SVD^3: Singular Value Decomposition for Visual Geometry Model Compression

Anonymous ACCV 2026 Submission

Paper ID #414

Abstract. Feed-forward visual geometry models have demonstrated remarkable success in jointly predicting diverse geometric attributes. Despite their strong performance, these models incur substantial computational cost: linear projections in both attention and MLP layers constitute the dominant bottleneck, accounting for the majority of FLOPs during inference. In this work, we revisit low-rank weight decomposition for visual geometry models via singular value decomposition (SVD), aiming to reduce computational complexity and accelerate inference. While SVD effectively reduces parameter count and FLOPs, we find that fixed-ratio compression is suboptimal, as the importance of singular values varies considerably across input samples. Motivated by this observation, we propose **SVD**^{3}, a training-free, data-adaptive compression framework for visual geometry models. SVD^3 introduces an entropy-guided compression allocation strategy that adaptively decomposes linear layers into low-rank matrices, and a retention mapping module that assigns each input a discrete retention ratio under a predefined compute budget estimated on a calibration dataset. Extensive experiments demonstrate that SVD^3 reduces inference cost by over 30% on both VGGT and π^3, with significantly lower performance degradation compared to fixed-ratio compression baselines. Our source code is provided in the supplementary material.

Keywords: Efficiency · 3D vision · Singular value decomposition

1 Introduction

Recent advances in feed-forward visual geometry models have achieved remarkable performance across a wide range of geometric understanding tasks, including depth estimation [7,32,34,35], camera pose estimation [21,25,31], and point cloud reconstruction [1,11,12]. Models such as VGGT [23] and π^3 [28] demonstrate that unified large-scale pretraining can generalize effectively across indoor, outdoor, synthetic, and real-world domains. Unlike optimization-based approaches such as NeRF [16] and 3D Gaussian Splatting [10], these feed-forward models produce direct predictions in a single forward pass, enabling substantially faster inference. Furthermore, their architectures are amenable to large-scale training, allowing them to scale gracefully to massive datasets and diverse deployment environments.

However, these feed-forward 3D models comprise numerous attention and MLP layers, and their performance gains come at a substantial computational

cost [18, 20, 29]. To address this inference-time bottleneck, recent methods focus on reducing the number of tokens processed by the model. For instance, several works [18, 20, 29] exploit token redundancy within these models, proposing to dynamically merge similar tokens and summarize less salient ones to reduce inference computation. Beyond token reduction, model quantization frameworks [6] have also been proposed to compress feed-forward visual geometry models for inference acceleration.

Distinct from these approaches, low-rank matrix decomposition offers a principled means of reducing the rank of weight matrices and lowering computational complexity [5, 13, 26, 27, 37], yet remains underexplored in the context of efficient visual geometry models. We argue that low-rank decomposition holds significant promise for accelerating visual geometry inference for two reasons: (i) visual geometry models are predominantly composed of linear layers that are naturally amenable to decomposition — as shown in Tab. 1, linear layers account for the vast majority of parameters in both VGGT and π^3: 91.65% and 99.74%, respectively; and (ii) low-rank decomposition is hardware-agnostic and robust to kernel variations [9]. Together, these properties make singular value decomposition (SVD) a natural and compelling solution for accelerating inference in visual geometry models.

Table 1: Parameter distribution comparison. The linear layers are the most dominant source of computation.

	VGGT		π^3	
Module	**#Params**	**%**	**#Params**	**%**
Linear	1151 M	**91.65%**	956 M	**99.74%**
Conv2d	82 M	6.56%	603 K	0.06%
ConvTranspose2d	6 M	0.50%	–	–
LayerNorm	410 K	0.03%	318 K	0.03%
LayerScale	163 K	0.01%	12 K	0.01%

However, directly applying SVD to feed-forward visual geometry models is non-trivial. We observe that applying the same degree of low-rank decomposition to different 3D scenes yields substantially different performance degradation. We attribute this to the varying representational demands of different scenes: since SVD compression retains information according to relative importance, inputs with greater complexity inherently require higher representation capacity, and thus benefit from a higher retention ratio. This motivates a data-adaptive compression strategy that dynamically allocates model capacity based on the characteristics of each input sample.

To this end, we propose **SVD^3**, a training-free, inference-time adaptive compression framework for feed-forward visual geometry models. We introduce an entropy-guided compression allocation strategy that estimates input complexity via the Shannon entropy of raw pixels. Normalized entropy scores are then mapped to discrete retention regimes, such that the expected retention ratio satisfies a predefined global compute budget derived from a small calibration dataset. At inference time, our dynamic rank allocation strategy assigns each input sample a different effective rank without incurring additional computational overhead or memory cost. Importantly, our approach is also orthogonal to prior token merging [20, 29, 33] and quantization [6] techniques, as each class of methods

addresses a distinct source of computational redundancy. Token merging decreases the number of visual tokens that must be processed, and quantization lowers numerical precision, whereas our method instead reduces the effective rank of weight matrices via adaptive low-rank decomposition. Consequently, these strategies can be seamlessly combined to yield cumulative efficiency improvements.

We evaluate our approach on two state-of-the-art visual geometry models, VGGT and π^3, across a wide range of benchmarks encompassing depth estimation, camera pose estimation, and point cloud reconstruction. Experimental results demonstrate that SVD^3 reduces inference cost by over 30% while incurring significantly less performance degradation than fixed-ratio alternatives. These findings suggest that aligning model capacity with input complexity yields a simple yet effective strategy for compressing feed-forward visual geometry models. Our contributions are summarized as follows:

- We identify that a uniform compression ratio leads to inconsistent performance across input samples of varying scene complexity, motivating the need for input-adaptive compression.
- We propose **SVD^3**, a data-adaptive low-rank decomposition framework for feed-forward visual geometry models that dynamically assigns per-sample compression ratios guided by raw-pixel Shannon entropy. Our method adjusts the effective rank at inference time without requiring multiple compressed model checkpoints for different retention ratios.
- Extensive experiments on two representative feed-forward visual geometry models validate that SVD^3 consistently outperforms fixed-ratio baselines across a wide range of tasks and compression settings.

2 Related Work

2.1 3D Visual Geometry Models

3D visual geometry models [23,24,28,33,39] have emerged as a promising paradigm for jointly predicting multiple high-quality 3D attributes within a single forward pass. These models are designed to collectively estimate key 3D properties such as point clouds, camera parameters, and depth maps, and are typically trained on large, diverse dataset collections spanning multiple domains. CUT3R [24] is a recurrent transformer that continuously updates its internal state representation upon each new scene observation. Fast3R [33] departs from the recurrent paradigm by processing multiple images in parallel. FLARE [39] introduces a cascaded pipeline that first estimates camera pose as an intermediate representation to condition downstream geometric predictions. VGGT [23] alternates between frame-level local attention and sequence-level global attention to effectively aggregate information across multiple input images. π^3 [28] achieves permutation-equivariant reconstruction by discarding reference-view bias through the removal of positional embeddings and camera tokens. Despite their strong performance across a wide array of downstream tasks, the large encoder–decoder architectures of these models — comprising numerous linear layers — raise significant efficiency concerns when scaling to long-sequence image inputs [18, 20, 29].

2.2 Efficient Visual Geometry Learning

Reducing the number of tokens processed by subsequent layers is a natural and effective strategy for improving inference efficiency and scalability. Several training-free methods have been proposed to prune or merge tokens between layers. FastVGGT [18] retains the most salient tokens and merges the remaining ones based on cosine similarity. LiteVGGT [20] refines this token merging strategy by quantifying the geometric importance of each token via pixel gradients and token variance. FlashVGGT [29] instead compresses spatial information from each frame into a compact set of descriptor tokens. Beyond token reduction, post-training quantization (PTQ) [14, 30, 38] offers a complementary route to inference acceleration: QuantVGGT [6] improves the quantization pipeline by smoothing inter-channel variance and constructing diverse calibration sets. While these methods are effective training-free approaches, the application of low-rank matrix decomposition to visual geometry models remains largely underexplored.

2.3 Singular Value Decomposition

SVD approximates a weight matrix $\mathbf{W}$ via low-rank factors by minimizing the Frobenius-norm reconstruction error: $\mathbf{W}_k^\star = \arg\min_{\mathbf{W}_k} \|\mathbf{W}_k - \mathbf{W}\|_F^2$, where $\mathbf{W}_k$ is the rank-k approximation. This property has made SVD an attractive tool for large model compression, particularly for large language models (LLMs). Rather than compressing weights in isolation, ASVD [37] incorporates input activations into the decomposition objective, enforcing the compressed layer to preserve the output of the original layer. Given activations $\mathbf{X}$ cached from a small calibration set, the objective becomes: $\mathbf{W}_k^\star = \arg\min_{\mathbf{W}_k} \|\mathbf{W}_k\mathbf{X} - \mathbf{W}\mathbf{X}\|_F^2$. However, ASVD does not guarantee that truncating smaller singular values monotonically reduces the activation-weighted compression loss. SVD-LLM [27] addresses this by introducing truncation-aware data whitening: a whitening matrix $\mathbf{S}$ is derived via Cholesky decomposition such that the whitened activation $\mathbf{S}^{-1}\mathbf{X}$ is orthogonal. SVD is then applied to $\mathbf{WS}$, followed by multiplication by $\mathbf{S}^{-1}$ to recover the compressed weight $\mathbf{W}_k$. SVD-LLM V2 [26] extends this by assigning layer-wise compression ratios according to a theoretical minimum truncation loss, accounting for varying degrees of weight redundancy across layers. AdaSVD [13] instead leverages the similarity between input and output activations to guide compression budget allocation. DipSVD [5] combines Fisher sensitivity and effective rank as dual compressibility heuristics for more robust layer-wise compression. Despite these advances, all existing SVD methods are *sample-agnostic*: they apply a fixed compression budget uniformly across all inputs, ignoring the varying scene complexity of individual samples.

3 Methodology

3.1 Preliminary

Given a weight matrix $\mathbf{W} \in \mathbb{R}^{m\times n}$, SVD factorizes it into three components:

$$\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top, \tag{1}$$

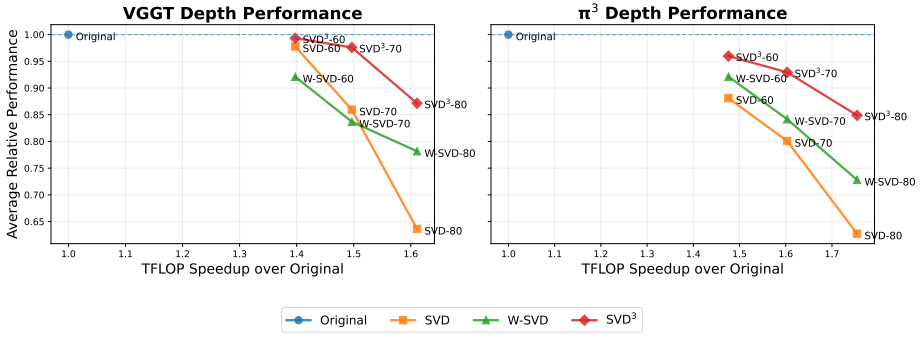

Fig. 1: Accuracy-efficiency trade-offs for depth estimation on Pi3 and VGGT. SVD³ consistently achieves higher relative performance than SVD and W-SVD at comparable TFLOP reductions.

where $\mathbf{U}$ and $\mathbf{V}$ are the matrices of left and right singular vectors, respectively, and $\mathbf{\Sigma}$ is a diagonal matrix whose entries are the singular values, encoding the relative importance of each component of $\mathbf{W}$. Given a predefined retention ratio $\alpha \in (0, 1)$ — the fraction of parameters to retain — the corresponding rank r is determined by:

$$\alpha = \frac{r(m+n)}{m \times n}. \tag{2}$$

The compressed weight matrix $\mathbf{W}'$ is then obtained by truncating the smallest singular values:

$$\mathbf{W}' = \mathbf{U}\,\mathrm{Trunc}_r(\mathbf{\Sigma})\mathbf{V}^\top, \tag{3}$$

where $\mathrm{Trunc}_r(\cdot)$ retains only the top-r singular values and zeros out the rest.

To guarantee a theoretical minimum truncation loss at each layer, SVD-LLM [27] proposes truncation-aware data whitening, which aligns the truncated singular values with the directions of minimal compression loss. Specifically, a whitening matrix $\mathbf{S}$ is computed from input activations $\mathbf{X}$ collected on a small calibration dataset, such that the whitened activation $\mathbf{S}^{-1}\mathbf{X}$ is orthogonal:

$$(\mathbf{S}^{-1}\mathbf{X})(\mathbf{S}^{-1}\mathbf{X})^\top = \mathbf{S}^{-1}\mathbf{X}\mathbf{X}^\top(\mathbf{S}^{-1})^\top = \mathbf{I}. \tag{4}$$

This reduces to $\mathbf{S}\mathbf{S}^\top = \mathbf{X}\mathbf{X}^\top$, which can be solved efficiently via Cholesky decomposition. SVD is then applied to the whitened weight matrix $\mathbf{W}\mathbf{S}$, and the compressed weight $\mathbf{W}'$ is recovered by inverting the whitening transform:

$$\mathbf{W}' = \mathbf{U}'\,\mathrm{Trunc}_r(\mathbf{\Sigma})\mathbf{V}'^\top\mathbf{S}^{-1}. \tag{5}$$

3.2 One Compression Ratio Does Not Fit All

To evaluate whether a fixed compression ratio yields robust performance across diverse inputs, we apply both standard SVD and whitening-based SVD (W-SVD)

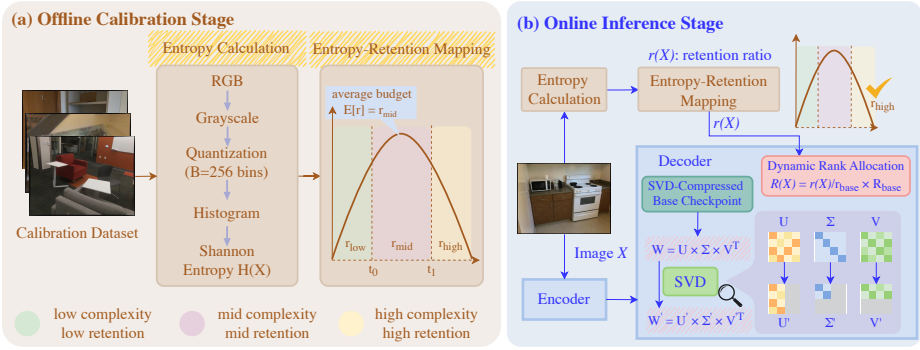

Fig. 2: SVD^3 pipeline. **(a) Offline Calibration Stage.** Input complexity is estimated via normalized raw-pixel entropy on a calibration set. Learned thresholds partition inputs into low/medium/high regimes with predefined average retention ratios. **(b) Online Inference Stage.** At inference, each input sample first computes its normalized entropy score to determine its retention level; after encoding, the decoder dynamically adjusts its effective rank by further slicing the SVD-decomposed weight matrices.

to compress the weight matrices of the π^3 decoder and measure monocular depth estimation performance on multiple benchmarks [3, 8, 17]. As illustrated in Fig. 1, both approaches display a clear trade-off between accuracy and efficiency: stronger compression enhances computational efficiency but causes marked, and often irregular, performance drops on both VGGT and π^3. This effect arises because a uniform compression ratio is applied to every sample, even though they differ in visual complexity and information density. These observations motivate **SVD^3**, a data-adaptive SVD framework that dynamically allocates sample-specific compression ratios according to input complexity, achieving more favorable accuracy-efficiency trade-offs.

3.3 SVD^3

Given that a uniform fixed compression ratio fails to generalize across input samples of varying complexity, we propose **SVD^3**, a training-free, data-adaptive SVD compression framework. As illustrated in Fig. 2, SVD^3 operates in two stages. During calibration, a small set of reference images is used to estimate input complexity via normalized raw-pixel entropy, and entropy thresholds are determined to partition inputs into discrete complexity regimes (low, medium, and high), each associated with a predefined retention ratio. At inference time, each input sample first computes its normalized entropy score to determine its assigned retention regime; the decoder then dynamically adjusts its effective rank by slicing the pre-decomposed weight matrices to match the corresponding retention ratio. Each component is described in detail in the following subsections.

Offline Calibration. As illustrated in Fig. 2, we first construct a small calibration dataset of RGB images to estimate per-sample input complexity. Given an

RGB image $\mathbf{X} \in \mathbb{R}^{3\times H\times W}$ with pixel values normalized to $[0,1]$, we convert it to a grayscale image $G(h,w)$ and uniformly quantize it into B discrete bins:

$$Q(h,w) = \lfloor G(h,w)\,(B-1) \rfloor\,. \tag{6}$$

We adopt $B = 256$ histogram bins to align with the standard 8-bit image intensity range. This choice provides a sufficiently fine-grained approximation of the grayscale intensity distribution. The pixel intensity histogram is:

$$p_i = \frac{1}{HW}\sum_{h,w} \mathbb{I}\big[Q(h,w) = i\big], \quad i = 0,\ldots,B-1, \tag{7}$$

where $\mathbb{I}[\cdot]$ is the indicator function. The pixel-level Shannon entropy of the input is then defined as:

$$H(\mathbf{X}) = -\sum_{i=0}^{B-1} p_i \log_2 p_i. \tag{8}$$

This entropy serves as a simple yet effective proxy for scene complexity, and is used to assign an adaptive compression ratio to each input sample. Importantly, our method does not require a highly precise estimator; rather, it only relies on a computationally inexpensive heuristic that sufficiently correlates with inference complexity to enable data-adaptive rank allocation. In order to enable data-adaptive compression while maintaining a fixed average compute budget, we map each per-image entropy score to a discrete retention ratio, as illustrated in Fig. 2. Raw-pixel entropy serves as a proxy for input complexity, and is quantized into a small set of predefined retention levels whose expected value is constrained to match a target budget under the calibration distribution.

Formally, rather than thresholding raw entropy values directly, we first normalize entropy scores computed on the calibration dataset. Specifically, we compute the 5th and 95th percentiles of the calibration entropy distribution, denoted p_5 and p_{95}, and map each score to a normalized value:

$$s_{\text{norm}}(\mathbf{X}) = \text{clip}\left(\frac{H(\mathbf{X}) - p_5}{p_{95} - p_5},\, 0,\, 1\right). \tag{9}$$

We then determine two thresholds $t_0 < t_1$ that partition the normalized entropy distribution into three regimes: low, medium, and high. The thresholds are set as symmetric quantiles of the calibration distribution: given a user-specified tail fraction $\alpha \in (0, 0.5)$, we set $t_0 = Q_\alpha(s_{\text{norm}})$ and $t_1 = Q_{1-\alpha}(s_{\text{norm}})$, where $Q_q(\cdot)$ denotes the q-th quantile. By construction, a fraction α of calibration samples fall into the low-entropy regime, a fraction α into the high-entropy regime, and the remaining $1-2\alpha$ into the medium-entropy regime. Importantly, we do not adjust α across datasets, tasks, or hardware platforms. In all experiments, we simply set $\alpha = 0.25$, which induces a stable split into low-, medium-, and high-entropy samples. We observe that this straightforward, fixed partition already offers sufficient diversity and eliminates the need for dataset-specific hyperparameter tuning. Each regime is assigned a fixed retention ratio $\{r_{\text{low}}, r_{\text{mid}}, r_{\text{high}}\}$, corresponding

to aggressive, moderate, and conservative compression, respectively. The resulting mapping assigns a retention ratio to each input as follows:

$$r(\mathbf{X}) = \begin{cases} r_{\text{low}}, & s_{\text{norm}}(\mathbf{X}) \le t_0, \\ r_{\text{mid}}, & t_0 < s_{\text{norm}}(\mathbf{X}) < t_1, \\ r_{\text{high}}, & s_{\text{norm}}(\mathbf{X}) \ge t_1. \end{cases} \tag{10}$$

Concrete examples of entropy-guided ratio assignment are provided in the supplementary material. By design, this construction enforces a fixed expected retention ratio under the calibration distribution:

$$\mathbb{E}[r] = \alpha\, r_{\text{low}} + (1 - 2\alpha)\, r_{\text{mid}} + \alpha\, r_{\text{high}} = r_{\text{mid}}, \tag{11}$$

where the constraint $r_{\text{low}} + r_{\text{high}} = 2r_{\text{mid}}$ ensures budget compliance. Consequently, while individual samples are compressed adaptively according to their complexity, the average retention ratio over the calibration distribution is guaranteed to equal the target budget r_{mid}.

Online Inference. At inference time, for each input image $\mathbf{X}$, we first compute its normalized entropy score as a lightweight proxy for visual complexity. Based on the pre-determined thresholds t_0 and t_1, the entropy score is mapped to one of three discrete retention ratios $\{r_{\text{low}}, r_{\text{mid}}, r_{\text{high}}\}$, as described in the previous subsection. Given the assigned retention ratio $r(\mathbf{X})$, the decoder dynamically adjusts its effective rank by slicing the pre-decomposed weight matrices accordingly.

Concretely, we maintain a single SVD-compressed model checkpoint with a base retention ratio r_{base}, where $r_{\text{base}} \ge r_{\text{high}}$, and denote the corresponding rank of any weight matrix in this checkpoint as R_{base}. Importantly, this checkpoint is *already heavily compressed* compared to the original dense model, because $r_{\text{base}} \ll 1$. In practice, we simply choose $r_{\text{base}} = r_{\text{high}} + 0.1$, which effectively removes the need to keep a copy of the full uncompressed model. The effective rank for input $\mathbf{X}$ is computed as:

$$R(\mathbf{X}) = \left\lfloor \frac{r(\mathbf{X})}{r_{\text{base}}}\, R_{\text{base}} \right\rceil, \quad 1 \le R(\mathbf{X}) \le R_{\text{base}}. \tag{12}$$

The input-adaptive compressed weight matrix is then obtained by retaining only the top-$R(\mathbf{X})$ components:

$$\mathbf{W}'(\mathbf{X}) = \mathbf{U}_{\text{base}}[:, : R(\mathbf{X})]\, \mathbf{\Sigma}_{\text{base}}[: R(\mathbf{X}), : R(\mathbf{X})]\, \mathbf{V}_{\text{base}}[: R(\mathbf{X}), :]^{\top}, \tag{13}$$

where $\mathbf{U}_{\text{base}}$, $\mathbf{\Sigma}_{\text{base}}$, and $\mathbf{V}_{\text{base}}$ are the left singular vectors, singular values, and right singular vectors of the base compressed weight matrix, satisfying $\mathbf{W} \approx \mathbf{U}_{\text{base}} \mathbf{\Sigma}_{\text{base}} \mathbf{V}_{\text{base}}^{\top}$. Importantly, by dynamically adjusting the effective rank at inference time from a single checkpoint, our approach eliminates the need to store multiple compressed models for different retention ratios.

Table 2: Monocular Depth Estimation results. We report GFLOPs, Rel, and $\delta < 1.25$. The **best** results are highlighted.

		VGGT							π^3						
			Sintel		Bonn		KITTI			Sintel		Bonn		KITTI	
α	Method	GFLOPs	Rel↓	$\delta<1.25$↑	Rel↓	$\delta<1.25$↑	Rel↓	$\delta<1.25$↑	GFLOPs	Rel↓	$\delta<1.25$↑	Rel↓	$\delta<1.25$↑	Rel↓	$\delta<1.25$↑
–	Original	293.65	0.3325	0.6012	0.0514	0.9722	0.0834	0.9455	249.70	0.2796	0.6164	0.0482	0.9736	0.0591	0.9708
60%	SVD	198.95	0.4011	0.5716	0.0580	**0.9670**	0.1233	0.8972	178.67	**0.2681**	**0.6607**	0.0496	**0.9752**	0.0748	0.9595
	W-SVD	198.95	0.3529	0.5932	0.0591	0.9391	0.1053	0.9183	178.67	0.3156	0.6134	0.0540	0.9603	0.0754	0.9538
	SVD^3	198.95	**0.3339**	**0.6012**	**0.0541**	0.9648	**0.0990**	**0.9251**	178.67	0.2706	0.6335	**0.0497**	0.9749	**0.0634**	**0.9661**
70%	SVD	183.17	0.4093	0.5647	0.0772	0.9526	0.1538	0.8181	166.84	0.3099	0.6278	0.0640	0.9626	0.1088	0.9231
	W-SVD	183.17	0.3916	0.5573	0.0661	0.9284	0.1334	0.8656	166.84	0.3377	0.6066	0.0679	0.9131	0.0986	0.9339
	SVD^3	183.17	**0.3539**	**0.5935**	**0.0577**	**0.9670**	**0.1052**	**0.9184**	166.84	**0.2687**	**0.6603**	**0.0497**	**0.9752**	**0.0754**	**0.9597**
80%	SVD	167.39	0.4755	0.5002	0.1595	0.7739	0.2026	0.6644	155.00	0.3969	0.5643	0.1252	0.8183	0.1989	0.6549
	W-SVD	167.39	0.4237	0.5262	0.0745	0.9424	0.2147	0.6243	155.00	0.3409	0.5902	0.0657	0.9398	0.1531	0.8027
	SVD^3	167.39	**0.3932**	**0.5703**	**0.0661**	**0.9527**	**0.1333**	**0.8661**	155.00	**0.3090**	**0.6272**	**0.0639**	**0.9627**	**0.0984**	**0.9341**

4 Experiments

4.1 Experiment Setup

Calibration Dataset. We randomly collect 256 samples from ScanNet V2 [36] for data whitening with the random seed set to 3. Similar to SVD-LLM [27], we choose to only sample a moderate number of images, which is proved to be enough for obtaining the whitening matrix S: increasing the calibration set size beyond 256 produces negligible changes in the learned entropy thresholds and downstream performance. All images are resized to 224×224. We use the exact same calibration dataset for budget-forcing and threshold learning.

Implementation Details. In order to preserve accurate image embeddings we decide to apply singular value decomposition (SVD) on the decoder's weights while keeping the encoder intact. We evaluate our proposed method on both VGGT and π^3. Specifically, for π^3, we decompose 144 linear layers across all 36 decoder layers. For VGGT, we target 192 linear layers spanning all 24 frame-level blocks and 24 global-level blocks in its aggregator (i.e., decoder). These layers include the QKV projections, attention output projections, and the two fully connected layers in each MLP block.

Evaluation metrics. For depth prediction, we adopt Absolute Relative Error (Rel) and accuracy under threshold $\delta < 1.25$. The former measures the average relative deviation between the predicted depth and the ground-truth depth, while the later evaluates the percentage of pixels whose predicted depth is sufficiently close to the ground truth. For camera pose estimation, we adopt three standard metrics: Absolute Trajectory Error (ATE), Relative Pose Error in translation (RPE-t), and Relative Pose Error in rotation (RPE-r). ATE measures the global alignment error between the predicted and ground-truth trajectories, reflecting overall pose consistency. RPE-t evaluates local translational drift between consecutive frames, while RPE-r measures the corresponding rotational drift. For point cloud reconstruction, we adopt reconstruction accuracy (Acc), completeness

Table 3: Video Depth Estimation results. We report GFLOPs, Rel, and $\delta < 1.25$. The **best** results are highlighted.

		VGGT							π^3						
			Sintel		Bonn		KITTI			Sintel		Bonn		KITTI	
α	Model	GFLOPs	Rel↓	$\delta<1.25$↑	Rel↓	$\delta<1.25$↑	Rel↓	$\delta<1.25$↑	GFLOPs	Rel↓	$\delta<1.25$↑	Rel↓	$\delta<1.25$↑	Rel↓	$\delta<1.25$↑
-	Original	293.65	0.2277	0.6844	0.0503	0.9711	0.0538	0.9668	249.70	0.2106	0.7272	0.0438	0.9753	0.0366	0.9853
60%	SVD	198.95	**0.2610**	0.6574	0.0551	**0.9763**	0.0783	0.9408	178.67	0.2081	0.6763	0.0577	0.9744	0.0784	0.9397
	W-SVD	198.95	0.3276	0.6362	0.0686	0.9053	0.0802	0.9370	178.67	0.2215	0.6621	0.0461	0.9801	0.0582	0.9757
	SVD^3	198.95	0.2645	**0.6766**	**0.0500**	0.9707	**0.0662**	**0.9565**	178.67	**0.2034**	**0.7296**	**0.0436**	**0.9795**	**0.0427**	**0.9829**
70%	SVD	183.17	0.3077	0.6367	0.0766	0.9584	0.1150	0.8714	166.84	0.2504	0.6487	0.0755	0.9570	0.1201	0.8521
	W-SVD	183.17	0.3156	0.5637	0.0672	0.9574	0.1092	0.8873	166.84	0.2514	0.6343	0.0683	0.8954	0.0761	0.9512
	SVD^3	183.17	**0.2613**	**0.6570**	**0.0548**	**0.9763**	**0.0782**	**0.9406**	166.84	**0.2053**	**0.7207**	**0.0457**	**0.9801**	**0.0536**	**0.9769**
80%	SVD	167.39	0.4242	0.5197	0.1812	0.7629	0.1649	0.7495	155.00	0.2895	0.6186	0.0873	0.9245	0.1494	0.7767
	W-SVD	167.39	0.3381	0.5209	0.0764	0.9308	0.1617	0.7588	155.00	0.3218	0.5965	0.0685	0.9530	0.1057	0.8859
	SVD^3	167.39	**0.3077**	**0.6366**	**0.0672**	**0.9583**	**0.1091**	**0.8872**	155.00	**0.2081**	**0.6763**	**0.0577**	**0.9744**	**0.0761**	**0.9513**

Table 4: Camera Pose Estimation results. We report ATE, RPE-t, and RPE-r. The **best** results are highlighted.

		VGGT									π^3								
			Sintel			TUM			ScanNet			Sintel			TUM			ScanNet		
α	Model	GFLOPs	ATE↓	RPE-t↓	RPE-r↓	ATE↓	RPE-t↓	RPE-r↓	ATE↓	RPE-t↓	RPE-r↓	GFLOPs	ATE↓	RPE-t↓	RPE-r↓	ATE↓	RPE-t↓	RPE-r↓	ATE↓	RPE-t↓	RPE-r↓
-	Original	293.65	0.1665	0.0607	0.4942	0.0121	0.0103	0.3116	0.0357	0.0152	0.3810	249.70	0.0732	0.0391	0.2767	0.0142	0.0089	0.3119	0.0305	0.0126	0.3464
60%	SVD	198.95	**0.1571**	0.0719	0.7417	0.0180	0.0128	0.3354	0.0488	0.0199	0.4943	178.67	0.1340	0.0607	0.4988	0.0194	0.0126	0.3352	0.0404	0.0165	0.4093
	W-SVD	198.95	0.1876	0.1013	0.7528	0.0254	0.0129	0.3512	0.0522	0.0210	0.7019	178.67	0.1061	0.0473	0.3220	0.0171	0.0109	0.3114	0.0371	0.0145	0.3746
	SVD^3	198.95	0.1626	**0.0697**	**0.6683**	**0.0136**	**0.0106**	**0.3095**	**0.0389**	**0.0168**	**0.4062**	178.67	**0.0936**	**0.0409**	**0.2832**	**0.0137**	**0.0093**	**0.3081**	**0.0332**	**0.0136**	**0.3588**
70%	SVD	183.17	0.1982	0.0957	**0.6069**	0.0284	0.0172	0.4076	0.0694	0.0289	0.7950	166.84	0.1740	0.0716	0.6161	0.0453	0.0199	0.4175	0.0607	0.0207	0.5323
	W-SVD	183.17	0.2446	0.0871	0.9075	0.0439	0.0163	0.4252	0.0826	0.0250	0.8725	166.84	**0.1034**	0.0569	0.3854	0.0297	0.0140	0.3248	0.0495	0.0165	0.4136
	SVD^3	183.17	**0.1516**	**0.0654**	0.7110	**0.0153**	**0.0112**	**0.3182**	**0.0421**	**0.0177**	**0.4463**	166.84	0.1064	**0.0467**	**0.3203**	**0.0171**	**0.0108**	**0.3109**	**0.0337**	**0.0144**	**0.3693**
80%	SVD	167.39	0.2527	0.1441	1.0020	0.1125	0.0579	1.0021	0.1476	0.0519	2.0569	155.00	0.1956	0.0773	0.7284	0.0509	0.0190	0.3945	0.0743	0.0212	0.5494
	W-SVD	167.39	0.2789	0.1024	1.0472	0.0668	0.0238	0.5779	0.1418	0.0363	1.3518	155.00	0.1864	0.0658	0.5424	0.0409	0.0155	0.3612	0.0856	0.0212	0.6763
	SVD^3	167.39	**0.1571**	**0.0719**	**0.7417**	**0.0180**	**0.0128**	**0.3354**	**0.0488**	**0.0199**	**0.4943**	155.00	**0.1034**	**0.0568**	**0.3839**	**0.0194**	**0.0126**	**0.3246**	**0.0404**	**0.0165**	**0.4093**

(Comp), and normal consistency (NC). Accuracy measures the average distance from reconstructed points to the ground-truth surface, reflecting global geometric precision. Completeness evaluates how well the reconstructed surface covers the ground-truth geometry. Normal consistency measures the alignment between predicted and ground-truth surface normals, capturing local structural fidelity. For efficiency quantification, we calculate the GFLOPs, measured in terms of a single forward pass over a single input image.

4.2 Results

In this subsection, we present our results across three downstream tasks: depth estimation (including monocular depth and video depth); camera pose estimation; point cloud reconstruction (including sparse scene reconstruction and dense scene reconstruction). We compare our method with two baselines introduced in the previous Sec. 3.1: SVD SVD and truncation-aware whitening SVD. To ensure the fair evaluation of the generalizability of our proposed approach, we report results across two 3D vision geometry models (i.e., VGGT and π^3 [23]) under three different computational budgets: 60%, 70%, and 80%.

Depth Estimation For both monocular depth and video depth estimation tasks, we report results on the most commonly used benchmark datasets: Sintel [3],

Table 5: Sparse Point Map Estimation results. We report GFLOPs, Acc, Comp, and NC. The **best** results are highlighted.

		VGGT							π^3						
			7scenes			NRGBD				7scenes			NRGBD		
α	Model	GFLOPs	Acc↓	Comp↓	NC↑	Acc↓	Comp↓	NC↑	GFLOPs	Acc↓	Comp↓	NC↑	Acc↓	Comp↓	NC↑
-	Original	293.65	0.0446	0.0637	0.7617	0.0527	0.0674	0.8957	249.70	0.0469	0.0736	0.7413	0.0239	0.0282	0.9086
60%	SVD	198.95	0.0797	0.1130	0.7343	0.0452	0.0553	0.8843	178.67	0.0553	0.0823	0.7307	0.0392	0.0431	0.8713
	W-SVD	198.95	0.0743	0.1140	0.7332	0.0478	0.0589	0.8806	178.67	0.0558	0.0886	0.7298	0.0829	0.0999	0.8396
	SVD^3	198.95	**0.0615**	**0.0909**	**0.7417**	**0.0360**	**0.0453**	**0.8967**	178.67	**0.0521**	**0.0804**	**0.7362**	**0.0316**	**0.0359**	**0.8883**
70%	SVD	183.17	0.0850	0.1458	0.7256	0.0609	0.0709	0.8619	166.84	0.0618	0.0894	0.7183	0.0540	0.0589	0.8428
	W-SVD	183.17	0.0774	0.1438	0.7229	0.1141	0.1509	0.8266	166.84	0.0642	**0.0794**	0.7306	0.0946	0.1195	0.8139
	SVD^3	183.17	**0.0739**	**0.1137**	**0.7328**	**0.0478**	**0.0590**	**0.8808**	166.84	**0.0524**	0.0821	**0.7326**	**0.0389**	**0.0428**	**0.8720**
80%	SVD	167.39	0.1178	0.2274	0.6827	0.1768	0.2412	0.7579	155.00	0.0929	0.1582	0.6560	0.1372	0.1782	0.7320
	W-SVD	167.39	0.0980	0.2331	0.6937	0.1880	0.3469	0.7588	155.00	0.0749	0.1332	0.7059	0.1476	0.1748	0.7668
	SVD^3	167.39	**0.0766**	**0.1432**	**0.7229**	**0.1141**	**0.1510**	**0.8265**	155.00	**0.0634**	**0.0792**	**0.7312**	**0.0945**	**0.1192**	**0.8138**

Table 6: Dense Point Map Estimation results. We report GFLOPs, Acc, Comp, and NC. The **best** results are highlighted.

		VGGT							π^3						
			7scenes			NRGBD				7scenes			NRGBD		
α	Model	GFLOPs	Acc↓	Comp↓	NC↑	Acc↓	Comp↓	NC↑	GFLOPs	Acc↓	Comp↓	NC↑	Acc↓	Comp↓	NC↑
-	Original	293.65	0.0191	0.0305	0.6913	0.0143	0.0161	0.8898	249.70	0.0158	0.0220	0.6886	0.0129	0.0136	0.8739
60%	SVD	198.95	0.0199	0.0331	**0.6913**	0.0266	0.0252	0.8469	178.67	0.0194	0.0239	0.6808	0.0298	0.0237	0.8259
	W-SVD	198.95	0.0229	0.0338	0.6865	0.0309	0.0278	0.8419	178.67	0.0193	0.0231	0.6821	0.0278	0.0218	0.8299
	SVD^3	198.95	**0.0189**	**0.0299**	0.6883	**0.0225**	**0.0216**	**0.8512**	178.67	**0.0178**	**0.0227**	**0.6827**	**0.0213**	**0.0182**	**0.8308**
70%	SVD	183.17	0.0302	0.0413	0.6880	0.0405	0.0351	0.8298	166.84	0.0275	0.0307	0.6703	0.0483	0.0361	0.8015
	W-SVD	183.17	0.0339	0.0456	0.6834	0.0475	0.0437	0.8207	166.84	0.0234	0.0253	**0.6830**	0.0460	0.0335	0.8123
	SVD^3	183.17	**0.0229**	**0.0337**	**0.6911**	**0.0309**	**0.0277**	**0.8421**	166.84	**0.0193**	**0.0231**	0.6824	**0.0278**	**0.0217**	**0.8303**
80%	SVD	167.39	0.0677	0.0910	0.6698	0.1058	0.0920	0.7664	155.00	0.0502	0.0435	0.6295	0.1001	0.0732	0.7162
	W-SVD	167.39	0.0713	0.1038	0.6664	0.1125	0.1196	0.7622	155.00	0.0368	0.0394	0.6704	0.0843	0.0635	0.7775
	SVD^3	167.39	**0.0338**	**0.0456**	**0.6837**	**0.0475**	**0.0437**	**0.8205**	155.00	**0.0234**	**0.0253**	**0.6831**	**0.0459**	**0.0335**	**0.8122**

Bonn [17], and KITTI [8]. For video depth estimation, we pick the first frame to measure the entropy score and assign compression ratios. Tab. 2 and Tab. 3 show that our data-adaptive strategy consistently outperforms fixed-compression baselines across different compression budgets. At a compression ratio of 60%, our adaptive variants achieve lower Absolute Relative Error and maintain higher $\delta < 1.25$ accuracy across most datasets by more conservatively compressing harder samples. When the compression ratio increases to 70% and 80%, the performance difference becomes more pronounced since scenes deteriorate rapidly under aggressive compression. Fixed-compression methods experience noticeable degradation in both error and accuracy, while the adaptive strategy maintains substantially better depth quality. This behavior is consistently observed in both monocular and video depth estimation settings.

Camera Pose Estimation For camera pose estimation, we report results on widely used benchmark datasets including Sintel [3], TUM [22], and ScanNet [4]. Experiments are conducted on both VGGT [23] and π^3 [28] under three compression ratios and we compare fixed-compression baselines against our data-adaptive variants. Tab. 4 shows that our data-adaptive strategy consistently outperforms

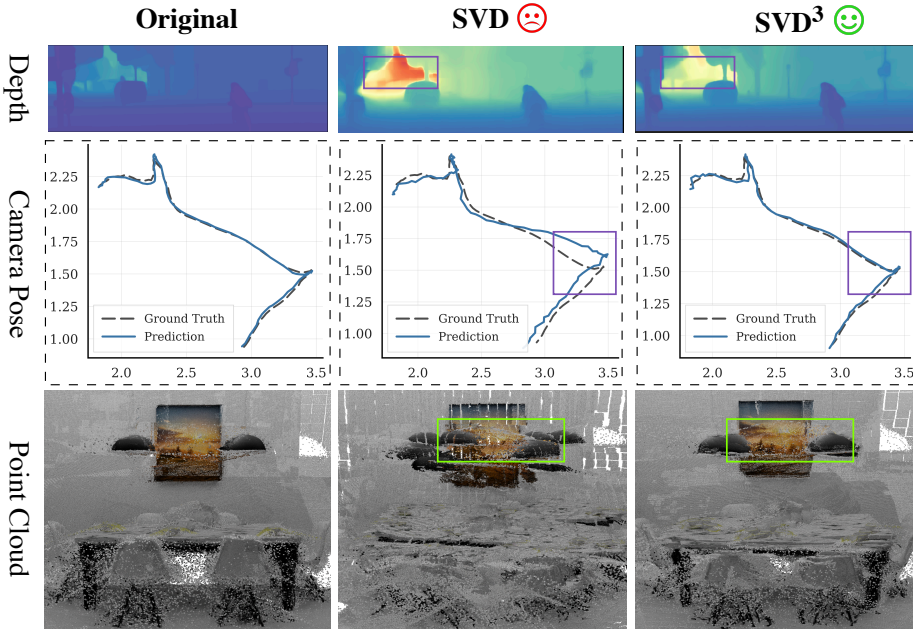

Fig. 3: Qualitative examples of video depth estimation, camera pose estimation, and point cloud reconstruction. Our data-adaptive method produces more stable depth results in high-complexity regions, preserves better alignment against the ground truth trajectory, and reconstructs more complete geometry.

fixed-compression baselines across different compression levels: at a compression ratio of 60%, the adaptive variants already achieve lower ATE and reduced relative pose errors across most datasets. The improvements are particularly evident in translational and rotational drift, indicating that adaptive compression-ratio allocation better preserves geometric consistency. As the compression ratio increases to 70% and 80%, the performance gap becomes more substantial since complex samples indeed require higher retention ratio to prevent performance collapse. Fixed-compression methods suffer from significant trajectory degradation under aggressive truncation, while our adaptive strategy maintains stronger global alignment and lower drift.

Point Cloud Reconstruction For both sparse and dense point map estimation, we report results on two commonly used benchmark datasets: 7-Scenes [19] and NRGBD [2]. We compare fixed-compression baselines against our data-adaptive variants under comparable GFLOPs budgets. Tab. 5 and Tab. 6 summarize the results. At a compression ratio of 60%, the adaptive variants already achieve lower reconstruction error and improved normal consistency across most datasets. The improvements are particularly evident in completeness and surface alignment, indicating that adaptive compression better preserves fine-grained geometric

structure. As the compression ratio increases to 70% and 80%, the performance gap becomes more stark. Fixed-compression methods suffer from noticeable degradation in both geometric accuracy and coverage, while the adaptive strategy maintains more stable reconstruction quality.

4.3 Qualitative Examples

Fig. 3 presents representative qualitative comparisons between the original model, fixed-compression baselines, and our data-adaptive approach across three tasks: camera pose estimation, video depth estimation, and point cloud reconstruction. For camera pose estimation, fixed compression results in more trajectory drift than our adaptive variant. For video depth estimation, fixed compression produces uncertain depth responses in high-complexity regions, but our adaptive strategy maintains more consistent scene geometry. In point cloud reconstruction presented, aggressive fixed compression leads to incomplete surfaces, whereas the data-adaptive method reconstructs more complete geometry and maintains more faithful structural boundaries.

4.4 Ablation Studies

In this section, we present a couple of important ablation studies regarding entropy-based offline calibration. We explore the potential of 1) integrating more visual representations to capture input complexity 2) implementing a continuous mapping between entropy and compute budget.

Data Augmentation. Motivated by recent work on visual abstraction [15], we explore whether augmenting raw-pixel entropy with simple visual-abstract representations could provide a stronger proxy for input complexity. Specifically, we construct auxiliary views of the input using standard image processing operators, including Otsu-based binarization and Canny edge detection. Shannon entropy is computed on each representation and normalized, and the resulting scores are aggregated via a simple summation to form an augmented complexity signal. However, the findings in Tab. 7 and Fig. 4 indicate that, although richer representations may capture more spatial or structural details, the added complexity in their

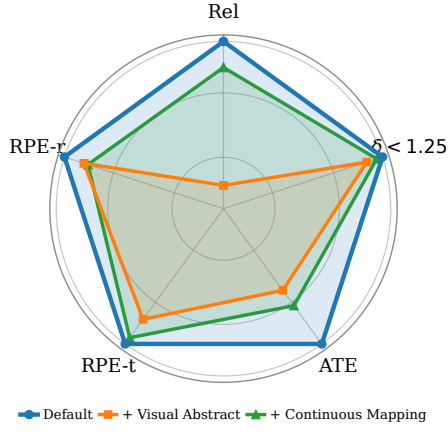

Fig. 4: Ablation of entropy calibration. All metrics are normalized to the default setting (radius = 1.0). Lower-is-better metrics are inverted, such that larger values uniformly denote better performance.

estimates does not translate into improved downstream performance in our scenario. In fact, simple first-order entropy statistics already offer an informative signal for adaptive compression.

Budget Mapping. To explore whether a fine-grained, continuous mapping from input entropy to retention ratio provide more effective adaptive compression than discrete buckets, we try to replace the discrete regime assignment with a *continuous* entropy to retention mapping. Given a normalized entropy score $s_{norm}(x) \in [0, 1]$, we define the retention ratio as

Table 7: Raw ablation results for entropy calibration on Sintel video-depth and ScanNet camera-distance evaluation. Lower is better for Rel, ATE, RPE trans, and RPE rot, while higher is better for $\delta < 1.25$.

		video-depth: Sintel		camera-dist: ScanNet		
Model	**GFLOPs**	Rel↓	δ<1.25↑	ATE↓	RPE trans↓	RPE rot↓
Original	155.00	0.2515	0.6348	0.0497	0.0165	0.4142
Augmented	155.00	0.2832	0.6266	0.0524	0.0169	0.4209
fine-grained	155.00	0.2567	0.6318	0.0516	0.0166	0.4223

$$\mathrm{r}(x) = r_{\min} + (r_{\max} - r_{\min})\,\sigma\big(\alpha\,(s_{norm}(x) - \beta)\big),$$

where $\sigma(\cdot)$ denotes the sigmoid function. The slope parameter α controls the sharpness of the transition, while the offset β is chosen to satisfy a global budget constraint on a calibration dataset:

$$\frac{1}{N}\sum_{i=1}^{N} \mathrm{r}(x_i) = r_{\mathrm{mid}}$$

Since $r(x)$ is a monotonic function, we simply solve for β via binary search. Nevertheless, finer granularity in budget allocation does not automatically translate into better accuracy efficiency tradeoffs; stability in regime assignment appears to be more important than continuity in mapping.

5 Conclusion

In this work, we propose a data-adaptive compression framework for 3D visual geometry models based on entropy-guided compression allocation. Instead of enforcing a uniform compression ratio across all inputs, our method dynamically adjusts retention according to input complexity via raw-pixel entropy. Extensive experiments across monocular depth estimation, video depth estimation, camera pose estimation, and point cloud reconstruction reveal that this simple yet principled strategy improves the inference speed of feedforward 3D models without retraining, implying that input-aware resource allocation is a promising direction for improving the efficiency of these large scale models.

References

1. Alonso, P., Li, T., Li, C.: Less is more: Efficient point cloud reconstruction via multi-head decoders (2025), https://arxiv.org/abs/2505.19057 1
2. Azinović, D., Martin-Brualla, R., Goldman, D.B., Nießner, M., Thies, J.: Neural rgb-d surface reconstruction (2022), https://arxiv.org/abs/2104.04532 12
3. Božič, A., Palafox, P., Thies, J., Dai, A., Nießner, M.: Transformerfusion: Monocular rgb scene reconstruction using transformers (2021), https://arxiv.org/abs/2107.02191 6, 10, 11
4. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes (2017), https://arxiv.org/abs/1702.04405 11
5. Ding, X., Sun, R., Zhang, Y., Yan, X., Zhou, Y., Huang, K., Fu, S., Xie, C., Zhu, Y.: Dipsvd: Dual-importance protected svd for efficient llm compression (2025), https://arxiv.org/abs/2506.20353 2, 4
6. Feng, W., Qin, H., Wu, M., Yang, C., Li, Y., Li, X., An, Z., Huang, L., Zhang, Y., Magno, M., Xu, Y.: Quantized visual geometry grounded transformer (2025), https://arxiv.org/abs/2509.21302 2, 4
7. Fu, X., Yin, W., Hu, M., Wang, K., Ma, Y., Tan, P., Shen, S., Lin, D., Long, X.: Geowizard: Unleashing the diffusion priors for 3d geometry estimation from a single image (2024), https://arxiv.org/abs/2403.12013 1
8. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. Int. J. Rob. Res. **32**(11), 1231–1237 (Sep 2013). https://doi.org/10.1177/0278364913491297, https://doi.org/10.1177/0278364913491297 6, 11
9. Jaderberg, M., Vedaldi, A., Zisserman, A.: Speeding up convolutional neural networks with low rank expansions (2014), https://arxiv.org/abs/1405.3866 2
10. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering (2023), https://arxiv.org/abs/2308.04079 1
11. Lee, J.J., Benes, B.: Rgb2point: 3d point cloud generation from single rgb images (2024), https://arxiv.org/abs/2407.14979 1
12. Li, W., Yang, Z., Han, W., Man, H., Wang, X., Fan, X.: Hyperbolic-constraint point cloud reconstruction from single rgb-d images (2024), https://arxiv.org/abs/2412.09055 1
13. Li, Z., Xia, M., Zhang, J., Hui, Z., Qin, H., Kong, L., Zhang, Y., Yang, X.: Adasvd: Adaptive singular value decomposition for large language models (2025), https://arxiv.org/abs/2502.01403 2, 4
14. Lin, J., Tang, J., Tang, H., Yang, S., Chen, W.M., Wang, W.C., Xiao, G., Dang, X., Gan, C., Han, S.: Awq: Activation-aware weight quantization for llm compression and acceleration (2024), https://arxiv.org/abs/2306.00978 4
15. Liu, D., Wang, Z., Ruan, M., Luo, F., Chen, C., Li, P., Liu, Y.: Thinking with visual abstract: Enhancing multimodal reasoning via visual abstraction (2025), https://arxiv.org/abs/2505.20164 13
16. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis (2020), https://arxiv.org/abs/2003.08934 1
17. Palazzolo, E., Behley, J., Lottes, P., Giguère, P., Stachniss, C.: Refusion: 3d reconstruction in dynamic environments for rgb-d cameras exploiting residuals (2019), https://arxiv.org/abs/1905.02082 6, 11
18. Shen, Y., Zhang, Z., Qu, Y., Zheng, X., Ji, J., Zhang, S., Cao, L.: Fastvggt: Training-free acceleration of visual geometry transformer (2025), https://arxiv.org/abs/2509.02560 2, 3, 4

19. Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., Fitzgibbon, A.: Scene coordinate regression forests for camera relocalization in rgb-d images. In: Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition. p. 2930–2937. CVPR '13, IEEE Computer Society, USA (2013). https://doi.org/10.1109/CVPR.2013.377, https://doi.org/10.1109/CVPR.2013.377 12
20. Shu, Z., Lin, C., Xie, T., Yin, W., Li, B., Pu, Z., Li, W., Yao, Y., Cao, X., Guo, X., Long, X.X.: Litevggt: Boosting vanilla vggt via geometry-aware cached token merging (2025), https://arxiv.org/abs/2512.04939 2, 3, 4
21. Sinha, S., Zhang, J.Y., Tagliasacchi, A., Gilitschenski, I., Lindell, D.B.: Sparsepose: Sparse-view camera pose regression and refinement (2022), https://arxiv.org/abs/2211.16991 1
22. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of rgb-d slam systems. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 573–580 (2012). https://doi.org/10.1109/IROS.2012.6385773 11
23. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer (2025), https://arxiv.org/abs/2503.11651 1, 3, 10, 11
24. Wang, Q., Zhang, Y., Holynski, A., Efros, A.A., Kanazawa, A.: Continuous 3d perception model with persistent state (2025), https://arxiv.org/abs/2501.12387 3
25. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: Dust3r: Geometric 3d vision made easy (2024), https://arxiv.org/abs/2312.14132 1
26. Wang, X., Alam, S., Wan, Z., Shen, H., Zhang, M.: Svd-llm v2: Optimizing singular value truncation for large language model compression (2025), https://arxiv.org/abs/2503.12340 2, 4
27. Wang, X., Zheng, Y., Wan, Z., Zhang, M.: Svd-llm: Truncation-aware singular value decomposition for large language model compression (2025), https://arxiv.org/abs/2403.07378 2, 4, 5, 9
28. Wang, Y., Zhou, J., Zhu, H., Chang, W., Zhou, Y., Li, Z., Chen, J., Pang, J., Shen, C., He, T.: π^3: Permutation-equivariant visual geometry learning (2025), https://arxiv.org/abs/2507.13347 1, 3, 11
29. Wang, Z., Xu, D.: Flashvggt: Efficient and scalable visual geometry transformers with compressed descriptor attention (2025), https://arxiv.org/abs/2512.01540 2, 3, 4
30. Wu, J., Wang, H., Shang, Y., Shah, M., Yan, Y.: Ptq4dit: Post-training quantization for diffusion transformers (2024), https://arxiv.org/abs/2405.16005 4
31. Xian, Q., Jiao, W., Cheng, H., van der Zwaag, B.J., Huang, Y.: T-graph: Enhancing sparse-view camera pose estimation by pairwise translation graph (2025), https://arxiv.org/abs/2505.01207 1
32. Xu, Z., Zhou, H., Peng, S., Lin, H., Guo, H., Shao, J., Yang, P., Yang, Q., Miao, S., He, X., Wang, Y., Wang, Y., Hu, R., Liao, Y., Zhou, X., Bao, H.: Towards depth foundation model: Recent trends in vision-based depth estimation (2025), https://arxiv.org/abs/2507.11540 1
33. Yang, J., Sax, A., Liang, K.J., Henaff, M., Tang, H., Cao, A., Chai, J., Meier, F., Feiszli, M.: Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass (2025), https://arxiv.org/abs/2501.13928 2, 3
34. Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., Zhao, H.: Depth anything: Unleashing the power of large-scale unlabeled data (2024), https://arxiv.org/abs/2401.10891 1

35. Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., Zhao, H.: Depth anything v2 (2024), https://arxiv.org/abs/2406.09414 1
36. Yeshwanth, C., Liu, Y.C., Nießner, M., Dai, A.: Scannet++: A high-fidelity dataset of 3d indoor scenes (2023), https://arxiv.org/abs/2308.11417 9
37. Yuan, Z., Shang, Y., Song, Y., Yang, D., Wu, Q., Yan, Y., Sun, G.: Asvd: Activation-aware singular value decomposition for compressing large language models (2025), https://arxiv.org/abs/2312.05821 2, 4
38. Yuan, Z., Xue, C., Chen, Y., Wu, Q., Sun, G.: Ptq4vit: Post-training quantization for vision transformers with twin uniform quantization (2024), https://arxiv.org/abs/2111.12293 4
39. Zhang, S., Wang, J., Xu, Y., Xue, N., Rupprecht, C., Zhou, X., Shen, Y., Wetzstein, G.: Flare: Feed-forward geometry, appearance and camera estimation from uncalibrated sparse views (2026), https://arxiv.org/abs/2502.12138 3