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

Yu Jiang[1,2] Hanwen Jiang[3] Ahmed Abdelkader[4] Wen-Sheng Chu[4] Brandon Y. Feng[1]
Zhangyang Wang[1] Qixing Huang[1]
[1]The University of Texas at Austin [2]Shanghai Jiao Tong University [3]Adobe Research
[4]Google Research

Abstract

With the emergence of 3D foundation models, there is growing interest in fine-tuning them for downstream tasks, where LoRA is the dominant fine-tuning paradigm. As 3D datasets exhibit distinct variations in texture, geometry, camera motion, and lighting, there are interesting fundamental questions: 1) Are there LoRA subspaces associated with each type of variation? 2) Are these subspaces disentangled (i.e., orthogonal to each other)? 3) How do we compute them effectively? This paper provides answers to all these questions. We introduce a robust approach that generates synthetic datasets with controlled variations, fine-tunes a LoRA adapter on each dataset, and extracts a LoRA subspace associated with each type of variation. We show that these subspaces are approximately disentangled. Integrating them leads to a reduced LoRA subspace that enables efficient LoRA fine-tuning with improved prediction accuracy for downstream tasks. In particular, we show that such a reduced LoRA subspace, despite being derived entirely from synthetic data, generalizes to real datasets. An ablation study validates the effectiveness of the choices in our approach.

1. Introduction

Foundation Models [2], pretrained on large-scale datasets with large compute, serve as powerful foundations for solving various downstream tasks via suitable fine-tuning. A popular fine-tuning approach is LoRA [13] and its variants, which constrain the number of trainable parameters to mitigate the problems of limited labeled data and overfitting. In this paper, we study efficient fine-tuning strategies for 3D foundation models.

To adopt LoRA for 3D vision tasks, we argue that we need to understand how 3D data differ from other domains, and we provide two perspectives. First, in many cases, it remains costly to obtain even a small amount of real-wold 3D data. One such example is to differentiate real 3D face images and 2D face (printed) images from micro-baseline multi-view images. This task has important forensics applications such as anti-spoofing, and yet collecting micro-baseline video data is laborious with privacy issues. Second, 3D data supports 3D vision tasks that usually focus on low-level visual attributes, such as texture, geometry, camera motions, and lighting. Therefore, we ask a fundamental question: can we create large-scale synthetic data that has **a different data distribution** from real 3D data, while leveraging LoRA components to discover the underlying patterns of different visual properties that are transferable and can be used for improving fine-tuning performance?

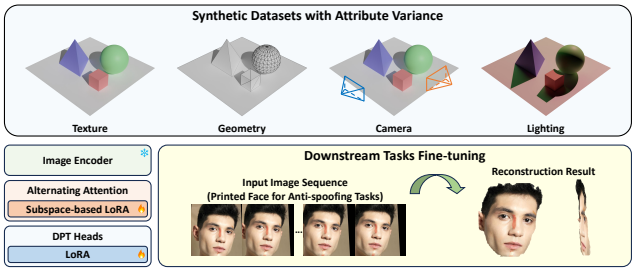

Figure 1. Our approach pre-computes LoRA subspaces associated with each type of 3D dataset variation in geometry, texture, camera, and lighting via curated synthetic datasets. These subspaces are integrated into a reduced LoRA basis for efficient fine-tuning.

In this paper, we provide positive answers to both questions. We show how to craft synthetic datasets with controlled variations of visual attributes to fine-tune VGGT [26]. We then develop an algorithm that, for each type of variation, extracts a shared subspace from the resulting LoRA adapters. Concatenating these shared spaces together yields a concise LoRA basis. We show the effectiveness of this basis across various downstream tasks, in which we improve both in-distribution tasks and out-of-distribution tasks. To compute the shared space of each attribute, we create synthetic datasets that hold all other attributes relatively fixed while varying the attribute of interest. Specifically, we create multiple data splits and apply LoRA-based fine-tuning on each of them. Each resulting LoRA displacement includes 1) a component shared across all LoRAs tuned for the specific attribute, which is also

the component that we aim to extract to represent the attribute, and 2) the components represent data-specific features, which shall be discarded. We formulate the extraction of the shared component as a generalized least squares optimization problem. Moreover, we assess the orthogonality between LoRA subspaces computed for different attributes and find that they are disentangled. Finally, we extract the principal component from the subspaces computed for all attributes, serving as the basis for fine-tuning on new data.

We evaluate our approach on the task of 3D face reconstruction from micro-baseline videos, 3D human reconstruction from a wide-baseline image setup, and transparent object reconstruction. Experimental results show that the subspaces discovered from our generated synthetic data are transferrable to real data, improving both efficiency and quality of fine-tuning results.

2. Related Work

3D foundation models. The 3D foundation models, e.g., DUSt3R [27], VGGT [26], and RayZer [14, 16, 39], have recently achieved strong performance in various 3D vision tasks. This has created two lines of follow-up work. The first line develops variants [3, 4, 29, 32, 33, 39] with expanded capabilities. Another line focuses on fine-tuning 3D foundation models (VGGT in particular) for various tasks [5, 18, 21, 30, 37], in which LoRA [13] is a widely used strategy. Due to the effectiveness of LoRAs for fine-tuning VGGT and the fact that 3D datasets exhibit disentangled variations in geometry, texture, camera, and lighting, this paper studies the connection between these variations and LoRA.

LoRA merging in generative models. Learning LoRA subspaces for distinct 3D variation factors and integrating them is closely related to rich prior work in image and video generation, which seeks to combine style LoRAs and content LoRAs. Early work, including BLoRA [11], ZipLoRA [23], and LoRA.rar [24], develops efficient training strategies to combine LoRAs. Specifically, BLoRA [11] identifies the transformer blocks responsible for style and content by curating the input conditions and learns to combine LoRAs from a single image. ZipLoRA [23] introduces column-specific weights to combine two separately trained LoRAs. LoRA.rar [24] trains a hypernetwork on LoRA corpus to predict column-wise coefficients, which are then used to fuse content and style LoRAs. More recent methods explore training-free approaches. K-LoRA [20] adaptively selects LoRAs based on the analysis of layer-wise LoRA elements. EST-LoRAs [34] presents a training-free approach to combine style and content LoRAs driven by a matrix-energy criterion. LiONLoRA [36] introduces a parameter-efficient LoRA fusion framework for video diffusion models, using three key insights: the orthogonality of camera control LoRAs, normalization of LoRA outputs, and the integration of scaling tokens into the attention mechanism for linear control over camera movement and motion strength.

Our approach differs from this line of work in two ways. First, rather than developing algorithms to combine two LoRAs, we investigate whether datasets for each type of variation can be encoded using a shared LoRA subspace and how to compute each subspace from synthetic data. Second, we show that the resulting subspaces corresponding to different variation types are approximately orthogonal and that they can be integrated into a shared LoRA basis for efficient training. Although this shared LoRA subspace is derived from synthetic data, it generalizes well to real data.

LoRA training strategies. Prior work has proposed several LoRA training strategies that explicitly or implicitly control the low-rank subspace in which updates reside. AdaLoRA [35] introduces trainable incremental matrices with dynamic ranks and replaces computationally expensive SVD with a penalty orthogonality loss. However, they did not explore in depth whether dynamically adjusting the rank is task-dependent or influenced by the specific attributes of the task. PiSSA [19] opts to use principal singular values and vectors to initialize LoRA matrices for faster convergence, rather than the usual random initialization while keeping the original weights frozen. LoRA-GA [28] initializes the LoRA matrices by applying SVD to the gradient matrices, thus approximating the direction of fully fine-tuning. GaLore [38] projects gradients into low-rank approximations, thereby updating within a subspace for memory-efficient optimization, while effectively mimicking the trajectory of fully fine-tuning. In contrast to these methods, we introduce precomputed LoRA subspaces for efficient training, with precomputation aligned to distinct variations in 3D attributes. In particular, we show how to derive these LoRA subspaces from synthetic data.

3. Shared LoRA Subspaces

This section presents our algorithm for extracting the shared subspace from multiple LoRAs adapters, which are obtained by fine-tuning a 3D foundation model on controlled synthetic data. In this work, we focus on VGGT [26], a representative 3D foundation model, which incorporates 48 sets of self-attention and linear layers. Specifically, each self-attention layer has two matrix parameters (QKV and attention projection), and each linear layer also has two matrix parameters.

Preliminary. When applying vanilla LoRA to fine-tune a transformer-based foundation model, it enforces the displacement $dW \in \mathbb{R}^{n\times m}$ of each weight matrix W to be of low-rank $dW = AB^T$, where $A \in \mathbb{R}^{n\times r}$, $B \in \mathbb{R}^{m\times r}$, and the rank r satisfies $r \ll \min(m, n)$.

LoRA Subspace. We further introduce a specific LoRA subspace parameterization defined by a pair of matrices $\overline{A} \in \mathbb{R}^{n\times d}$ and $\overline{B} \in \mathbb{R}^{m\times d}$. When performing LoRA fine-tuning

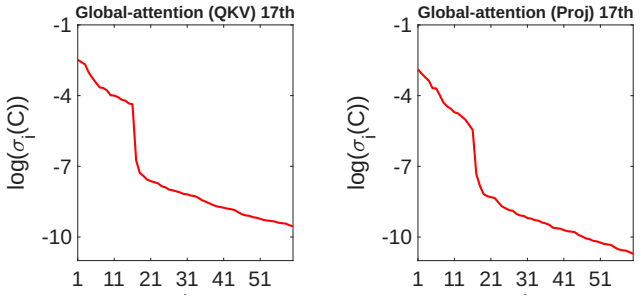

Figure 2. Singular values of C computed from QKV and projection matrices of the 17-th global self-attention layer of 10 LoRAs with respect to geometry variations. Note that the singular values are reported in log-scale.

Figure 3. Examples of synthetic datasets for each type of variations. We empirically push the variation in each type to the extreme while maintaining small variations among other types. Note that these images are very different from real-world images.

within this subspace, the weight update dW is parameterized as $dW = \overline{A}M\overline{B}^T$, where the matrix $M \in \mathbb{R}^{d\times d}$ is the only trainable parameter. This formulation reduces the total number of variables optimized during fine-tuning.

Shared LoRA Subspace. Finally, we address the problem of computing a shared LoRA subspace from an ensemble of k pairs of LoRA weight matrices $\{A_i \in R^{n\times r}\}_{1\le i\le k}$ and $\{B_i \in \mathbb{R}^{m\times r}\}_{1\le i\le k}$. Our goal is to find a pair of $A \in R^{n\times d'}$ and $B \in \mathbb{R}^{m\times d'}$ for some pre-defined d', such that AB^T optimally approximates all individual updates $A_iB_i^T$. This objective is formalized through the following optimization problem:

$$\min_{A,B} \sum_{i=1}^{k} \|AB^T - A_iB_i^T\|_{\mathcal{F}}^{\alpha}. \tag{1}$$

Here, $\|\cdot\|_{\mathcal{F}}$ denotes the matrix Frobenius norm, and the parameter α is introduced to mitigate the influence of potential LoRA outliers in $A_iB_i^T$, which may result from the construction of the datasets.

Note that Eq. (1) does not admit closed-form expressions when $\alpha \neq 2$. To solve it effectively, we employ an iteratively reweighted least squares formulation by introducing a weight in front of each term:

$$\min_{A,B} \sum_{i=1}^{k} w_i \|AB^T - A_iB_i^T\|_{\mathcal{F}}^2. \tag{2}$$

Starting from $w_i = 1, 1 \le i \le k$, we solve Eq. (1) by alternating between solving Eq. (2) with fixed w_i and fixing A and B to update w_i. When w_i are fixed, it is easy to see that Eq. (2) is equivalent to

$$\min_{A,B} \|AB^T - C\|_{\mathcal{F}}^2, \qquad C = \sum_{i=1}^{k} w_iA_iB_i^T / \sum_{i=1}^{k} w_i. \tag{3}$$

Let $C = U\Sigma V^T$ be the singular value decomposition of C where the diagonal of $\Sigma = \text{diag}(\sigma_i)$ encodes the singular values σ_i in decreasing order. It is well-known that optimal solutions of A and B are given by $A = U_{d'}\Sigma_{d'}^{\frac{1}{2}}$ and $B = V_{d'}\Sigma_{d'}^{\frac{1}{2}}$ where $\Sigma_{d'} = \text{diag}(\sigma_1, \cdots, \sigma_{d'})$ and $U_{d'}$ and $V_{d'}$ encode the corresponding singular vectors. When A and B are fixed, we update w_i as

$$w_i = 1/(\varepsilon^2 + \|AB^T - A_iB_i^T\|_{\mathcal{F}}^2)^{\frac{2-\alpha}{2}}.$$

Evidence for the Existence of Subspaces. Fig. 2 illustrates the singular values of C among the self-attention matrices of the 17-th layer of VGGT when fine-tuned on a synthetic dataset with texture variations, where $d = 16$ for each LoRA. We can see that there is indeed a significant drop between the 17-th singular value and the 16-th singular value of C, indicating the shared LoRA subspace. However, we also observed that this spectral gap varies between different layers and with respect to different variations. We will discuss such phenomena in Sec. 4.

Leveraging Subspaces for Fine-tuning. After obtaining the extracted subspaces for all attributes $\{A_iB_i^T\}_{i\in\Lambda}$, we can incorporate them after orthogonalization into our subspace LoRA fine-tuning:

$$\overline{A}M\overline{B} = (\|_{i\in\Lambda}A_i) \cdot (\|_{i\in\Lambda}\{B_i\})^T.$$

4. Subspaces of VGGT

This section details the procedure for extracting LoRA subspaces with respect to different types of variation. We begin with the generation of the controlled datasets. We then present analysis of subspaces with respect to each type of variation by applying the approach in Sec. 3. Finally, we analyze the correlations between these different subspaces.

4.1. Controlled Dataset Generation

Our datasets were constructed using MegaSynth [15], which allows control over variations in texture, geometry, camera, and lighting. To extract LoRA subspaces that correspond to different types of variation, we generated multiple specialized datasets. As shown in Fig. 3, we generate

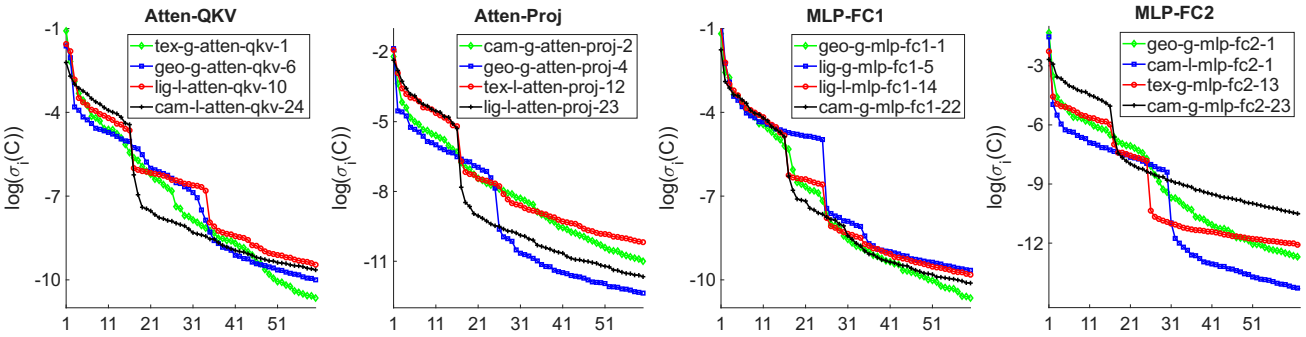

Figure 4. There are four spectral patterns of matrix C across some layers with respect to different dataset variations. They are colored in red, green, blue, and black. 'tex-g-atten-qkv-1' indicates the QKV matrix in the first global attention layer fine-tuned on texture variations. Note that we have applied a log scale to the singular values. It is clear that there is a significant drop in these curves, which indicates the existence of shared subspaces.

each dataset in each type by varying the corresponding attribute while fixing the remaining attribute types. For different datasets of the same type, the remaining fixed attributes are randomized. The motivation is that we can decompose the corresponding LoRA into two components, where the first component is shared (the attribute of interest) and the second component is random noise (remaining attributes). We can then apply the approach described in Sec. 3 to extract the first component, which is desired.

Motivated by the principles of domain randomization [25], we maximize the variations of each target type to the extreme. The core motivation here is to ensure that the variations are larger than the real-synthetic domain gap. By doing so, we aim to learn more robust subspaces that exhibit generalization capabilities to real-world data. The detailed procedures for generating these synthetic datasets are deferred to the supp. material.

4.2. Interpreting Individual LoRA Sub-Spaces

In the following, we analyze the resulting shared LoRA subspaces, focusing on their behaviors across different layers of each type of variation and their properties across different types of variation.

As illustrated in Fig. 4, there are four types of spectral behavior of the LoRA subspace C among different matrices in different layers, which are colored black, blue, red, and green. Similarly to Fig. 2, the black curves correspond to the layers in which there is a drop at r, which is the rank of individual LoRA. We observe that these layers correspond to deep attention layers in VGGT or fully connected layers. This is expected as deep attention or fully connected layers of VGGT focus on global patterns in the input of one attribute and are insensitive to differences across the inputs introduced by relatively small variations with respect to other attributes.

The blue curves correspond to the layers in which we still observe a drop in singular values, but the transition point is larger than d. Those layers are typically early layers (e.g., 4-7 in VGGT). This can be understood as the fact that these layers tend to capture more local patterns in the collection of synthetic datasets, which also include patterns that do not belong to the attribute of interest. The red curves show two transitions in singular values. They correspond to the layers between those of the blue curves and those of the black curves (e.g., 12-17 in VGGT). In contrast, the black curves show no transitions in singular values. We observe that they correspond to the first 1-3 layers in VGGT. This is expected as they record all local patterns in the synthetic datasets, whose size grows as the number of datasets increases.

We then analyze the relative spectral properties between different types of variations. Although their spectral properties are different, the matrix C is still considered low-rank as $\sigma_{2d}(C)/\sigma_{\max}(C) < 10^{-3}$. Fig. 5 plots the maximum singular values of the QKV, projection, and MLP matrices in different layers. We observe three behaviors. First, the magnitudes of the attention matrices drop for deep layers. In other words, global patterns in pre-trained models are generalizable to data variations including synthetic data. In contrast, early layers require significant adjustments in response to local patterns in synthetic datasets. Second, the magnitudes of the matrices in MLP are relatively stable. This again can be understood as the fact that they encode relational pre-trained patterns that are generalizable across real and synthetic datasets.

Moreover, the relative magnitudes between different variations change drastically, particularly among the MLP layers. This means that it is important to extract an individual subspace for each type variation, as extracting a shared subspace from all types may discard useful subspaces.

4.3. Do Subspaces Disentangle? Yes!

We proceed to study the relation between the extracted subspaces that correspond to different variations. The subspace of each variation type in each layer is encoded as $S = AB^T$. Therefore, we first introduce the distance between two subspaces S and S'. This is non-trivial in our context for two reasons. First, AB^T is invariant if we transform $A \rightarrow AX$ and $B \rightarrow BX^{-1^T}$. In other words, we cannot compare

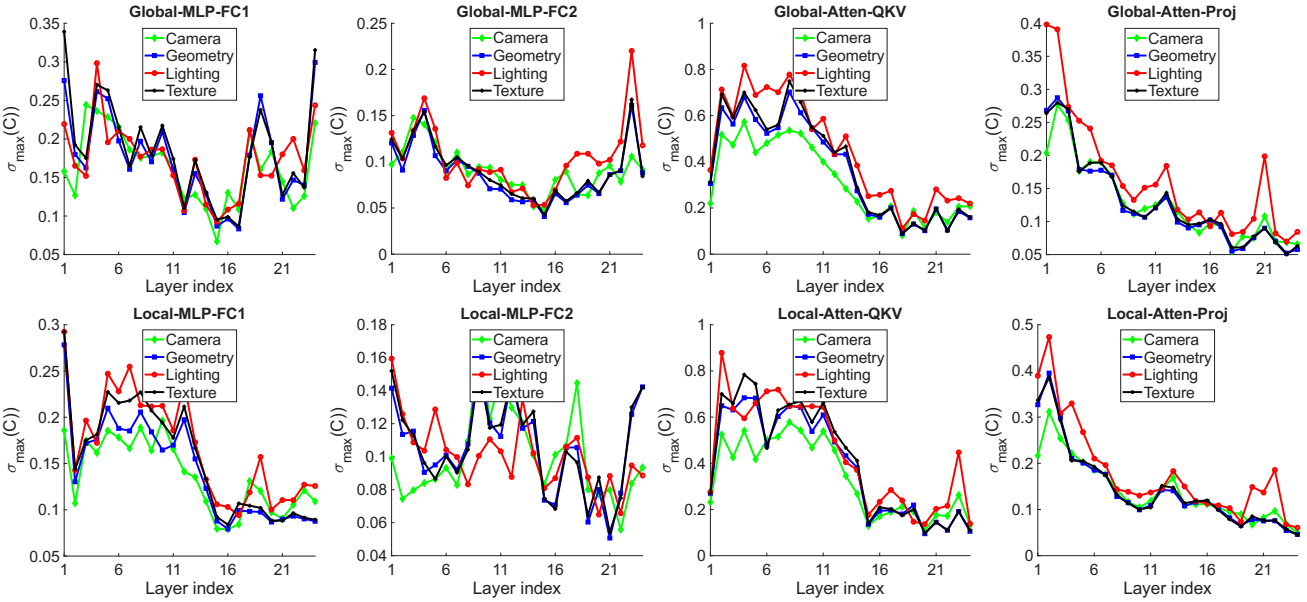

Figure 5. Maximum singular values of each shared LoRA subspace. Linear layers, QKV, and projection matrices of global self-attention and local self-attention are plotted in the top and bottom rows, respectively. We show four curves in each plot, which correspond to texture, geometry, camera, lighting variations.

A and A' or B and B' directly. We address this issue by enforcing that $A = U\Sigma^{\frac{1}{2}}$ and $B = V\Sigma^{\frac{1}{2}}$, where U, V, Σ come from SVD of $AB^T = U\Sigma V^T$. Second, $S = AB^T$ encodes the same subspace under scaling aS. Note that although scaling each column of S encodes the same linear space, it changes the absolute strength of each column, which is useful for characterizing the orthogonality between two subspaces. To address this issue, we define the angle between two matrices S and S' as

$$d(S, S') = \min_{\boldsymbol{x},\boldsymbol{x}'} \frac{\|S\boldsymbol{x} - S'\boldsymbol{x}'\|^2}{\|S\boldsymbol{x}\|^2 + \|S'\boldsymbol{x}'\|^2} = \min_{\boldsymbol{x},\boldsymbol{x}'} \frac{\|(S, -S')(\boldsymbol{x};\boldsymbol{x}')\|^2}{\|\mathrm{diag}(S, S')(\boldsymbol{x};\boldsymbol{x}')\|^2}. \quad (4)$$

It is clear that if $\boldsymbol{x}$ and $\boldsymbol{x}'$ is an optimal solution to $d(S, S')$, then $\boldsymbol{x}$ and $\frac{1}{a}\boldsymbol{S}'$ are an optimal solution to $d(S, aS')$. Therefore, $d(S, S')$ is invariant when scaling S and S'. It is easy to see that the optimal solution to Eq. (4) is given by the smallest generalized eigen-vector problem:

$$\begin{pmatrix} S^TS & -S^TS' \\ -S'^TS & S'^TS' \end{pmatrix} \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{x}' \end{pmatrix} = \lambda \begin{pmatrix} S^TS & 0 \\ 0 & S'^TS' \end{pmatrix} \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{x}' \end{pmatrix}.$$

Fig. 6 shows the eigvenvalues in Eq. 5 for typical layers between six pairs of subspaces (S, S'). In general, most of the smallest eigenvalues are above 0.5, indicating that the learned subspaces are disentangled (1 means that they are orthogonal). Moreover, both the geometry and texture subspaces show strong orthogonality with the camera subspace.

Although images and corresponding learned features shall be a complex non-linear function of attribution variations, recent results in deep learning theory, neural tangent kernels [9] and diffusion generalization [10], have shown that this non-linear function can be approximated by a linear function defined by the Jacobian of the network. Intuitively, this linear relationship promotes disentanglement while high-order residuals characterize correlations. We will leave a rigorous analysis of this matter for future work.

5. Experimental Evaluation

We adopt VGGT [26] as our base model. We first conducted 2D face anti-spoofing experiments to validate the effectiveness of the extracted subspaces. To evaluate generalization, we further conducted clothed human reconstruction experiments, demonstrating that subspaces extracted from synthetic data can generalize to real-world data. Finally, we evaluated our method on transparent object reconstruction, showing that the learned subspaces remain effective even under challenging settings with complex materials and limited training data.

Baselines. We compare our subspace-based fine-tuning strategy with full fine-tuning, and several representative PEFT methods, including LoRA [13] and PiSSA [19]. For prediction, we use the depth head instead of the point head. During fine-tuning, we freeze the DINO encoder and maintain the same number of training steps across all experiments, employing a two-stage learning rate scheduler that combines linear and cosine decay.

5.1. 2D Face Anti-Spoofing

For the subspaces, we apply the texture and geometry subspaces extracted on our created synthetic data under micro-baseline settings. Each subspace is derived from five LoRA adapters with a rank of 16.

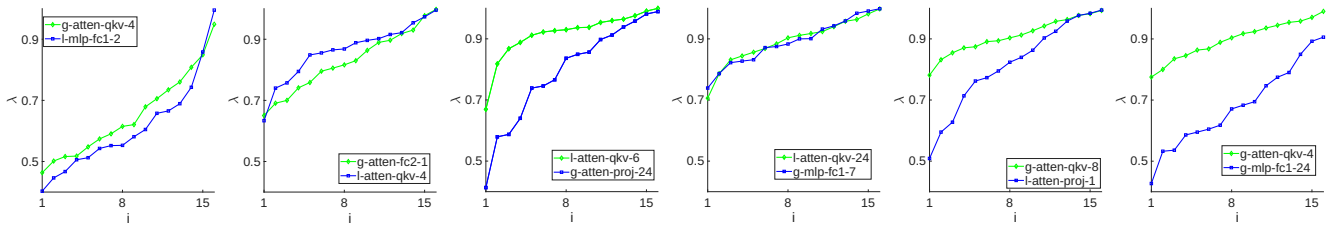

Figure 6. We show overlap ratios between six pairs of four subspaces that correspond to variations in geometry, texture, camera, and lighting. We show three representative layers of the global QKV attention. This metric, akin to reprojection error, reveals that each pair of subspaces is approximately orthogonal. Notably, the orthogonality is most significant between texture and camera, as well as between geometry and camera.

Datasets. We fine-tune on data of indoor scenes created by MegaSynth [15] and rendered under micro-baseline cameras, following the protocol in Sec. 4.1. The test set of human face consists of two parts: one part consists of face images collected from the internet, which are used to render micro-baseline videos. The other part includes real-world face data captured with an iPhone equipped with LiDAR hardware to estimate the depth of printed images [6, 7]. All test image sequences consist of 42 images, with one image selected every six frames for the synthetic evaluation, and one image selected every two frames for the real-world evaluation.

Metrics. For the synthetic face dataset, we evaluate the quality of the point cloud using the Chamfer Distance ($\times 10^{-3}$) and the normal consistency. For Chamfer Distance, we report its two directional components: *Accuracy* and *Completeness*. Additionally, we first use SAM 2 [22] to extract the face mask from the video, and the metrics are calculated based on this mask. The predicted point maps are first aligned with the ground truth using the Kabsch-Umeyama algorithm [17] for an initial Sim(3) alignment, followed by refinement using the Iterative Closest Point (ICP) algorithm [1]. For the real-world face dataset, we evaluate the quality of depth estimation using the Absolute Relative Error ($\times 10^{-2}$) and the prediction accuracy at a threshold of $\delta < 1.25$. These metrics are evaluated under joint scale and 3D translation alignment.

As presented in Table 1, our proposed subspace-based fine-tuning method significantly outperforms other fine-tuning methods on the synthetic face test set. Our method achieves comparable results on the real-world data set, while utilizing fewer trainable parameters. In contrast, LoRA lacks interpretability, and PiSSA exhibits overfitting.

Qualitative visual results of point cloud reconstruction on the synthetic test dataset are shown in Fig. 7. The original VGGT model is heavily tricked by its visual semantic priors learned on normal real-world 3D face data, leading to scattered artifacts in the reconstructed faces. All fine-tuning strategies alleviate these issues to some degree by consuming micro-baseline data for fine-tuning, while our method produces the most accurate and robust reconstructions with noticeably fewer artifacts, showing the transferability of our method to out-of-distribution data.

Method	# Trainable Param.	Synthetic Face Dataset			Real Face Dataset	
		Acc ↓	Comp ↓	NC ↑	Abs Rel ↓	$\delta < 1.25$ ↑
VGGT	-	9.006	4.965	80.74	2.651	98.59
Full	853.6 M	5.585	3.531	85.77	2.203	98.85
LoRA (rank=16)	16.3 M	5.767	3.385	84.78	**2.115**	98.92
LoRA (rank=32)	32.7 M	6.251	3.841	84.64	2.159	98.93
LoRA (rank=64)	65.3 M	6.393	3.971	84.59	2.157	<u>98.94</u>
LoRA (rank=128)	130.7 M	6.590	4.242	84.92	2.162	**98.95**
PiSSA (rank=16)	16.3 M	5.729	3.532	85.30	2.433	98.81
PiSSA (rank=32)	32.7 M	6.488	4.198	84.64	3.185	98.41
PiSSA (rank=64)	65.3 M	6.526	4.407	84.84	3.106	98.60
PiSSA (rank=128)	130.7 M	6.890	4.706	84.77	4.020	98.32
Ours (d=8)	3.8 M	5.921	**1.966**	76.77	2.774	98.26
Ours (d=16)	4.0 M	**3.831**	<u>2.037</u>	**86.65**	2.170	98.92
Ours (d=32)	4.7 M	<u>4.287</u>	2.395	<u>86.43</u>	<u>2.151</u>	<u>98.94</u>

Table 1. **Evaluation on Synthetic and Real-World Human Face Datasets under Micro-Baseline Settings.** The synthetic dataset contains 50 face images, which are used to render videos from nine different viewpoints. The real-world dataset consists of 50 printed images, each captured in one long-burst bundle. **Bold**: best; <u>underline</u>: second best.

5.2. Clothed Human Reconstruction

Datasets. Following the approach in HART [5], we select 2,345 human scans from the THuman 2.1 dataset [31] as our fine-tuning dataset. The subjects are rendered from 96 distinct viewpoints along a 360-degree azimuthal trajectory. We used two datasets for testing. One is the THuman 2.1 test split, which contains 100 subjects for in-domain evaluation. The second is the test set from the 2K2K dataset [12], which is used for cross-domain evaluation and offers greater age diversity. All comparisons with baselines are conducted using a fixed setting of 8 input views. We apply all four extracted subspaces derived in the object-centering settings. Each subspace is computed from a bundle of ten LoRA adapters, each having a rank of $r = 16$.

The point cloud reconstruction evaluation results are shown in Tab. 2. We observe that all fine-tuning methods can, under certain configurations, degrade the performance of the original base model. For LoRA, a low rank (r) is insufficient to capture the new variations present in the target

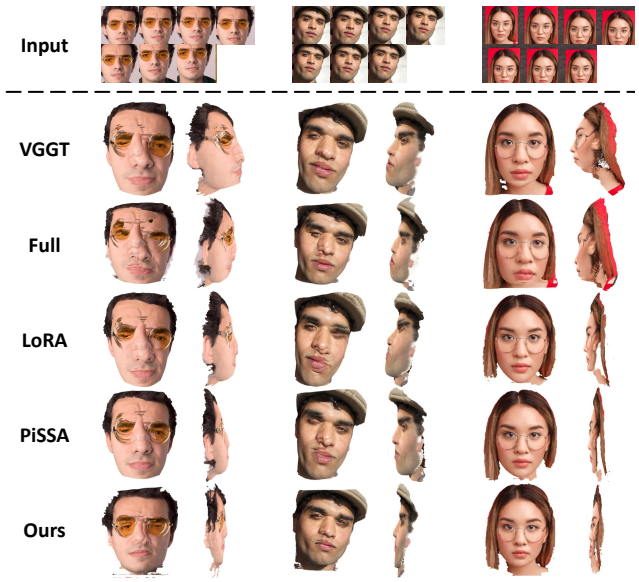

Figure 7. **Qualitative Comparison of 2D Face Anti-Spoofing Tasks.** Compared to other fine-tune strategies, our method produces more accurate and robust reconstruction with fewer artifacts.

Method	# Trainable Param.	**THuman** (In-domain)			**2K2K** (Cross-domain)		
		Acc ↓	Comp ↓	NC ↑	Acc ↓	Comp ↓	NC ↑
VGGT	-	2.816	1.911	91.51	3.103	2.122	92.81
Full	853.6 M	3.053	1.932	91.17	3.655	2.213	92.25
LoRA (rank=16)	16.3 M	3.195	2.089	91.63	2.717	1.829	92.73
LoRA (rank=32)	32.7 M	2.849	1.922	91.85	2.633	1.773	93.00
LoRA (rank=64)	65.3 M	2.791	1.902	92.12	3.017	1.968	93.18
LoRA (rank=128)	130.7 M	3.188	2.507	**92.48**	2.986	1.959	93.86
LoRA (rank=256)	261.4 M	3.521	4.978	91.81	2.517	1.769	**93.99**
PiSSA (rank=16)	16.3 M	3.009	1.921	90.81	2.791	1.866	92.92
PiSSA (rank=32)	32.7 M	3.228	2.028	90.59	2.991	1.972	92.59
PiSSA (rank=64)	65.3 M	3.931	2.351	89.42	3.730	2.328	91.13
PiSSA (rank=128)	130.7 M	4.052	2.416	90.10	3.745	2.250	91.52
PiSSA (rank=256)	261.4 M	4.292	3.689	90.19	4.122	2.288	91.38
Ours (d=16)	4.7 M	3.392	2.138	90.91	3.019	1.887	91.70
Ours (d=32)	7.6 M	3.332	2.220	91.56	2.825	1.878	93.00
Ours (d=64)	19.3 M	**2.745**	**1.882**	91.82	**2.513**	**1.754**	93.56

Table 2. **Evaluation of Clothed Human Reconstruction: Point Map Estimation on THuman 2.1 [31] and 2K2K [12].** Our method achieves the best performance across nearly all metrics.

data. In contrast, when the rank is increased, LoRA tends to overfit the training data. PiSSA, which initializes LoRA using principal components derived from SVD, also shows a degradation as the rank increases.

Our proposed method, while achieving sub-optimal performance when the subspace dimension d is small, shows a significant trend: as d increases, the extracted subspace becomes increasingly robust. This robustness stabilizes the fine-tuning process and leads to superior generalization performance, ultimately achieving superior results across nearly all metrics with fewer parameters.

Fig. 8 shows qualitative results. The VGGT base model produces noticeable artifacts in the presence of new variations, stemming from incorrect visual matching, particularly along object edges. After fine-tuning, the model's ability to handle these problems improves, but noise remains significant in detailed regions, such as the hands. On the 2K2K dataset, our approach shows the best generalization over all other methods.

5.3. Transparent Object Reconstruction

We also evaluate on ClearPose [8], a challenging real-world dataset designed for transparent object reconstruction. ClearPose comprises 51 real-world scenes, captured using Intel RealSense L515. They include 63 transparent objects (e.g., bottles and cups). We select 32 scenes for training and use the remaining scenes as the test set. The results, with both *Accuracy* and *Completeness* reported in units of 10^{-2}, are presented in Table 3. We can see that with a similar number of trainable parameters, i.e., 16.3M, our approach outperforms all baselines. Note that this is achieved without introducing any sampling of transparent textures when learning subspaces. To achieve similar performance, LoRA and AdaLoRA [35] require a much more number of parameters.

Method	# Trainable Param.	**In-domain**		
		Acc ↓	Comp ↓	NC ↑
VGGT	-	3.123	3.271	67.77
Fully fine-tuned	853.6 M	1.653	2.559	74.18
LoRA (rank=16)	16.3 M	1.808	2.522	73.74
LoRA (rank=32)	32.7 M	1.811	2.562	73.76
LoRA (rank=64)	65.3 M	1.787	2.580	73.81
LoRA (rank=128)	130.7 M	**1.753**	2.571	73.80
AdaLoRA (rank=16)	15.0 M	1.859	2.479	72.67
AdaLoRA (rank=32)	29.9 M	1.832	2.464	72.89
AdaLoRA (rank=64)	59.7 M	1.826	**2.453**	73.07
AdaLoRA (rank=128)	119.4 M	1.836	2.465	73.20
Ours	16.3 M	1.764	2.462	**73.86**

Table 3. **Evaluation of Transparent Object Reconstruction: Point Map Estimation on ClearPose [8].** Our method achieves comparable performance across all metrics.

5.4. Ablation Study

Effectiveness of extracted subspaces. In the first experiment, We can replace $\overline{A}$ and $\overline{B}$ in the LoRA parametrization $dW = \overline{A}M\overline{B}$ with the principal singular vectors of the original model's weights. As shown in Tab. 4, our approach shows a clear advantage over this alternative when using the same rank.

Rank of LoRA Pairs. In this experiment, we fix the dimension d of the shared sub-space while changing the rank r of the input LoRA pairs. As shown in Tab. 5, the performance trends of the two subspaces are generally similar when varying d with fixed r are similar. The difference may stem from the number of LoRA pairs and the quality of the dataset used to extract the subspaces. This also highlights the importance of increasing the variance and diversity of synthetic datasets.

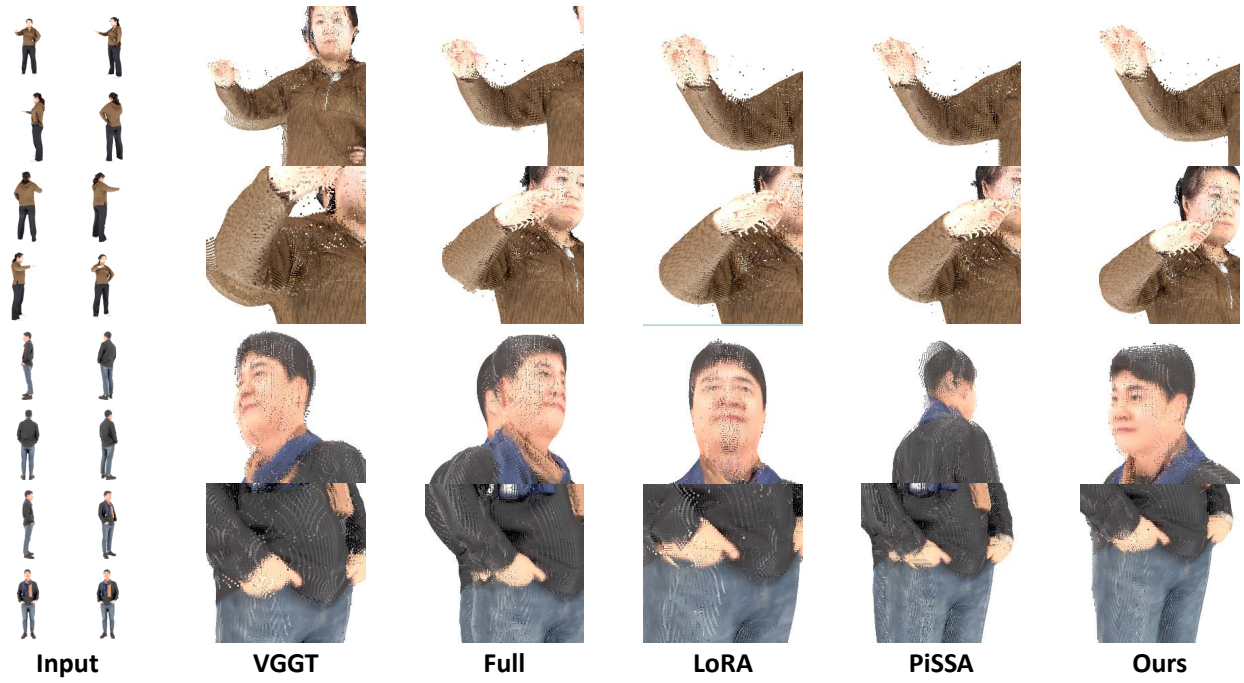

Figure 8. **Visual Results of Clothed Human Reconstruction Tasks.** Each input consists of eight different viewpoints. The first row is selected from the THuman 2.1 test split, while the second row is chosen from the 2K2K test set. Our model produces fewer artifacts on the object, but its performance on more detailed regions, such as the hands, is less ideal.

Method	**THuman** (In-domain)			**2K2K** (Cross-domain)		
	Acc ↓	Comp ↓	NC ↑	Acc ↓	Comp ↓	NC ↑
PSV (d=32)	5.605	3.086	89.58	6.844	4.523	89.27
PSV (d=64)	4.066	2.423	90.65	3.805	2.202	90.84
PSV (d=128)	3.709	2.394	91.10	3.290	2.128	92.50
PSV (d=256)	2.785	1.904	91.76	2.542	1.762	93.42
Ours (d=8)	5.839	3.363	89.33	5.712	2.602	89.98
Ours (d=16)	3.392	2.138	90.91	3.019	1.887	91.70
Ours (d=32)	3.332	2.220	91.56	2.825	1.878	93.00
Ours (d=64)	2.745	1.882	91.82	2.513	1.754	93.56

Table 4. Computing $\overline{A}$ and $\overline{B}$ using our approach is superior to using the leading singular vectors of the original weight matrix.

Method	**THuman** (In-domain)			**2K2K** (Cross-domain)		
	Acc ↓	Comp ↓	NC ↑	Acc ↓	Comp ↓	NC ↑
r=64 (d=8)	6.621	3.092	88.76	6.070	3.299	88.94
r=64 (d=16)	4.030	2.343	89.94	4.813	3.031	89.93
r=64 (d=32)	3.797	2.482	91.31	3.246	2.094	92.43
r=64 (d=64)	2.783	1.871	91.41	2.563	1.756	93.19
r=16 (d=8)	5.839	3.363	89.33	5.712	2.602	89.98
r=16 (d=16)	3.392	2.138	90.91	3.019	1.887	91.70
r=16 (d=32)	3.332	2.220	91.56	2.825	1.878	93.00
r=16 (d=64)	2.745	1.882	91.82	2.513	1.754	93.56

Table 5. **Comparison of Subspaces Extracted from LoRAs with Different Ranks.** Ablation study to validate our shared space hypothesis.

6. Conclusions and Future Work

In this paper, we introduce the problem of extracting subspaces of a transformer-based 3D foundation model for LoRA-based fine-tuning. We show that such subspaces that correspond to variations in geometry, texture, camera motion, and lighting do exist, and they are approximately disentangled. We present an algorithm that computes them from synthetic datasets generated in a controlled manner. A striking message is that these subspaces lead to efficient fine-tuning procedures for downstream tasks, achieving better predictive accuracy than state-of-the-art approaches.

We hope that our work inspires exploration into the understanding of 3D foundation models. There are ample opportunities for future research. First of all, we study only static scenes. An obvious extension is to add motion variations to understand recent 4D foundation models. Another direction is to study common and differences across different 3D foundation models. Finally, this paper focuses on the use of synthetic data for fine-tuning, and it is interesting to study how to combine large-scale synthetic and small-scale real datasets to enhance fine-tuning performance.

Acknowledgments. This project was supported by NSF-2047677, 2413161, 2504906, 2515626, GIFTs from Adobe and Google, and computing support on the Vista GPU Cluster through the Center for Generative AI (CGAI) and TACC at UT Austin.

References

[1] Paul J Besl and Neil D McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992. 6

[2] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ B. Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen Creel, Jared Quincy Davis, Dorottya Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren E. Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, and et al. On the opportunities and risks of foundation models. *CoRR*, abs/2108.07258, 2021. 1

[3] Yohann Cabon, Lucas Stoffl, Leonid Antsfeld, Gabriela Csurka, Boris Chidlovskii, Jérôme Revaud, and Vincent Leroy. Must3r: Multi-view network for stereo 3d reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 1050–1060. Computer Vision Foundation / IEEE, 2025. 2

[4] Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. Ttt3r: 3d reconstruction as test-time training, 2025. 2

[5] Xiyi Chen, Shaofei Wang, Marko Mihajlovic, Taewon Kang, Sergey Prokudin, and Ming Lin. Hart: Human aligned reconstruction transformer. *arXiv preprint arXiv:2509.26621*, 2025. 2, 6

[6] Ilya Chugunov, Yuxuan Zhang, Zhihao Xia, Xuaner Zhang, Jiawen Chen, and Felix Heide. The implicit values of a good hand shake: Handheld multi-frame neural depth refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2852–2862, 2022. 6

[7] Ilya Chugunov, Yuxuan Zhang, and Felix Heide. Shakes on a plane: Unsupervised depth estimation from unstabilized photography. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13240–13251, 2023. 6

[8] Chen et al. Clearpose: Large-scale transparent object dataset and benchmark. In *ECCV*, 2022. 7

[9] Du et al. Gradient descent finds global minima of deep neural networks. In *ICML*, 2019. 5

[10] Kadkhodaie et al. Generalization in diffusion models arises from geometry-adaptive harmonic representations. In *ICLR*, 2024. 5

[11] Yarden Frenkel, Yael Vinker, Ariel Shamir, and Daniel Cohen-Or. Implicit style-content separation using b-lora. In *European Conference on Computer Vision*, pages 181–198. Springer, 2024. 2

[12] 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. 6, 7

[13] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. 1, 2, 5

[14] Hanwen Jiang, Hao Tan, Peng Wang, Haian Jin, Yue Zhao, Sai Bi, Kai Zhang, Fujun Luan, Kalyan Sunkavalli, Qixing Huang, et al. Rayzer: A self-supervised large view synthesis model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4918–4929, 2025. 2

[15] 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. 3, 6

[16] Haian Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snavely, and Zexiang Xu. Lvsm: A large view synthesis model with minimal 3d inductive bias. *arXiv preprint arXiv:2410.17242*, 2024. 2

[17] James F. Lawrence, Javier Bernal, and Christoph Witzgall. A purely algebraic justification of the kabsch-umeyama algorithm. *Journal of Research of the National Institute of Standards and Technology*, 124(124028), 2019. 6

[18] Ziqi Lu, Heng Yang, Danfei Xu, Boyi Li, Boris Ivanovic, Marco Pavone, and Yue Wang. Lora3d: Low-rank self-calibration of 3d geometric foundation models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. 2

[19] Fanxu Meng, Zhaohui Wang, and Muhan Zhang. Pissa: Principal singular values and singular vectors adaptation of large language models. *Advances in Neural Information Processing Systems*, 37:121038–121072, 2024. 2, 5

[20] Ziheng Ouyang, Zhen Li, and Qibin Hou. K-lora: Unlocking training-free fusion of any subject and style loras. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 13041–13050. Computer Vision Foundation / IEEE, 2025. 2

[21] Quanhao Qian, Guoyang Zhao, Gongjie Zhang, Jiuniu Wang, Ran Xu, Junlong Gao, and Deli Zhao. Gp3: A 3d geometry-aware policy with multi-view images for robotic manipulation, 2025. 2

[22] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 6

[23] Viraj Shah, Nataniel Ruiz, Forrester Cole, Erika Lu, Svetlana Lazebnik, Yuanzhen Li, and Varun Jampani. Ziplora:

Any subject in any style by effectively merging loras. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part I*, pages 422–438. Springer, 2024. 2

[24] Donald Shenaj, Ondrej Bohdal, Mete Ozay, Pietro Zanuttigh, and Umberto Michieli. Lora. rar: Learning to merge loras via hypernetworks for subject-style conditioned image generation. *arXiv preprint arXiv:2412.05148*, 2024. 2

[25] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2017, Vancouver, BC, Canada, September 24-28, 2017*, pages 23–30. IEEE, 2017. 4

[26] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotný. VGGT: visual geometry grounded transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 5294–5306. Computer Vision Foundation / IEEE, 2025. 1, 2, 5

[27] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jérôme Revaud. Dust3r: Geometric 3d vision made easy. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 20697–20709. IEEE, 2024. 2

[28] Shaowen Wang, Linxi Yu, and Jian Li. Lora-ga: Low-rank adaptation with gradient approximation. *Advances in Neural Information Processing Systems*, 37:54905–54931, 2024. 2

[29] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning, 2025. 2

[30] Chengtang Yao, Zhidan Liu, Jiaxi Zeng, Lidong Yu, Yuwei Wu, and Yunde Jia. 3d visual illusion depth estimation. *arXiv preprint arXiv:2505.13061*, 2025. 2

[31] 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. 6, 7

[32] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. 2

[33] Jiahui Zhang, Yuelei Li, Anpei Chen, Muyu Xu, Kunhao Liu, Jianyuan Wang, Xiao-Xiao Long, Hanxue Liang, Zexiang Xu, Hao Su, Christian Theobalt, Christian Rupprecht, Andrea Vedaldi, Kaichen Zhou, Paul Pu Liang, Shijian Lu, and Fangneng Zhan. Advances in feed-forward 3d reconstruction and view synthesis: A survey, 2025. 2

[34] Jia-Chen Zhang and Yu-Jie Xiong. Subject or style: Adaptive and training-free mixture of loras, 2025. 2

[35] Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *ICLR*, 2023. 2, 7

[36] Yisu Zhang, Chenjie Cao, Chaohui Yu, and Jianke Zhu. Lion-lora: Rethinking lora fusion to unify controllable spatial and temporal generation for video diffusion. *International Conference on Computer Vision (ICCV)*, 2025. 2

[37] Hangtian Zhao, Xiang Chen, Yizhe Li, Qianhao Wang, Haibo Lu, and Fei Gao. Fastvidar: Real-time omnidirectional depth estimation via alternative hierarchical attention. *arXiv preprint arXiv:2509.23733*, 2025. 2

[38] Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. Galore: Memory-efficient llm training by gradient low-rank projection. *arXiv preprint arXiv:2403.03507*, 2024. 2

[39] Qitao Zhao, Hao Tan, Qianqian Wang, Sai Bi, Kai Zhang, Kalyan Sunkavalli, Shubham Tulsiani, and Hanwen Jiang. E-rayzer: Self-supervised 3d reconstruction as spatial visual pre-training. *arXiv preprint arXiv:2512.10950*, 2025. 2

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.

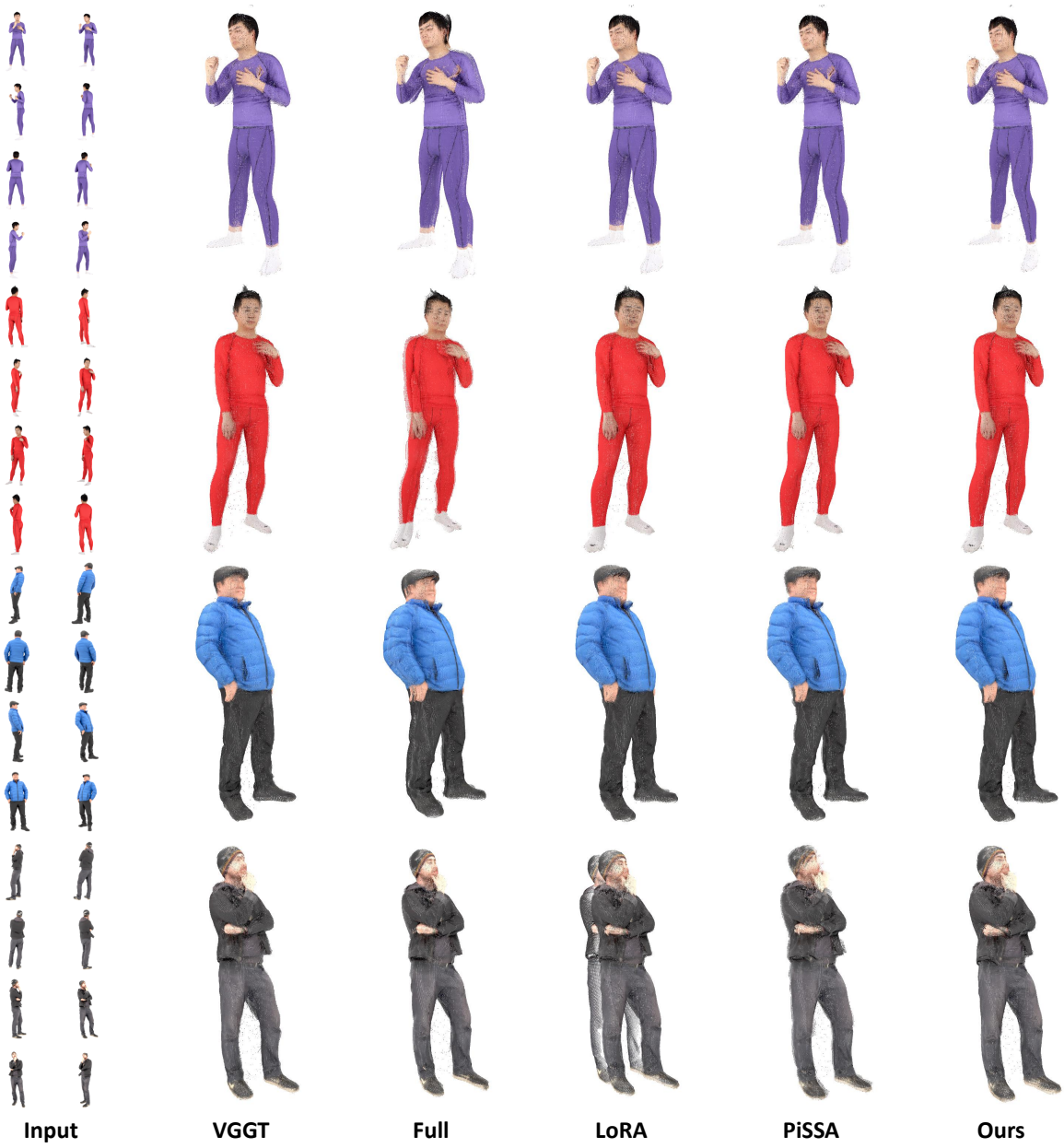

Figure 2. More visual comparison of Clothed Human Reconstruction Tasks.

4.1. Texture Subspace

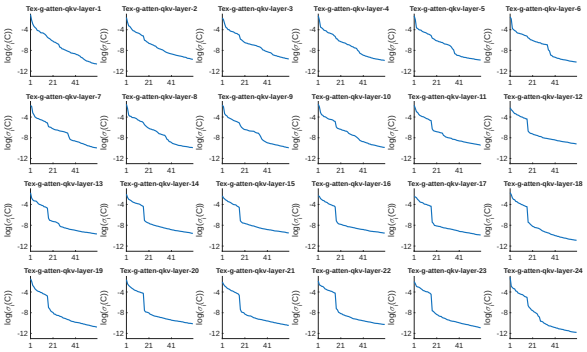

Figure 3. Singular values of the QKV matrix in the global attention layer with respect to texture variations.

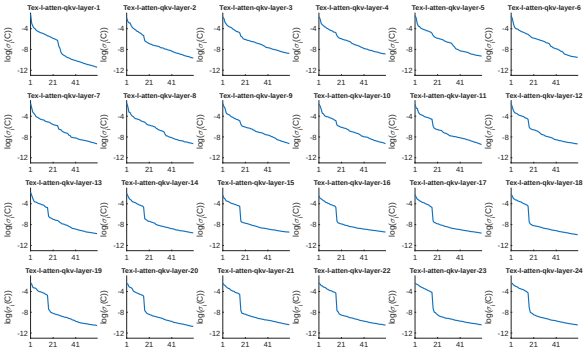

Figure 4. Singular values of the QKV matrix in the frame attention layer with respect to texture variations.

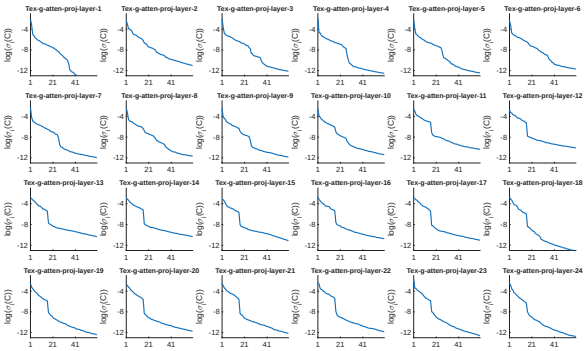

Figure 5. Singular values of the projection matrix in the global attention layer with respect to texture variations.

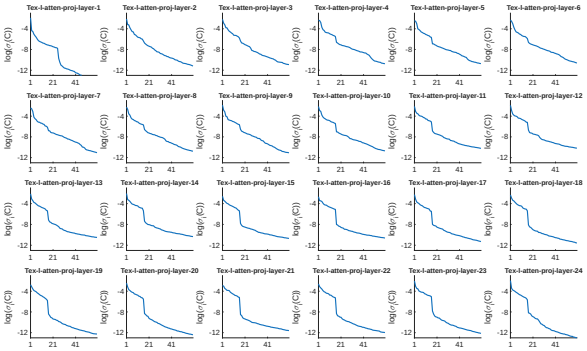

Figure 6. Singular values of the projection matrix in the frame attention layer with respect to texture variations.

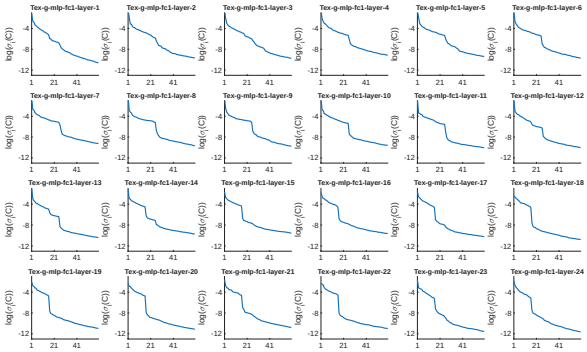

Figure 7. Singular values of the first fully connected matrix in the global attention layer with respect to texture variations.

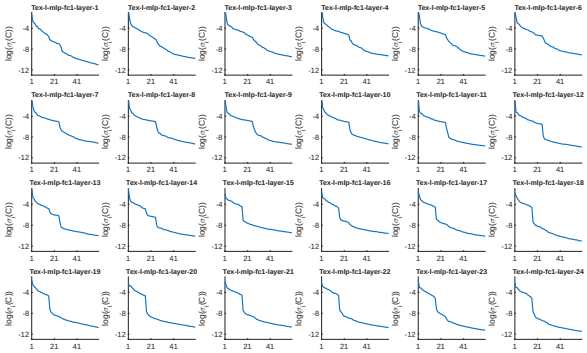

Figure 8. Singular values of the first fully connected matrix in the frame attention layer with respect to texture variations.

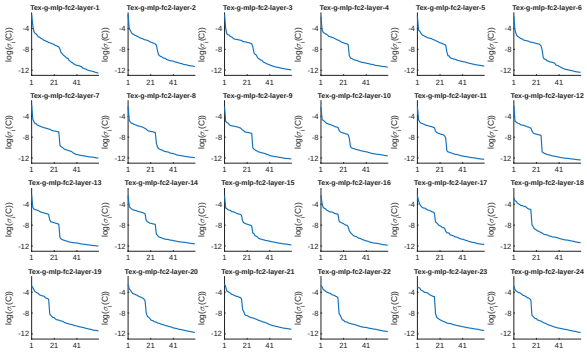

Figure 9. Singular values of the second fully connected matrix in the global attention layer with respect to texture variations.

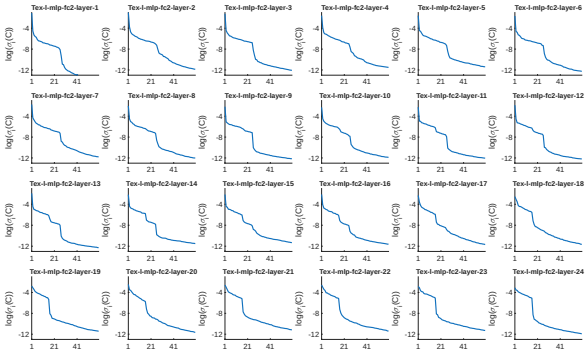

Figure 10. Singular values of the second fully connected matrix in the frame attention layer with respect to texture variations.

4.2. Geometry Subspace

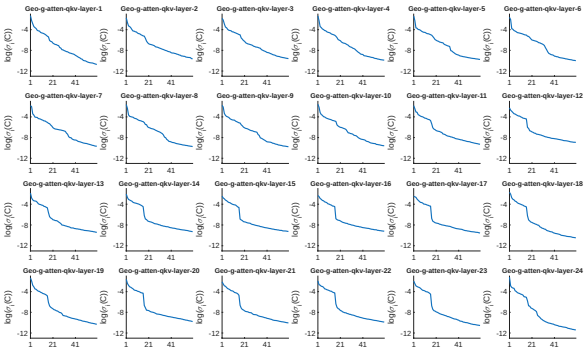

Figure 11. Singular values of the QKV matrix in the global attention layer with respect to geometry variations.

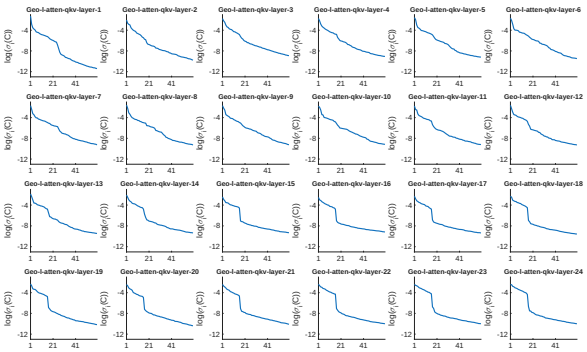

Figure 12. Singular values of the QKV matrix in the frame attention layer with respect to geometry variations.

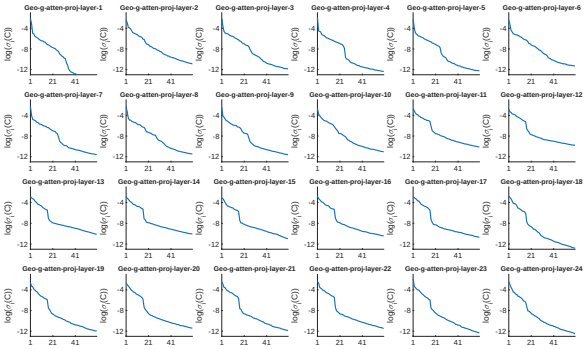

Figure 13. Singular values of the projection matrix in the global attention layer with respect to geometry variations.

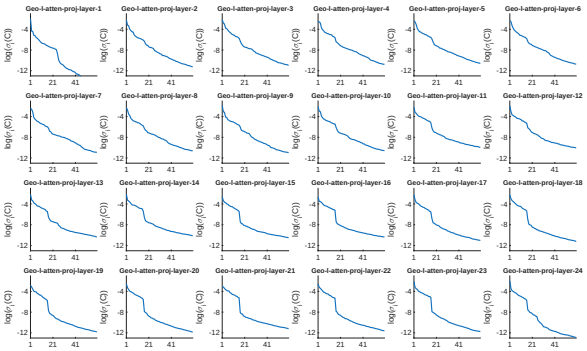

Figure 14. Singular values of the projection matrix in the frame attention layer with respect to geometry variations.

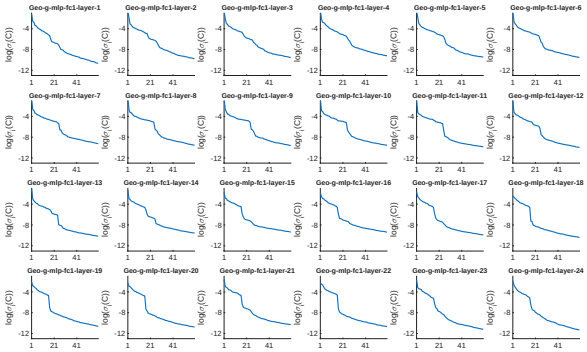

Figure 15. Singular values of the first fully connected matrix in the global attention layer with respect to geometry variations.

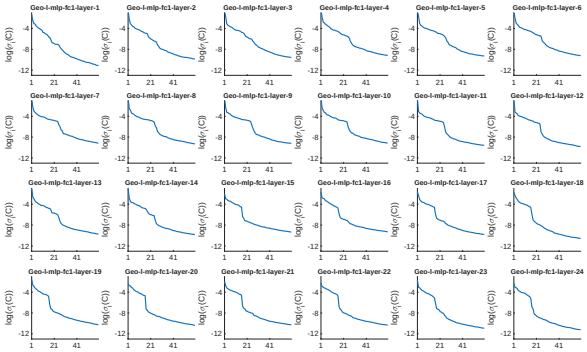

Figure 16. Singular values of the first fully connected matrix in the frame attention layer with respect to geometry variations.

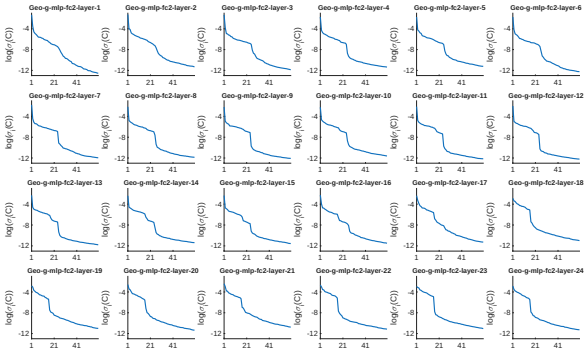

Figure 17. Singular values of the second fully connected matrix in the global attention layer with respect to geometry variations.

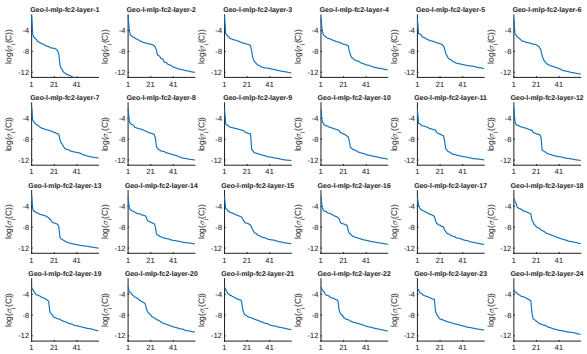

Figure 18. Singular values of the second fully connected matrix in the frame attention layer with respect to geometry variations.

4.3. Camera Motion Subspace

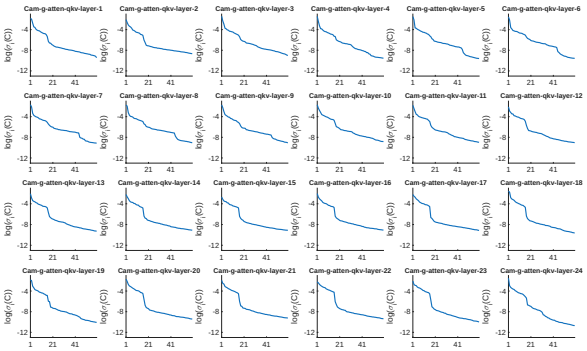

Figure 19. Singular values of the QKV matrix in the global attention layer with respect to camera variations.

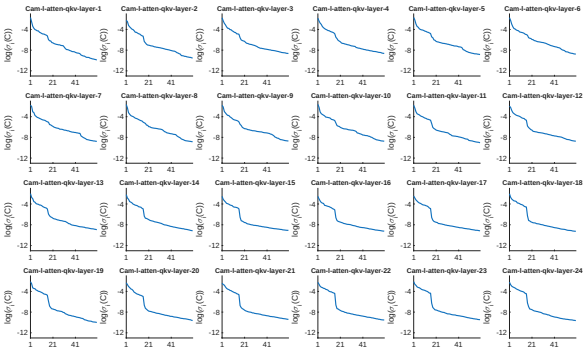

Figure 20. Singular values of the QKV matrix in the frame attention layer with respect to camera variations.

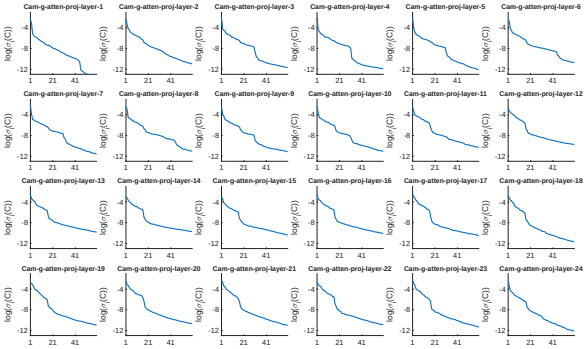

Figure 21. Singular values of the projection matrix in the global attention layer with respect to camera variations.

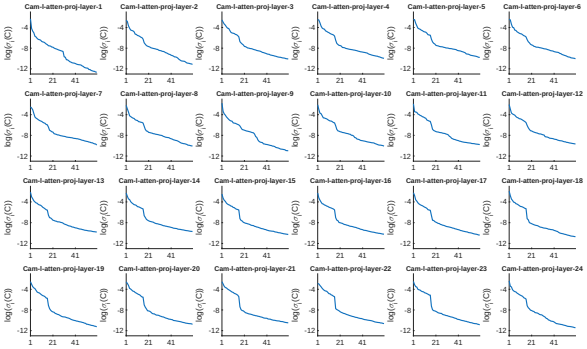

Figure 22. Singular values of the projection matrix in the frame attention layer with respect to camera variations.

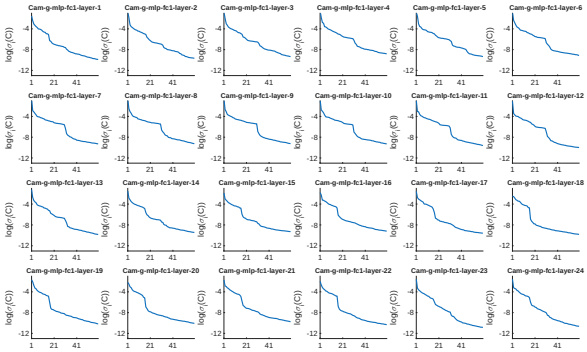

Figure 23. Singular values of the first fully connected matrix in the global attention layer with respect to camera variations.

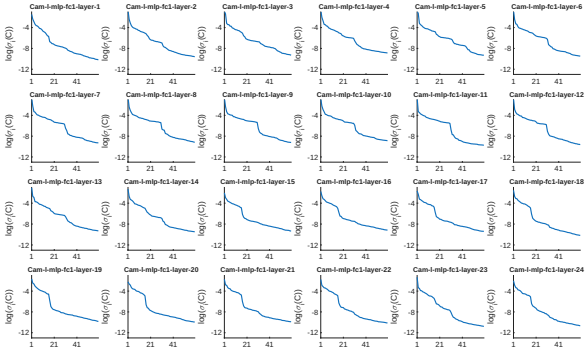

Figure 24. Singular values of the first fully connected matrix in the frame attention layer with respect to camera variations.

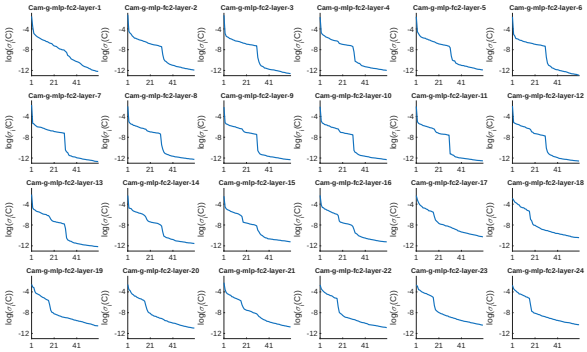

Figure 25. Singular values of the second fully connected matrix in the global attention layer with respect to camera variations.

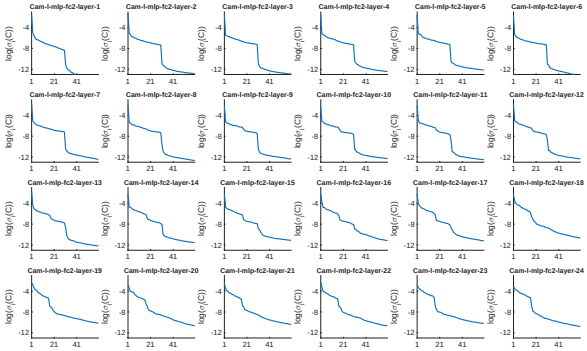

Figure 26. Singular values of the second fully connected matrix in the frame attention layer with respect to camera variations.

4.4. Lighting Subspace

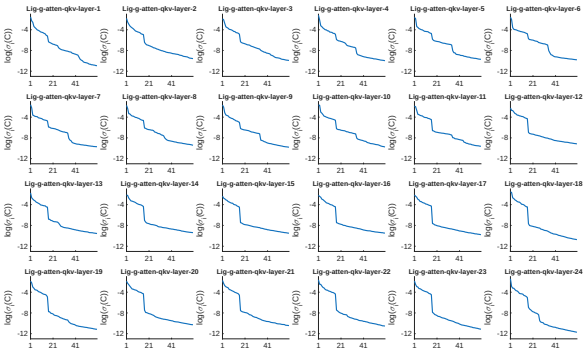

Figure 27. Singular values of the QKV matrix in the global attention layer with respect to lighting variations.

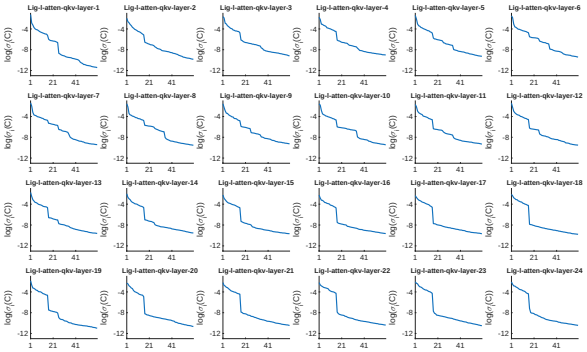

Figure 28. Singular values of the QKV matrix in the frame attention layer with respect to lighting variations.

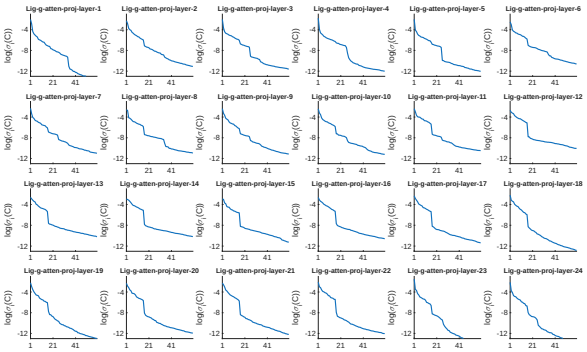

Figure 29. Singular values of the projection matrix in the global attention layer with respect to lighting variations.

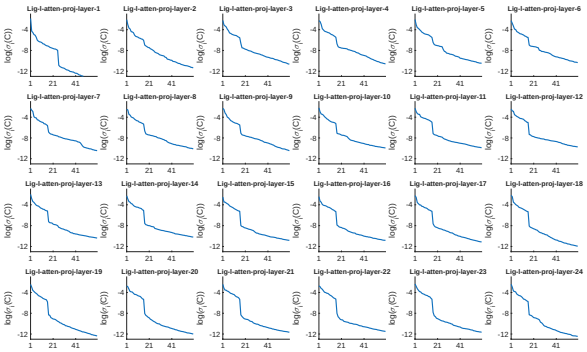

Figure 30. Singular values of the projection matrix in the frame attention layer with respect to lighting variations.

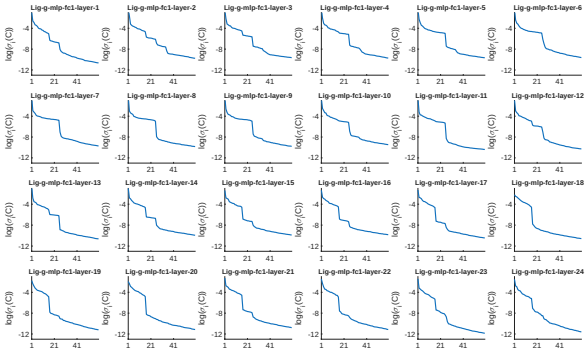

Figure 31. Singular values of the first fully connected matrix in the global attention layer with respect to lighting variations.

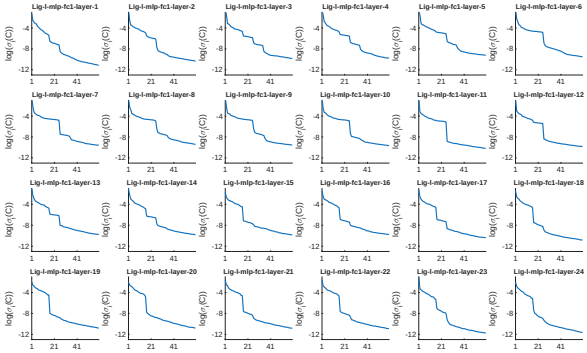

Figure 32. Singular values of the first fully connected matrix in the frame attention layer with respect to lighting variations.

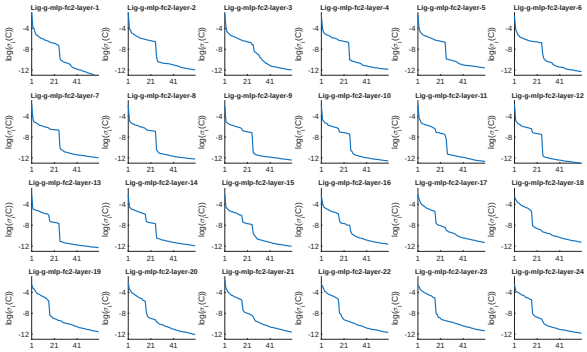

Figure 33. Singular values of the second fully connected matrix in the global attention layer with respect to lighting variations.

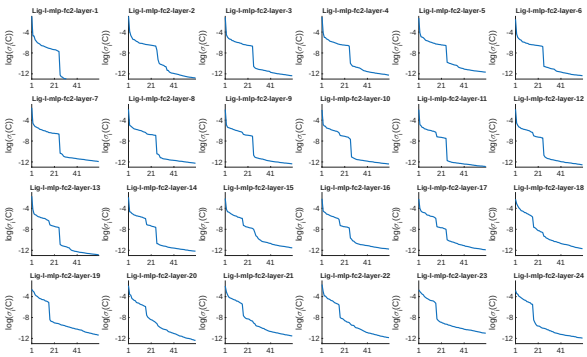

Figure 34. Singular values of the second fully connected matrix in the frame attention layer with respect to lighting variations.

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

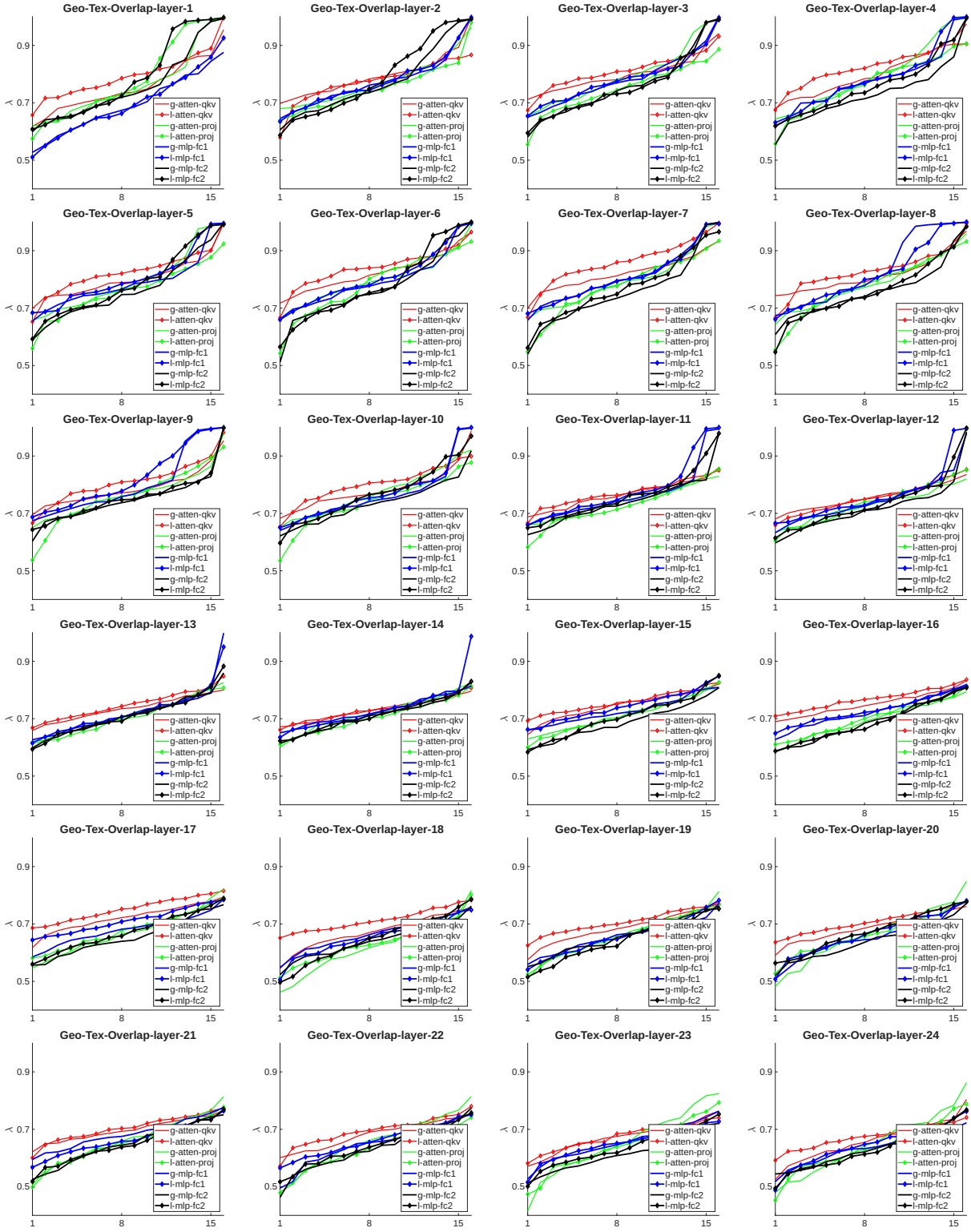

Figure 35. The overlap ratio between subspaces that correspond to variations in geometry and texture.

5.2. Geometry vs Camera

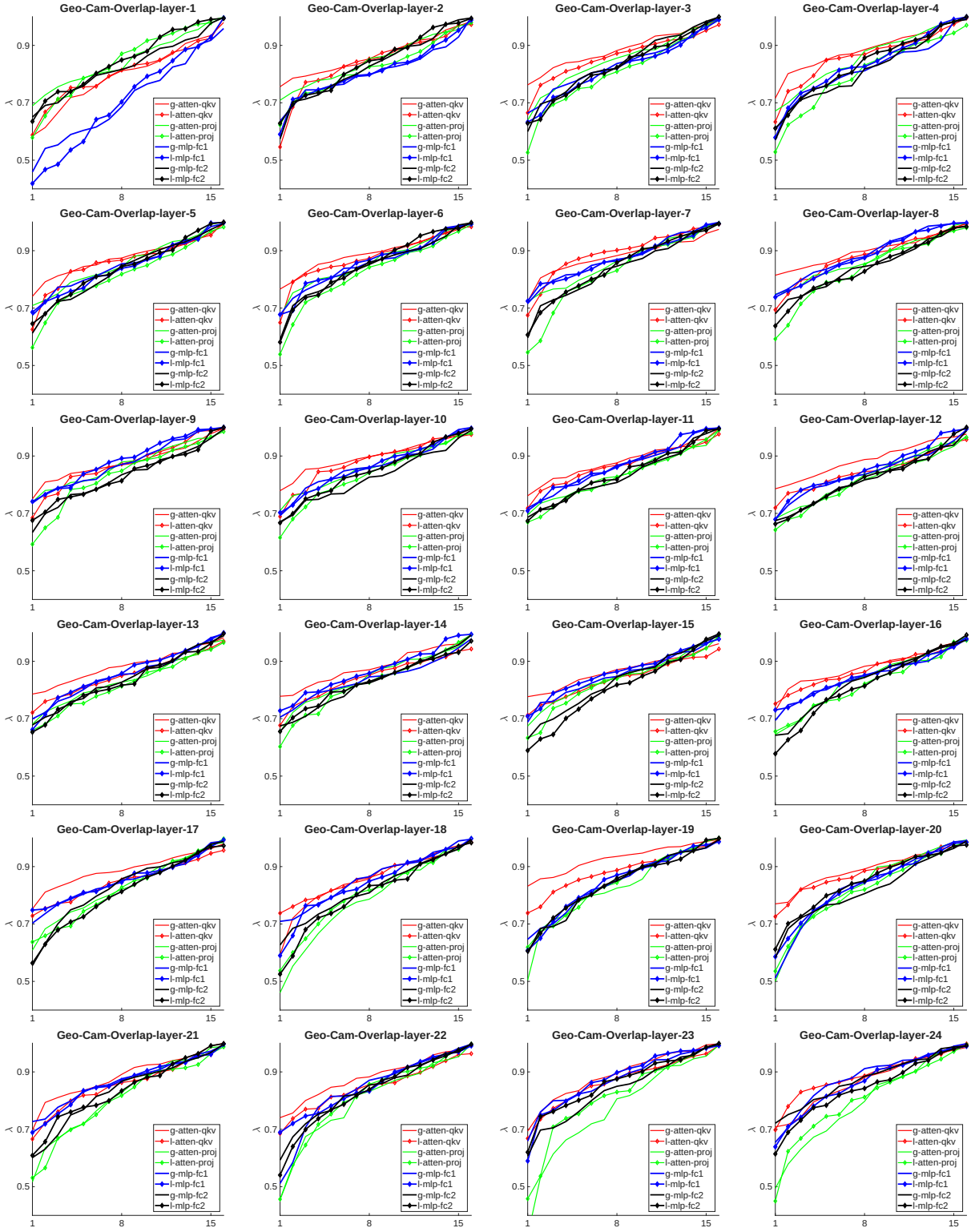

Figure 36. The overlap ratio between subspaces that correspond to variations in geometry and camera motion.

5.3. Geometry vs Lighting

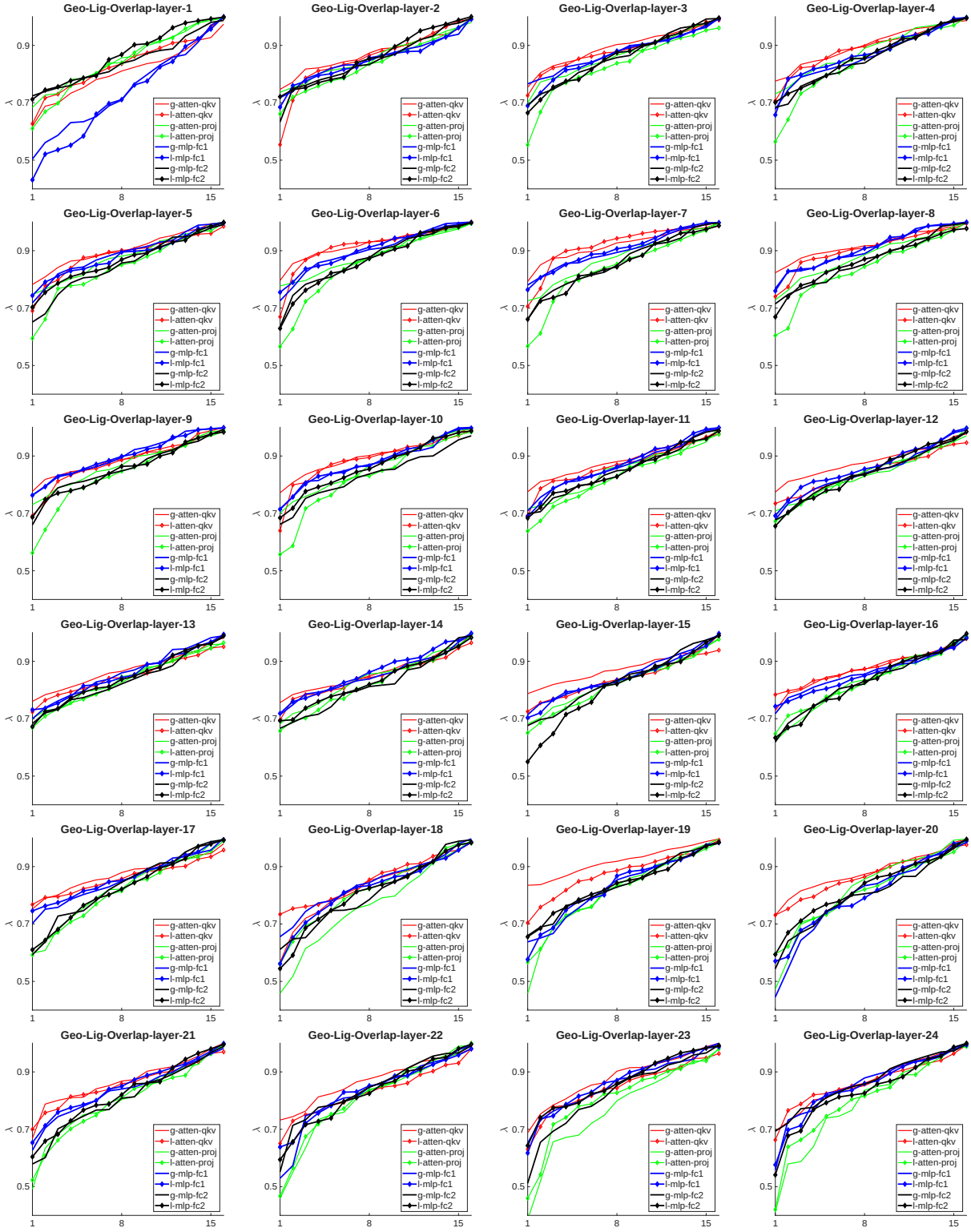

Figure 37. The overlap ratio between subspaces that correspond to variations in geometry and lighting.

5.4. Texture vs Camera

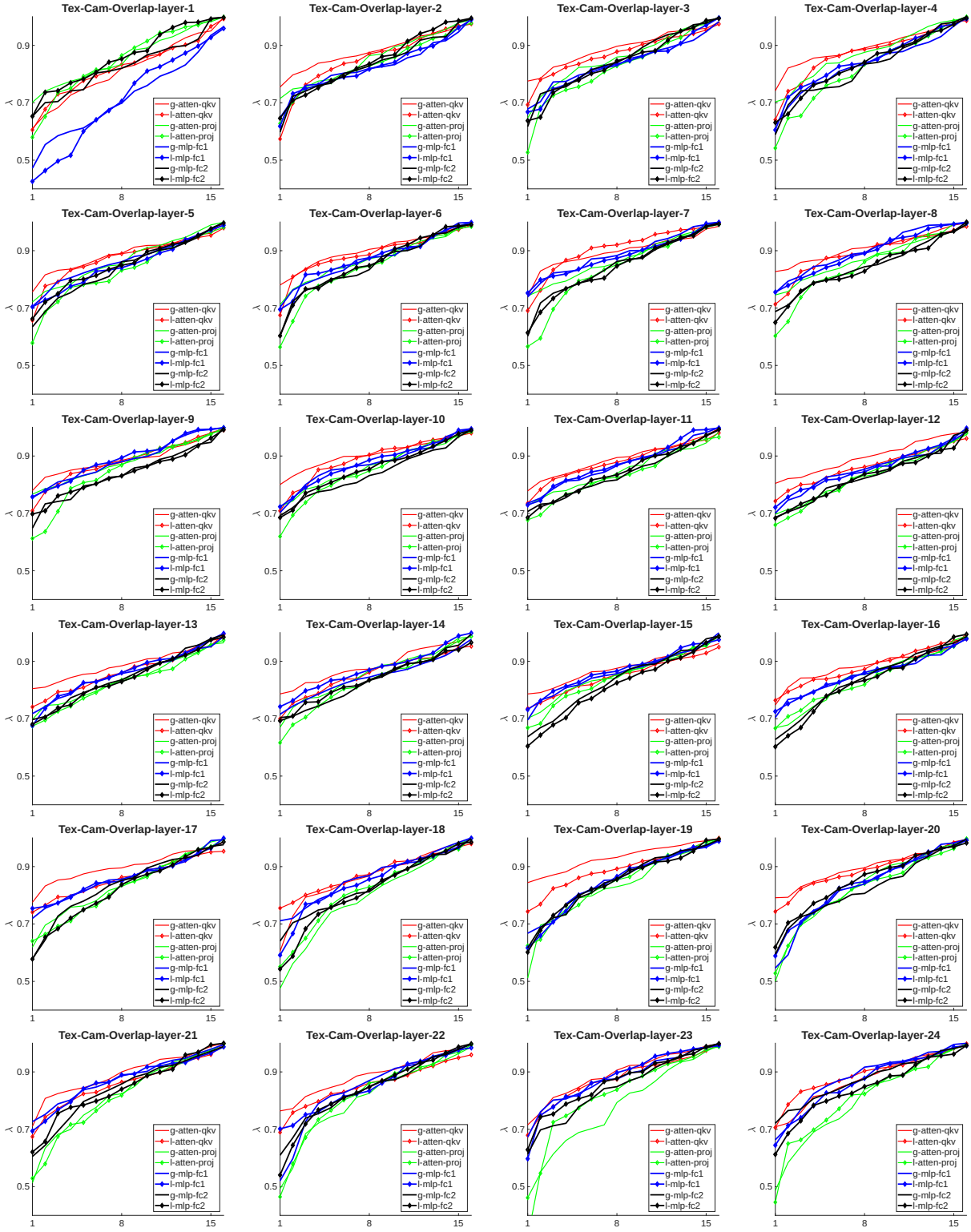

Figure 38. The overlap ratio between subspaces that correspond to variations in texture and camera motion.

5.5. Texture vs Lighting

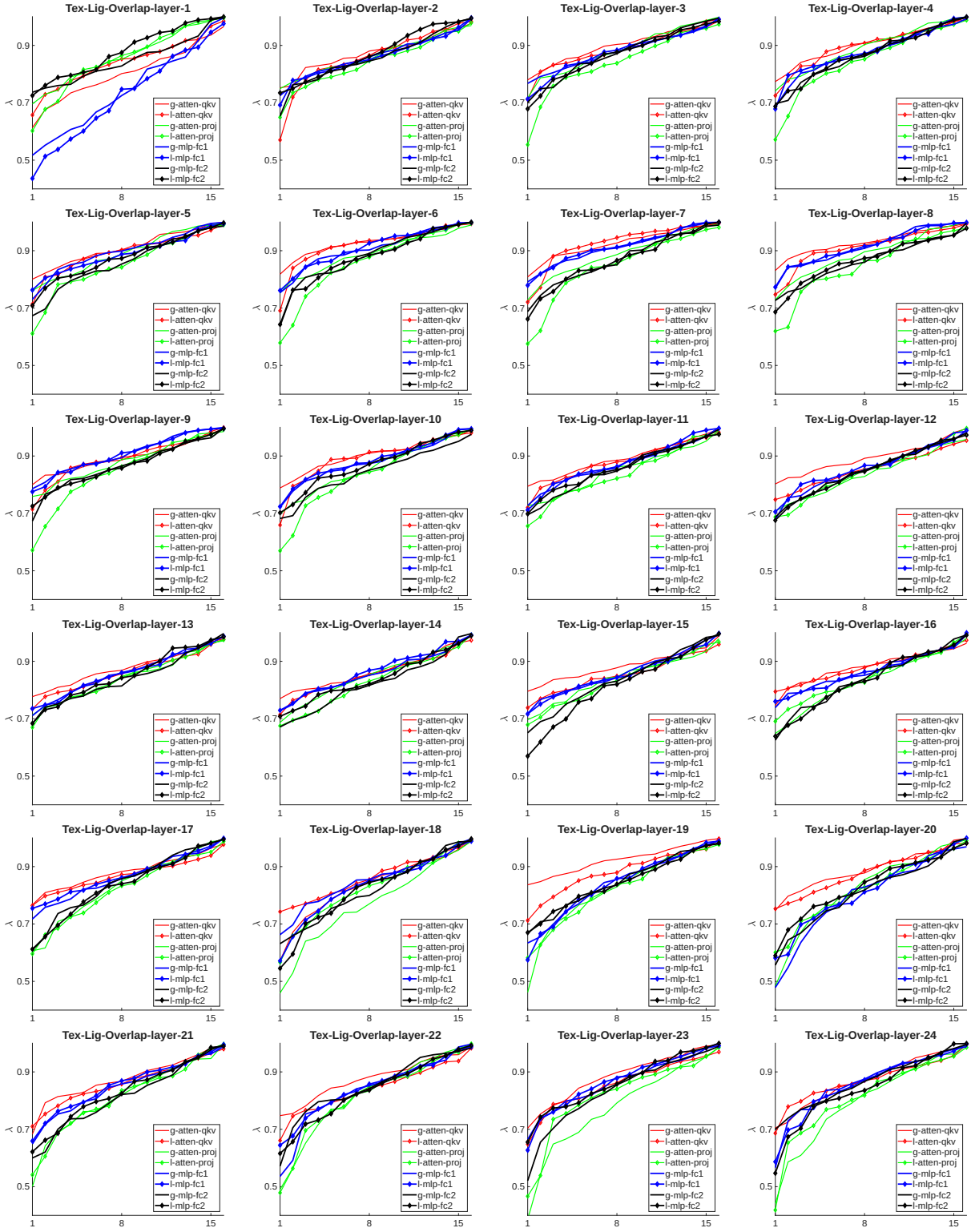

Figure 39. The overlap ratio between subspaces that correspond to variations in texture and lighting.

5.6. Camera vs Lighting

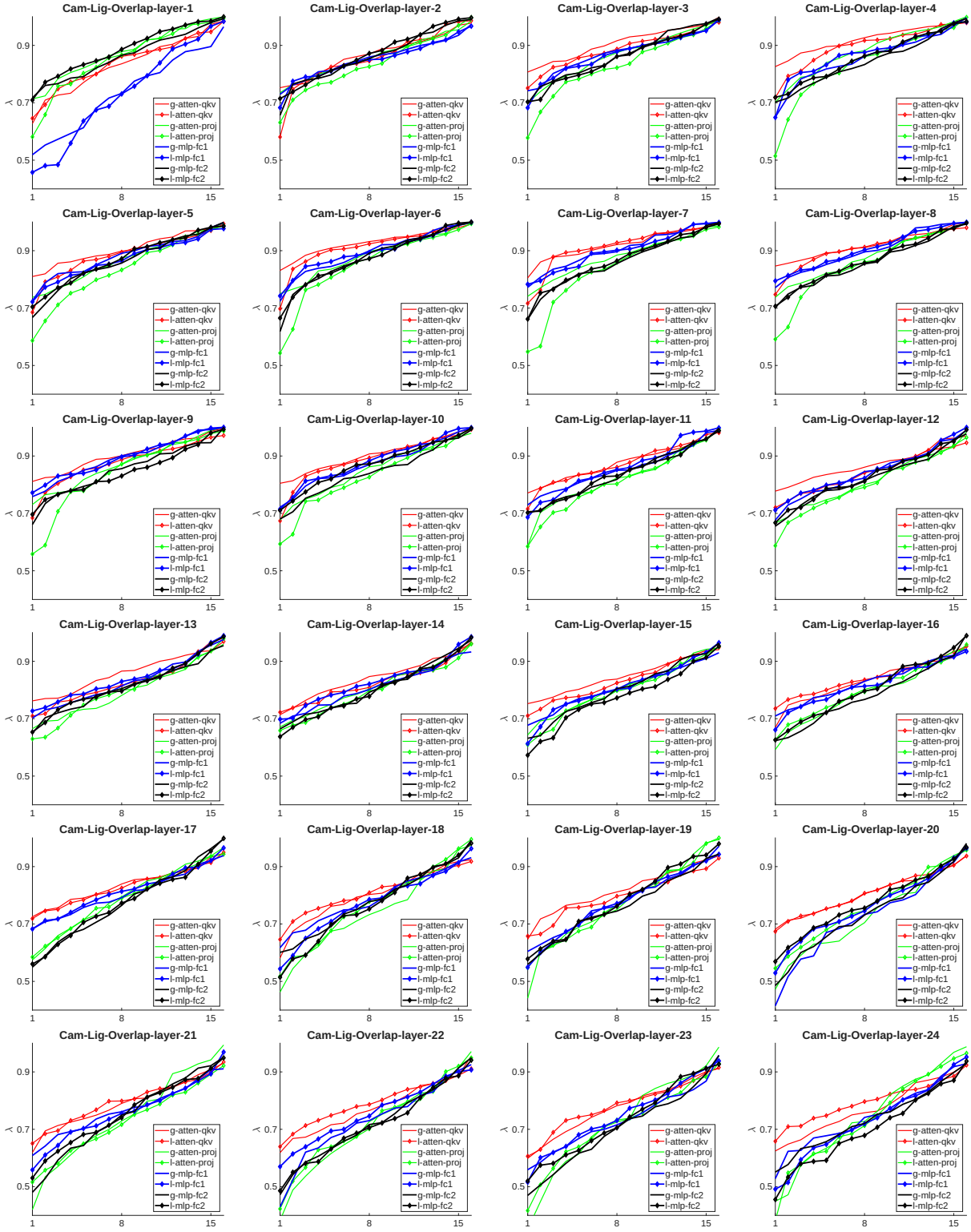

Figure 40. The overlap ratio between subspaces that correspond to variations in camera motion and lighting.

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