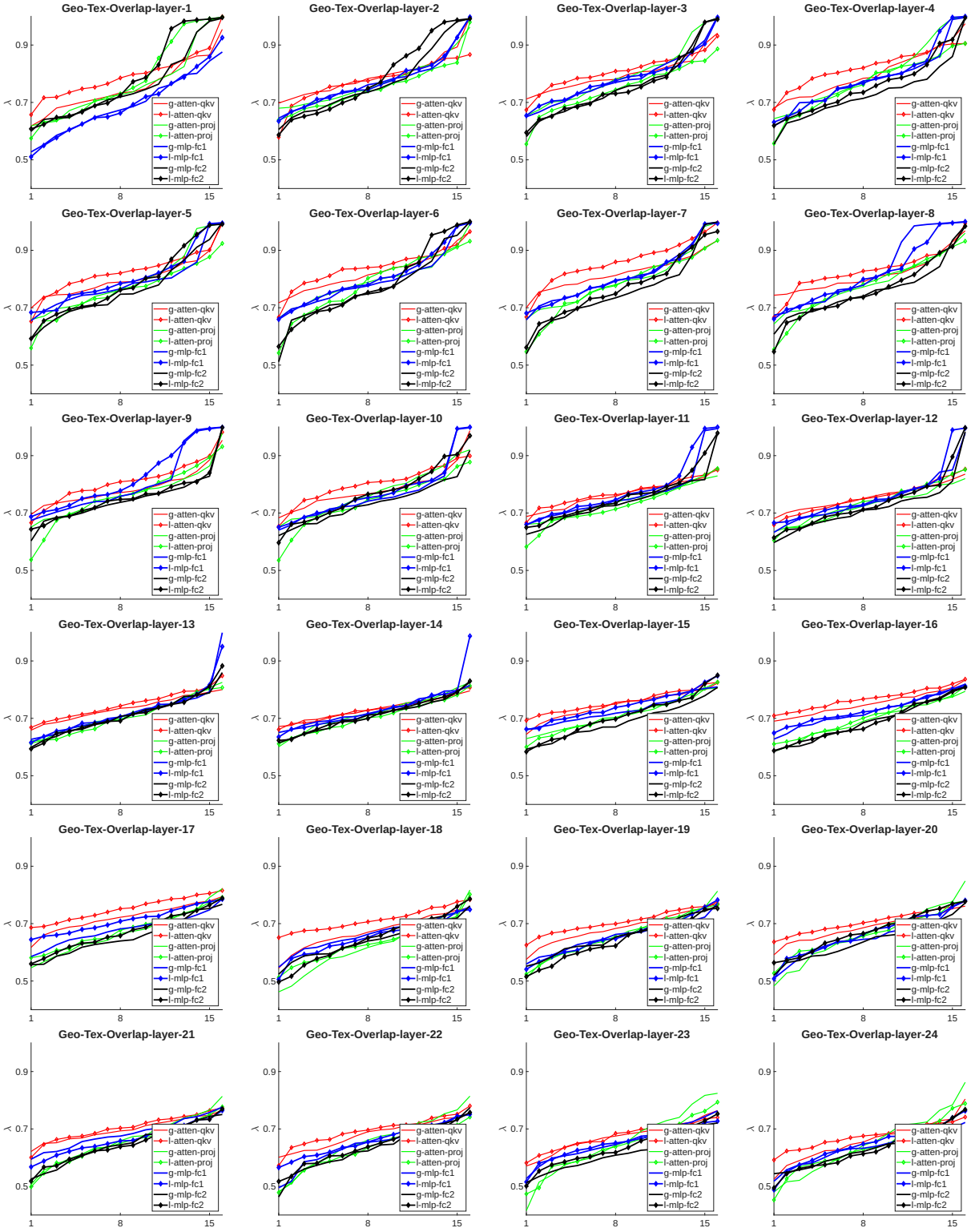

Figure 35. The overlap ratio between subspaces that correspond to variations in geometry and texture.

5.2. Geometry vs Camera

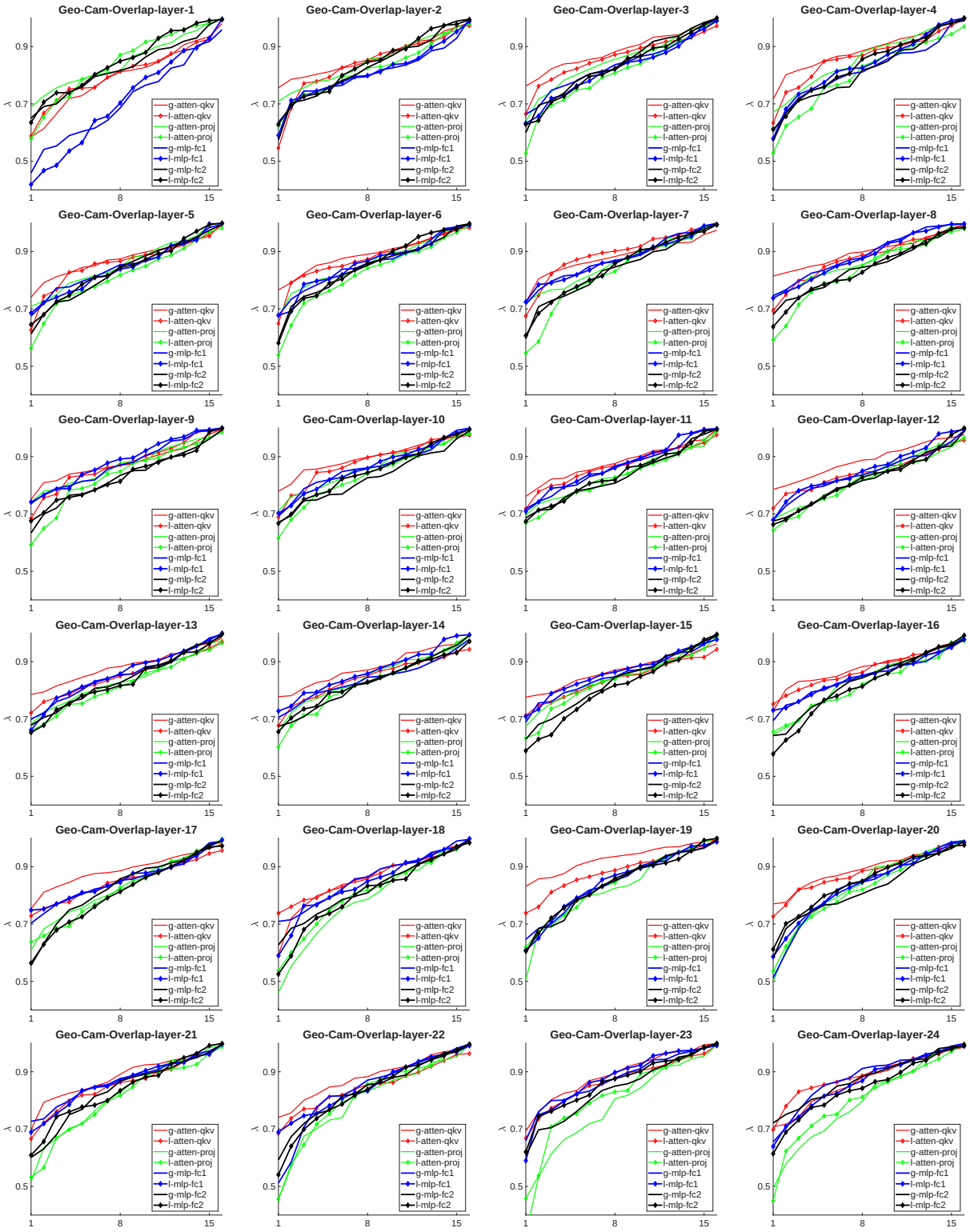

Figure 36. The overlap ratio between subspaces that correspond to variations in geometry and camera motion.

5.3. Geometry vs Lighting

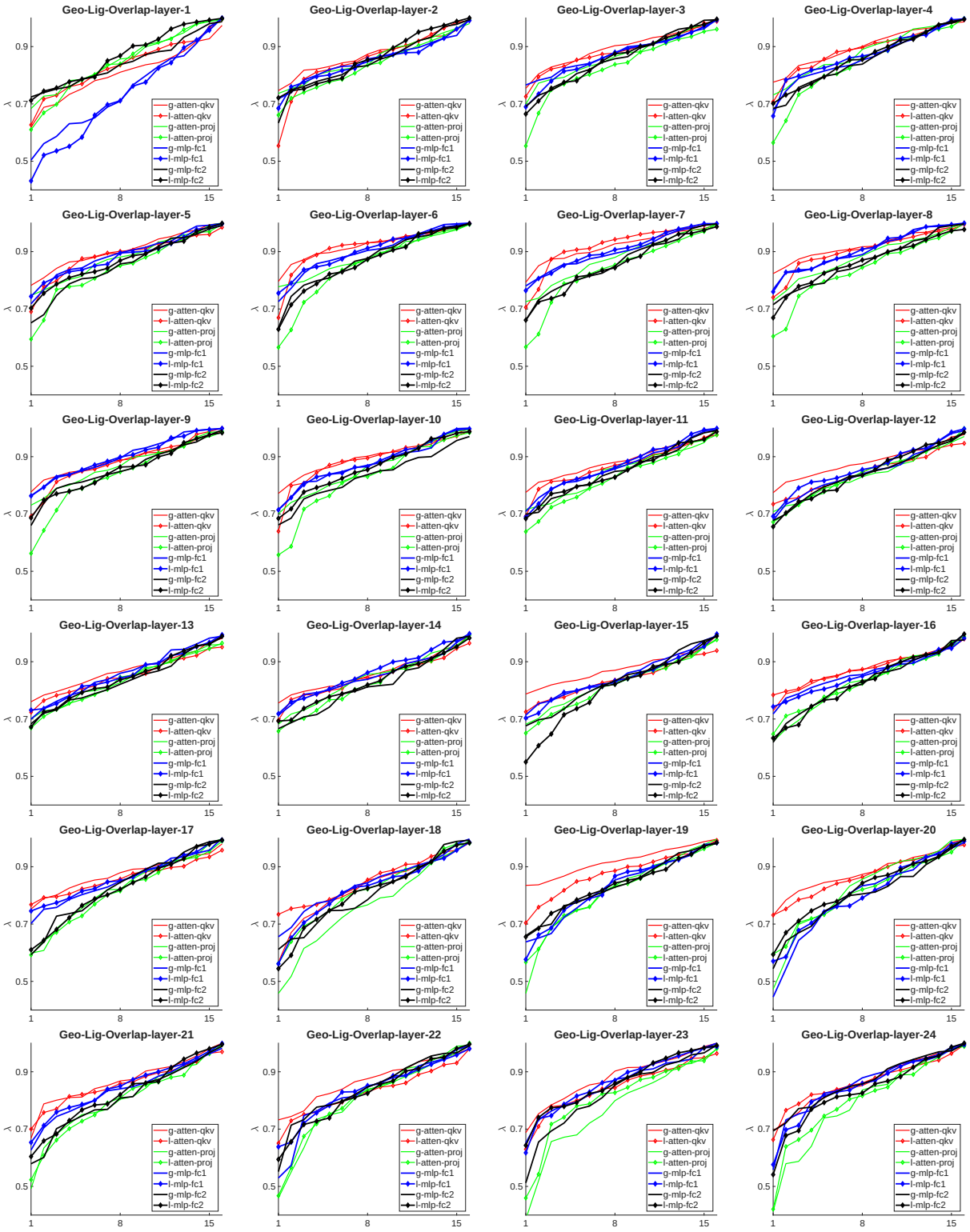

Figure 37. The overlap ratio between subspaces that correspond to variations in geometry and lighting.

5.4. Texture vs Camera

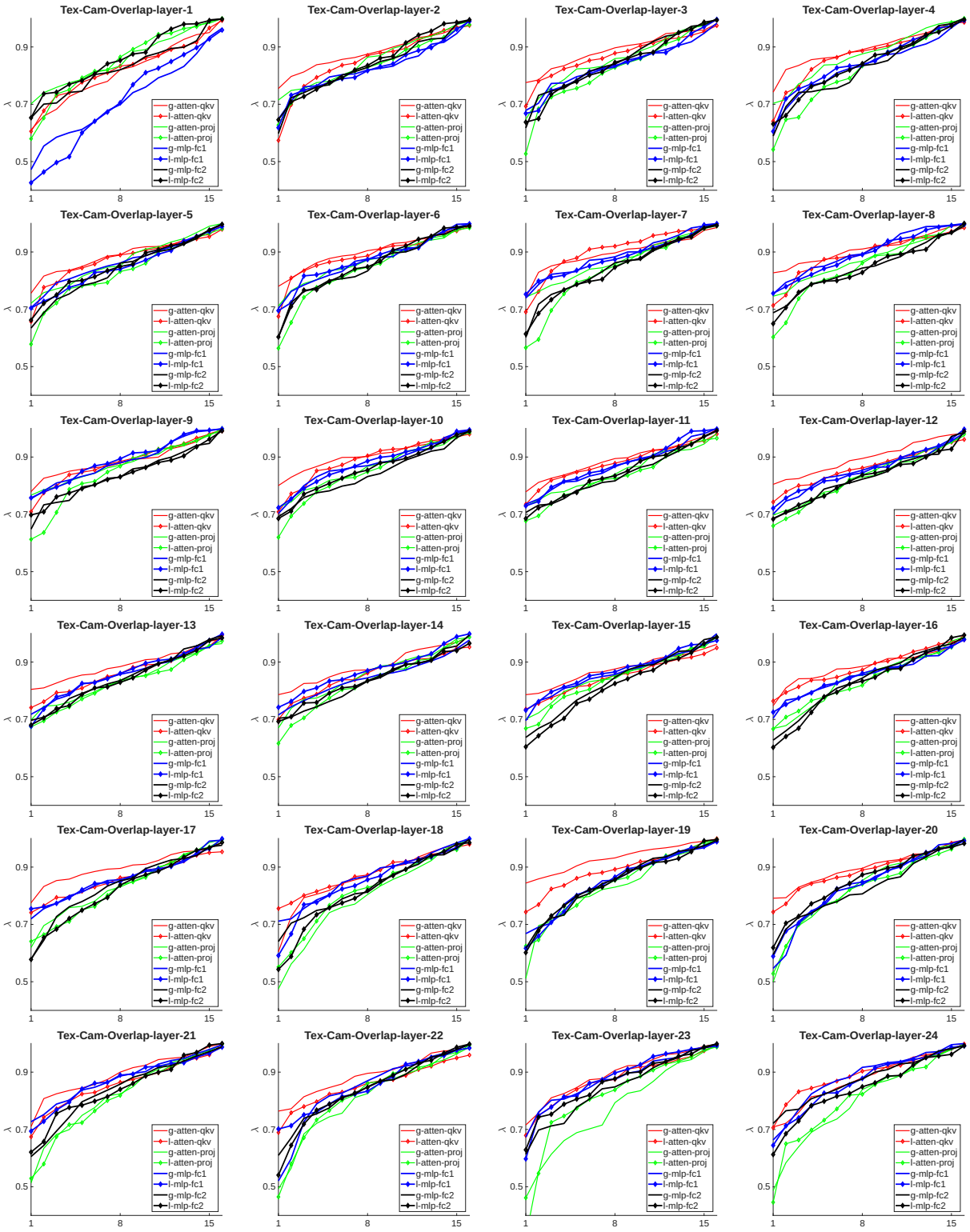

Figure 38. The overlap ratio between subspaces that correspond to variations in texture and camera motion.

5.5. Texture vs Lighting

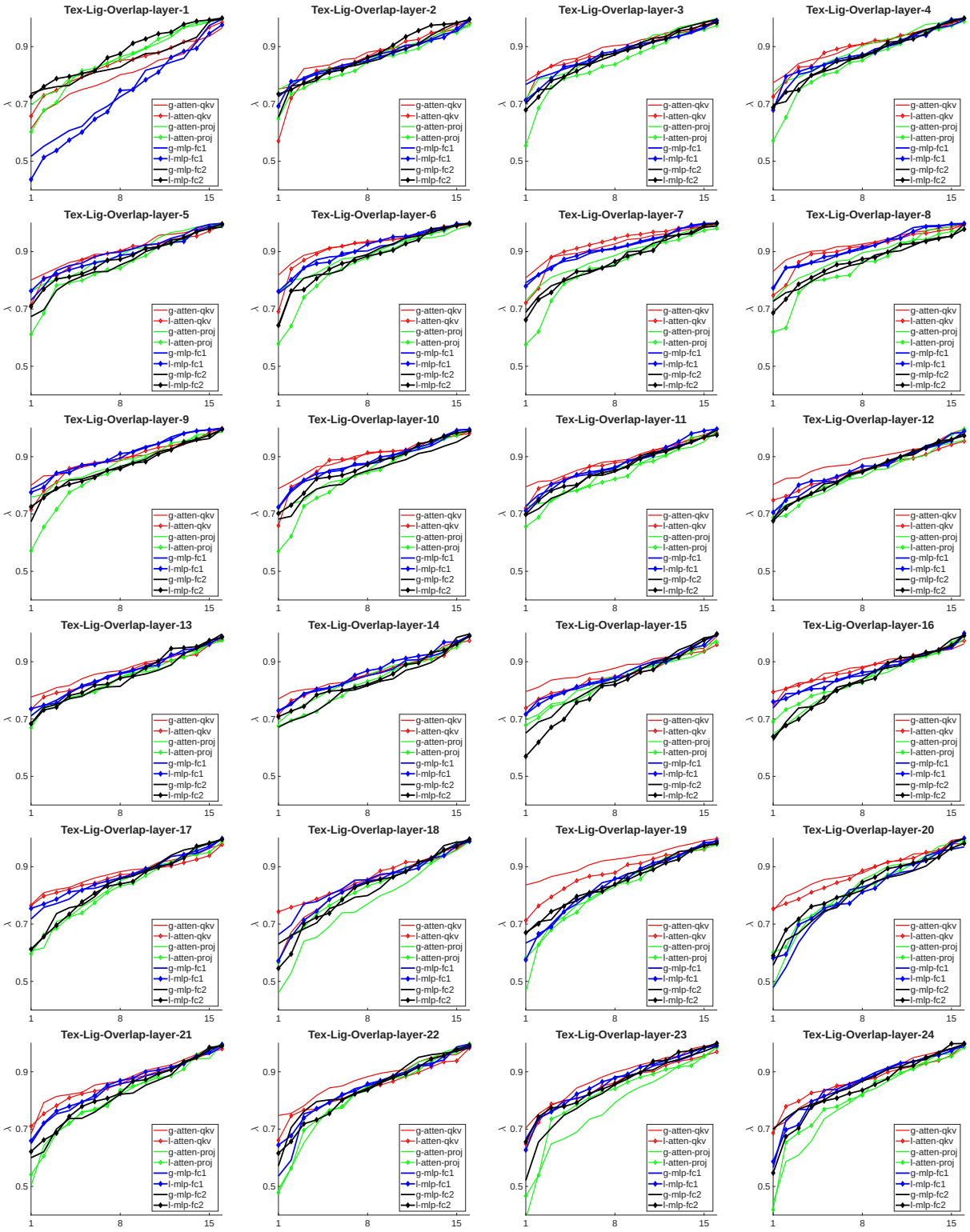

Figure 39. The overlap ratio between subspaces that correspond to variations in texture and lighting.

5.6. Camera vs Lighting

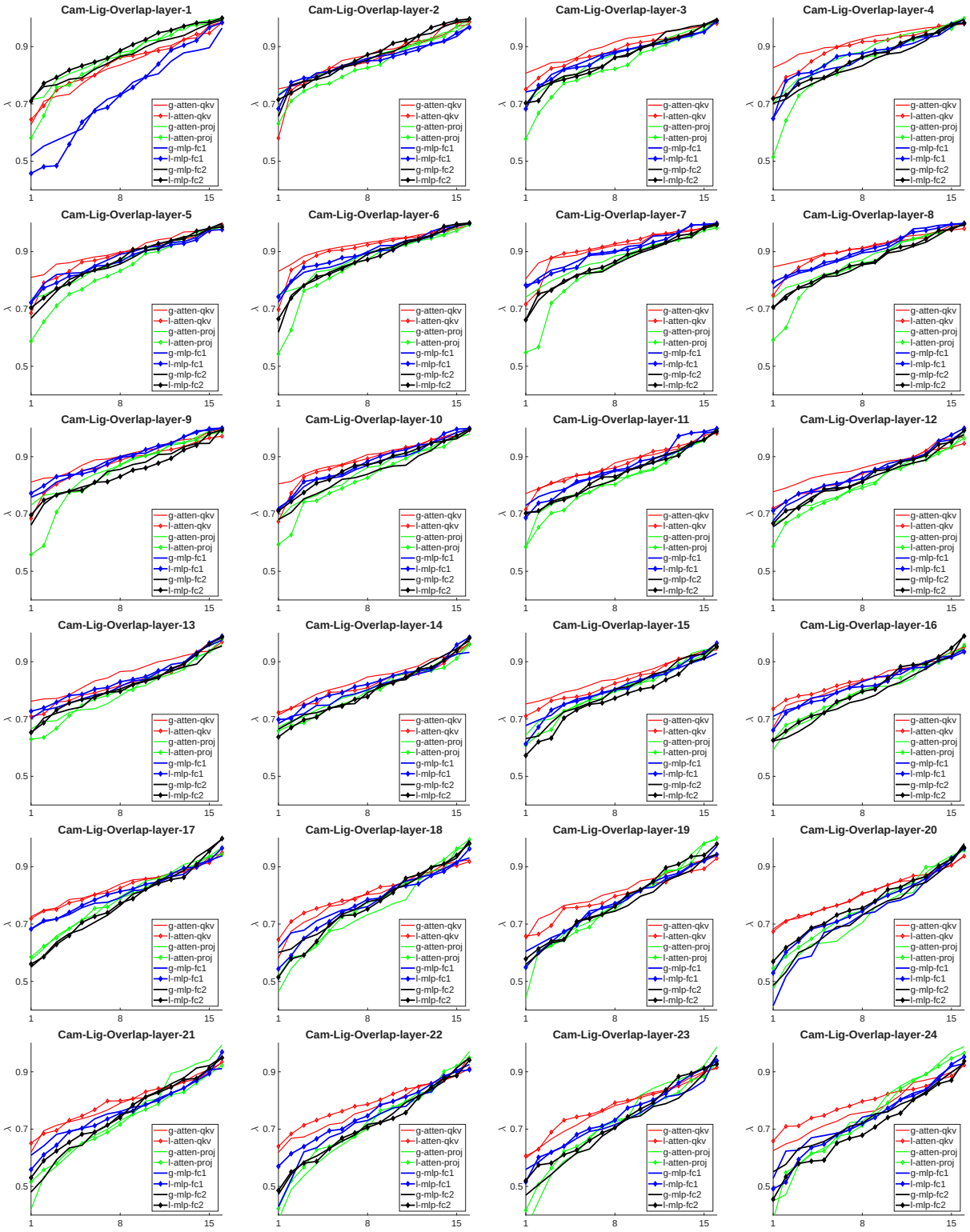

Figure 40. The overlap ratio between subspaces that correspond to variations in camera motion and lighting.

References

[1] Sang-Hun Han, Min-Gyu Park, Ju Hong Yoon, Ju-Mi Kang, Young-Jae Park, and Hae-Gon Jeon. High-fidelity 3d human digitization from single 2k resolution images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12869–12879, 2023. 1

[2] Hanwen Jiang, Zexiang Xu, Desai Xie, Ziwen Chen, Haian Jin, Fujun Luan, Zhixin Shu, Kai Zhang, Sai Bi, Xin Sun, et al. Megasynth: Scaling up 3d scene reconstruction with synthesized data. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16441–16452, 2025. 1

[3] Giuseppe Vecchio and Valentin Deschaintre. Matsynth: A modern pbr materials dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22109–22118, 2024. 1

[4] Tao Yu, Zerong Zheng, Kaiwen Guo, Pengpeng Liu, Qionghai Dai, and Yebin Liu. Function4d: Real-time human volumetric capture from very sparse consumer rgbd sensors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5746–5756, 2021. 1