DiaStream: Distribution-Aware KV Cache for Streaming 3D Reconstruction

Dongjae Jeong Inha Lee Kyungdon Joo†
Ulsan National Institute of Science and Technology (UNIST), Republic of Korea
{jeong_dongjae,epsilon8854,kyungdon}@unist.ac.kr

Abstract

Streaming visual geometry transformers rely on key-value (KV) caches to process long image sequences, but effective cache management under a fixed memory budget remains challenging. Existing methods consider token importance and key-space diversity independently, overlooking the underlying distribution of cached keys and limiting their ability to retain tokens that are both informative and non-redundant. We observe that cached keys exhibit structured redundancy in the key space, while their statistical characteristics vary substantially across transformer layers, suggesting that token retention and layer-wise cache allocation should be jointly considered. Based on this insight, we propose DiaStream, a training-free distribution-aware KV cache management framework. DiaStream combines Important-Distinctive Caching (IDC), which preserves tokens that are both distinctive and reconstruction-relevant by using statistical leverage and attention-based importance, with Key Statistics-Aware Budget Allocation (KABA), which dynamically allocates the global cache budget according to layer-wise key statistics. Experiments on multiple streaming 3D reconstruction benchmarks show that DiaStream consistently outperforms existing memory-bounded methods in 3D reconstruction and camera pose estimation while requiring at most 30% of the KV cache budget used by previous methods.

1. Introduction

Recent geometric foundation models, such as DUSt3R [25] and VGGT [22], have transformed multi-view 3D reconstruction by directly estimating camera parameters and dense scene geometry from multiple images in a single forward pass, eliminating the need for iterative geometric optimization. Building on this progress, recent efforts have focused on extending these models from fixed-size multi-view reconstruction to long image sequences for streaming visual geometry reconstruction. To this end, StreamVGGT [33] and STream3R [7] replace global attention with temporal causal attention and maintain a key-value (KV) cache of historical tokens, enabling online visual geometry reconstruction over long image sequences.

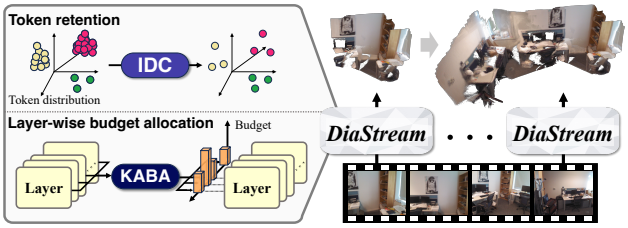

Figure 1. **Overview of *DiaStream*.** *DiaStream* combines IDC for token retention and KABA for layer-wise cache allocation to achieve memory-efficient streaming visual geometry reconstruction under a limited cache budget.

While temporal causal attention together with the KV cache enables efficient online visual geometry reconstruction over long image sequences, the KV cache grows linearly with the input sequence length. As each incoming frame contributes additional KV tokens, both memory consumption and inference latency increase accordingly. To keep these costs manageable, existing streaming visual geometry models typically impose a fixed cache budget, maintaining only a subset of cached tokens.

The key question then becomes which tokens should be retained to preserve an informative scene representation over long input streams. Recent methods have addressed this problem using retention criteria based on accumulated attention weights [12], key-space diversity [31], and spatio-temporal cache compression [24]. However, constructing a compact yet informative KV cache remains challenging because existing methods often focus on either token importance or key-space diversity without jointly accounting for both. Consequently, importance-based strategies may retain redundant tokens, whereas diversity-based strategies may retain less informative ones. Moreover, existing diversity measures may not fully capture the underlying structure of the cached-key distribution.

To better understand the limitations of existing methods, we analyze the KV cache and make two key observations. First, redundant cached keys are distributed across multiple

dominant directions in key space rather than concentrated around a single global centroid, as nearby views capture overlapping scene content and produce similar key representations. As illustrated in Fig. 2, keys associated with common scene elements, such as floor and table, occupy partially overlapping regions of the PCA-projected space. Consequently, retention strategies that summarize the key distribution using a single global representation may fail to capture redundancy among tokens that lie far from that representation. Second, key characteristics vary substantially across transformer layers, as reflected in both the dispersion of the key space and the key magnitude. This layer-wise heterogeneity indicates that the cache capacity required to preserve scene information differs across layers, making a uniform layer-wise cache allocation inefficient. These observations suggest that an effective cache retention strategy should account for directional redundancy within each layer and allocate the global cache budget across layers according to their cache statistics.

Motivated by these observations, we propose *distribution-aware KV cache for streaming 3D reconstruction* (*DiaStream*), a training-free KV cache management framework that maintains a compact scene representation by exploiting the cached-key distribution (see Fig. 1). *DiaStream* consists of two complementary components that operate at the token and layer levels. Important-Distinctive Caching (IDC) evaluates each cached token using statistical leverage scores that measure its distinctiveness relative to the directional support of the cached-key distribution, with accumulated attention weights. It thereby retains tokens that are both distinctive within the cache and important for subsequent reconstruction. Key Statistics-Aware Budget Allocation (KABA) dynamically allocates the global cache budget across layers based on the distribution of leverage scores and the magnitude of raw key representations, thereby accounting for layer-wise heterogeneity. While IDC determines which tokens to retain within each layer, KABA determines how much cache capacity each layer receives. Together, these components enable *DiaStream* to preserve scene information with a compact KV cache. Extensive experiments show that *DiaStream* achieves state-of-the-art performance among memory-bounded streaming methods in both 3D reconstruction and camera pose estimation, while using a smaller KV cache budget than prior methods.

Our contributions are summarized as follows:

- We introduce *DiaStream*, a distribution-aware KV cache management framework inspired by the observation that cached keys exhibit structured redundancy and layer-wise statistical heterogeneity.
- We propose Important-Distinctive Caching (IDC), a token retention strategy that maintains a compact KV cache by reflecting both the distributional structure of cached keys and reconstruction-relevant importance.
- We introduce Key Statistics-Aware Budget Allocation (KABA), a layer-level management strategy that adaptively allocates cache capacity according to the heterogeneous representation characteristics and key statistics of each layer.

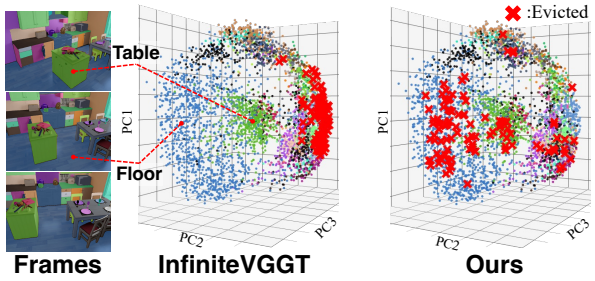

Figure 2. **Comparison of evicted tokens in the key space distribution.** Red crosses denote evicted tokens projected onto the top three principal components using PCA. InfiniteVGGT shows biased eviction concentrated in limited regions, whereas *DiaStream* selects eviction candidates across diverse regions by considering the underlying distribution.

2. Related work

Image-based 3D reconstruction. Classical SLAM and online dense reconstruction systems estimate camera motion and scene geometry from an image stream using explicit geometric and photometric constraints together with iterative optimization [2, 13, 14]. Recent feed-forward visual geometry models regress geometric quantities directly from uncalibrated images. DUSt3R [25] predicts dense, view-consistent pointmaps from image pairs, while MASt3R [8] augments this formulation with dense local features for 3D-aware image matching. Fast3R [30] and VGGT [22] further extend feed-forward reconstruction to multiple views in a single forward pass. In particular, VGGT jointly estimates camera parameters and dense geometric attributes through alternating frame-wise and global attention. To improve efficiency, subsequent studies have explored token merging [17, 26], chunk-wise inference [3, 29], or SLAM-style sequential processing [11]. While these approaches substantially improve scalability, their reliance on global attention over all input views still limits efficient processing of arbitrarily long image streams.

Streaming visual geometry. Recent methods extend feed-forward visual geometry models to streaming settings by maintaining persistent scene representations through recurrent states, spatial memories, or KV caches. Specifically, Spann3R [21] employs an external spatial memory to predict pointmaps in a common coordinate system, CUT3R [23] recurrently updates a persistent latent state,

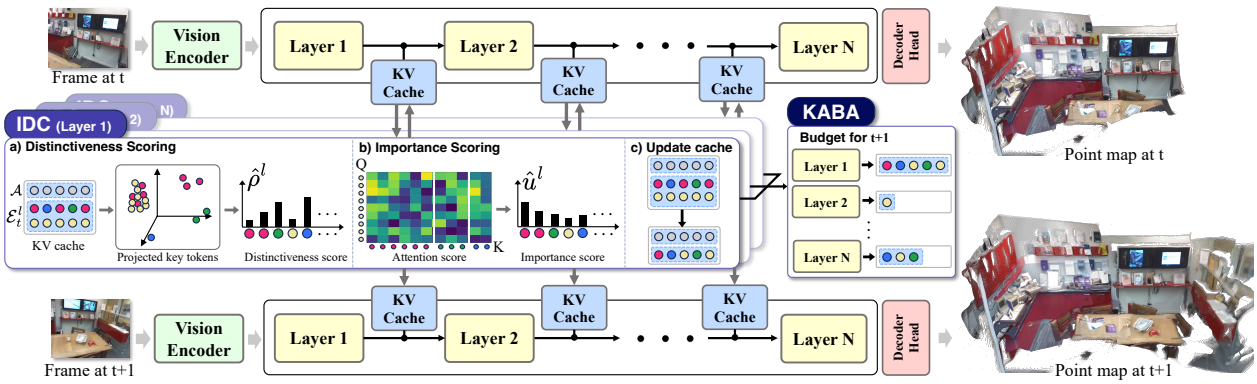

Figure 3. **Pipeline of *DiaStream*.** For each incoming frame, the corresponding KV representations are first appended to the layer-wise KV cache. IDC retains distinctive and important KV tokens, while KABA dynamically allocates the global cache budget across transformer layers. The resulting compact KV cache is then used for streaming visual geometry reconstruction.

and Point3R [27] maintains an explicit spatial pointer memory associated with 3D positions. These methods instantiate memory through recurrent or spatially structured modules. In particular, StreamVGGT [33] adopts temporal causal attention and maintains a KV cache of preceding frames. By reusing cached states, it avoids recomputing historical representations, enabling efficient incremental 3D geometry estimation.

Despite these advances, the KV cache and associated attention cost still grow linearly with sequence length, motivating explicit cache management under a fixed memory budget.

KV cache management. Fixed-budget KV cache management has been extensively studied in long-context language models, where informative tokens are preserved under limited memory budgets using attention- or observation-based retention criteria [4, 9, 28, 32]. Recent streaming visual geometry methods adapt these ideas to visual geometry reconstruction through task-specific retention policies and memory organizations. Importance-based methods retain tokens according to accumulated attention, such as Evict3R [12] and STAC [24], while diversity-based methods such as InfiniteVGGT [31] preserve tokens based on key-space diversity. Other approaches, such as OVGGT [10], further combine structured memory organization with token selection.

Unlike these methods, *DiaStream* jointly models the second-order distributional structure of cached keys and layer-wise heterogeneity to construct a compact yet informative KV cache under a fixed memory budget.

3. Method

In this section, we propose *DiaStream*, a distribution-aware KV cache management framework for streaming visual geometry transformers that retains compact yet informative KV tokens under a fixed cache budget. We first provide an overview of the framework, formulate memory-bounded streaming KV cache management, discuss the limitations of existing cache management methods, and finally present the proposed token- and layer-level cache management modules.

3.1. Overview

As illustrated in Fig. 3, *DiaStream* performs online visual geometry reconstruction by updating the transformer KV cache through two core components: Important-Distinctive Caching (IDC), which determines which cached tokens to retain, and Key Statistics-Aware Budget Allocation (KABA), which dynamically allocates the global cache budget across transformer layers.

For each incoming frame, the corresponding KV tokens are first appended to the existing cache at every transformer layer. IDC then evaluates each cached token according to its key-space distinctiveness and attention-based importance, retaining the highest-scoring tokens within the allocated layer budget while preserving the first-frame reference tokens. KABA dynamically allocates the global cache budget across transformer layers using layer-wise cached-key statistics. The resulting compact KV cache is then used by the streaming visual geometry transformer to estimate camera parameters and dense scene geometry.

3.2. Streaming KV Cache

Streaming visual geometry transformers [7, 33] process image sequences incrementally by utilizing a layer-wise KV cache under temporal causal attention. At each transformer layer, the KV cache stores the key and value tokens of previously processed frames, allowing subsequent frames to reuse historical KV tokens without recomputing them. When the t-th frame arrives, the KV cache at layer l is up-

dated by appending the newly generated key-value pairs:

$$\mathcal{M}_t^l = \mathcal{M}_{t-1}^l \cup \left(\mathbf{K}_t^l, \mathbf{V}_t^l\right), \tag{1}$$

where $\mathcal{M}_t^l$ denotes the KV cache at layer l, and $\mathbf{K}_t^l$ and $\mathbf{V}_t^l$ denote the key and value representations generated from the t-th frame.

While Eq. (1) enables efficient streaming inference by reusing historical KV tokens, the KV cache grows linearly with the sequence length, making it impractical to retain all historical tokens for image streams. To satisfy a fixed memory budget, the cache must be compacted after each update by retaining only a subset of the cached tokens:

$$\hat{\mathcal{M}}_t^l = \Phi(\mathcal{M}_t^l), \quad s.t. \sum_{l=1}^{L} |\hat{\mathcal{M}}_t^l| \leq B, \tag{2}$$

where $\hat{\mathcal{M}}_t^l$ denotes the retained cache at layer l, $\Phi(\cdot)$ denotes a cache retention function that selects a compact subset of cached KV tokens, L is the number of transformer layers, $|\cdot|$ denotes the number of cached KV token pairs, and B denotes the total KV cache budget shared across all transformer layers. Let b_t^l denote the layer-wise cache budget allocated to layer l at time step t, satisfying $\sum_{l=1}^{L} b_t^l = B$.

Existing cache management methods define the function $\Phi(\cdot)$ using different token retention criteria. Importance-based methods prioritize tokens that receive high accumulated attention, whereas diversity-based methods retain tokens that appear distinct in the key space. However, these methods overlook two important characteristics of KV cache. First, they typically rely on either token importance or key-space diversity as the primary retention signal, failing to prioritize tokens that are both informative and non-redundant with respect to the underlying cached-key distribution. Second, they often overlook layer-wise variations in cached-key statistics across transformer layers by allocating the cache budget uniformly across layers. To better understand these limitations, we analyze the cached keys and identify two observations: (1) redundancy across multiple dominant key-space directions and (2) heterogeneous cached-key statistics across transformer layers. These observations motivate the proposed token- and layer-level cache management modules presented in the following sections.

3.3. Important-Distinctive Caching

Motivated by the first observation, IDC maintains a compact KV cache by ranking KV tokens according to key-space distinctiveness and attention-based importance. In streaming image sequences, redundant cached keys are distributed across multiple dominant directions in key space due to small viewpoint changes. Consequently, first-order measures based on the distance from a single global centroid may fail to capture the distributional structure of the cached keys. We therefore measure key-space distinctiveness using statistical leverage scores [5], which quantify each token's contribution to the principal subspace of the cached-key distribution. Since key-space distinctiveness alone does not indicate whether a token contributes to subsequent reconstruction, IDC additionally incorporates attention-based importance into the cache retention criterion.

Key-space distinctiveness scoring. To evaluate the key-space distinctiveness of the tokens, we first define the evictable candidate token set while preserving the anchor tokens inherited from the first frame. Let $\mathcal{A}^l$ denote the anchor token set retained throughout the sequence, and let $\mathcal{E}_t^l$ denote the evictable candidate token set consisting of all cached non-anchor tokens together with the current-frame tokens. Before token eviction, the KV cache at layer l is defined as

$$\tilde{\mathcal{M}}_t^l = \mathcal{A}^l \cup \mathcal{E}_t^l. \tag{3}$$

Although the anchor tokens are included in the layer-wise cache budget b_t^l, they are excluded from token scoring. To make the distinctiveness score depend on key direction rather than magnitude, we normalize the key features of each attention head and concatenate them into a layer-level key representation. We then stack these representations into the layer-level key matrix denoted by $\hat{\mathbf{K}}_t^l \in \mathbb{R}^{|\mathcal{E}_t^l| \times d}$, where d denotes the layer-level feature dimension. To efficiently compute the statistical leverage scores, we project $\hat{\mathbf{K}}_t^l$ into a lower-dimensional subspace using a fixed Gaussian random projection matrix. This projection substantially reduces the computational cost while enabling an efficient approximation of the statistical leverage scores. The projected key matrix is computed as

$$\mathbf{Z}_t^l = \hat{\mathbf{K}}_t^l \mathbf{R}^l, \tag{4}$$

where $\mathbf{R}^l \in \mathbb{R}^{d \times r}$ is a fixed Gaussian random projection matrix with $r \ll d$.

The statistical leverage scores are obtained via the compact singular value decomposition (SVD) of $\mathbf{Z}_t^l$:

$$\mathbf{Z}_t^l = \mathbf{U}_t^l \mathbf{\Sigma}_t^l (\mathbf{W}_t^l)^\top, \qquad \rho_i^l = \|(\mathbf{U}_t^l)_{i,:}\|_2^2, \tag{5}$$

where ρ_i^l denotes the statistical leverage score of the i-th candidate token and $\|\cdot\|_2^2$ denotes the squared L2 norm. A larger leverage score indicates that the corresponding token contributes more strongly to the principal subspace of the cached-key distribution and is therefore considered more distinctive.

To better understand how leverage scores account for directional redundancy, Eq. (5) can be equivalently expressed using the Gram matrix:

$$\mathbf{G}_t^l = (\mathbf{Z}_t^l)^\top \mathbf{Z}_t^l, \qquad \rho_i^l = \mathbf{z}_i^l (\mathbf{G}_t^l)^\dagger (\mathbf{z}_i^l)^\top, \tag{6}$$

where $\mathbf{z}_i^l \in \mathbb{R}^{1\times r}$ denotes the i-th row of $\mathbf{Z}_t^l$ and $\dagger$ denotes Moore–Penrose pseudoinverse. The pseudoinverse Gram matrix assigns smaller weights to key-space directions that are strongly supported by many cached tokens and larger weights to weakly supported directions. Consequently, tokens aligned with redundant directions receive lower leverage scores, whereas tokens contributing to under-represented directions receive higher scores. This allows IDC to identify distinctive tokens while suppressing redundancy across multiple key-space directions.

Attention-based importance scoring. To evaluate how candidate tokens contribute to reconstruction, we estimate token importance using the attention weights received from subsequent frames. While statistical leverage captures token distinctiveness, the attention-based score reflects how strongly each token is used during streaming inference.

Since the number of cached tokens varies over time, we normalize the attention weights by the size of the post-eviction KV cache so that the importance scores remain comparable across different cache sizes:

$$\bar{\alpha}_{t,i}^l = |\hat{\mathcal{M}}_t^l| \alpha_{t,i}^l, \tag{7}$$

where $\alpha_{t,i}^l$ denotes the attention probability assigned to the i-th candidate token by the current-frame query tokens, averaged over query tokens and attention heads.

To obtain a robust importance score from multiple views, we compute it as a decay-weighted average:

$$u_i^l = \frac{\sum_{j=1}^{N_i} \gamma^{N_i-j} \bar{\alpha}_{t'+j,i}^l}{\sum_{j=1}^{N_i} \gamma^{N_i-j}}, \tag{8}$$

where $\gamma \in (0,1)$ is the decay factor, t' is the source frame index of the i-th candidate token, and N_i is the number of subsequent frames processed after t', capped at T. The importance score is updated only during the first T frames after cache insertion, after which it remains fixed, allowing the importance score to better reflect each token's long-term utility for reconstruction.

Combined score and retention. Finally, we combine the key-space distinctiveness score and the attention-based importance score into a unified ranking criterion for cache retention. Since the two scores have different numerical ranges, we first compute their mean-normalized scores, denoted by $\hat{\rho}_i^l$ and $\hat{u}_i^l$, respectively. The final retention score is then computed as

$$s_i^l = \hat{\rho}_i^l + \hat{u}_i^l. \tag{9}$$

IDC retains the $b_t^l - |\mathcal{A}^l|$ candidate tokens with the largest s_i^l, while always preserving the anchor token set A^l. If $|\tilde{\mathcal{M}}_t^l| \le b_t^l$, no token eviction is performed.

3.4. Key Statistics-Aware Budget Allocation

Motivated by our second observation that cached-key statistics vary substantially across transformer layers, KABA dynamically distributes a fixed global cache budget according to the representation demand of each layer. In particular, some layers exhibit leverage scores concentrated on a small subset of tokens, whereas others distribute them across a broader set, and the average raw key magnitude also varies across layers. A uniform layer-wise allocation ignores this heterogeneity and may therefore allocate cache capacity inefficiently.

To address this, KABA characterizes each layer from two complementary perspectives: the diversity of information and the magnitude of its raw keys. Based on these statistics, KABA estimates the representation demand of each layer and dynamically allocates the global cache budget.

Leverage-score participation ratio. The first statistic measures how broadly the leverage scores are distributed across the candidate tokens within each transformer layer. To quantify this distribution, we compute the participation ratio [15] over the leverage scores, yielding the leverage-score participation ratio:

$$\psi_t^l = \frac{\left(\sum_{i\in\mathcal{E}_t^l} \rho_i^l\right)^2}{\sum_{i\in\mathcal{E}_t^l} (\rho_i^l)^2}. \tag{10}$$

This estimates the effective number of tokens contributing to the leverage-score distribution. A smaller value of ψ_t^l indicates that the leverage scores are concentrated on only a few candidate tokens, whereas a larger value indicates that they are distributed more broadly across many tokens. The resulting ψ_t^l is used as a layer-wise weight for allocating the global cache budget in KABA.

Key-magnitude scale. The second statistic measures the overall magnitude of raw key vectors within each transformer layer. While the leverage participation ratio captures how leverage scores are distributed across candidate tokens, it does not reflect the absolute scale of the underlying key vectors. Previous studies have shown that larger key magnitudes tend to be associated with stronger attention-related token influence in KV cache compression [4]. Motivated by this observation, we incorporate key magnitude as a complementary statistic for layer-wise cache budget allocation:

$$\mu_t^l = \frac{1}{|\mathcal{E}_t^l|} \sum_{i\in\mathcal{E}_t^l} \|\mathbf{k}_i^l\|_2, \tag{11}$$

where $\|\cdot\|_2$ denotes L2 norm and $\mathbf{k}_i^l$ is the key vector of the i-th candidate token. A larger value indicates a larger

scale of key representations and therefore suggests a greater representation demand for the corresponding layer.

Layer-wise budget allocation. The two statistics capture complementary characteristics of each transformer layer. Layers require a larger cache budget only when they exhibit both broadly distributed leverage scores and large key magnitudes. We thus combine them into a single allocation weight:

$$w_t^l = \left(\psi_t^l \mu_t^l\right)^\eta, \tag{12}$$

where $\eta > 0$ controls the sharpness of the allocation. The layer weights are normalized to distribute the fixed global cache budget across transformer layers:

$$b_{t+1}^l = \left\lfloor B \frac{w_t^l}{\sum_{\ell=1}^{L} w_t^\ell} \right\rfloor, \tag{13}$$

where $\lfloor \cdot \rfloor$ denotes the floor function. Any remaining token slots are assigned to layers according to the descending order of the fractional parts of their unrounded allocations, ensuring that the final layer budgets sum to B. The resulting layer budgets are applied to cache retention for the next frame.

4. Experiments

4.1. Experimental Setup

Implementation details. We extend the streaming visual geometry transformer [33] to perform online 3D reconstruction and camera pose estimation under a bounded KV cache budget without any additional training. Unless otherwise stated, we use a cache budget $B = 60$K, projected dimension $r = 256$, attention window $T = 5$, and allocation exponent $\eta = 0.7$. We use a fixed random seed for all experiments to ensure reproducibility. We reuse the CUDA column-sum operator from STAC [24] to efficiently aggregate attention probabilities. All experiments are conducted on a single NVIDIA RTX A6000 GPU.

Evaluation settings. We evaluate 3D reconstruction on 7-Scenes [18], NRGBD [1], and ETH3D [16], and camera pose estimation on 7-Scenes, Replica [19], and TUM-Dynamics [20]. Following OVGGT [10], all methods use keyframes sampled with a stride of 2 for all datasets except ETH3D, where all input frames are processed due to the short sequence length. To evaluate performance across different streaming horizons, we consider two evaluation settings: short-sequence and full-sequence. For short-sequence evaluation, we use the first 100 input frames. ETH3D is excluded from this setting because its sequences contain fewer than 100 frames. For full-sequence evaluation, all methods process the entire input sequence, with sequence lengths ranging from 198 to 1,000 frames across the other datasets. We report Chamfer Distance (CD) and F1 score for 3D reconstruction. CD measures the geometric discrepancy between the reconstructed and ground-truth point clouds, while the F1 score evaluates reconstruction accuracy by combining precision and recall under a predefined distance threshold. The F1 score is computed using a threshold of $5\,\mathrm{cm}$ for 7-Scenes and NRGBD and a $25\,\mathrm{cm}$ threshold for ETH3D. For camera pose estimation, we report Absolute Trajectory Error (ATE) in meters, computed using the evo toolkit [6] with Umeyama alignment applied between estimated and ground-truth trajectories..

Baselines. We compare *DiaStream* with two groups of methods. The first group includes recent streamable feed-forward reconstruction methods, including CUT3R [23], Point3R [27], Stream3R [7], and the VGGT-based StreamVGGT [33]. The second group comprises KV cache management methods for bounded-memory streaming, including InfiniteVGGT [31], STAC [24], and OVGGT [10].

4.2. 3D Reconstruction

Tab. 1 summarizes the quantitative reconstruction results. *DiaStream* delivers the strongest reconstruction performance across the evaluated datasets under both the short- and full-sequence settings. Compared with OVGGT, *DiaStream* achieves higher reconstruction accuracy across all evaluated datasets while using only a 60K-token cache, approximately 70% smaller than OVGGT's 200K-token budget.

In contrast, StreamVGGT and STream3R maintain unbounded KV caches and therefore cannot complete full-sequence evaluation due to GPU memory exhaustion. Furthermore, even with only a 30K-token cache, *DiaStream* consistently outperforms all existing memory-bounded methods, demonstrating that its performance does not rely on a large cache budget. These results demonstrate that the proposed cache management strategy effectively preserves informative scene representations, maintaining stable reconstruction quality even as the sequence length increases under strict memory constraints.

Qualitative results in Fig. 4 further support the quantitative findings. Compared with existing cache-based methods, *DiaStream* produces cleaner and more structurally consistent reconstructions on both 7-Scenes and NRGBD. In the *Pumpkin* scene, *DiaStream* better preserves cabinet boundaries while suppressing surface artifacts. In the *White room* scene, it more faithfully reconstructs the table and thin window structures. These observations are consistent with the quantitative improvements reported in Tab. 1.

Method	Budget	Short sequence				Full sequence						FPS↑
		7-Scenes		NRGBD		7-Scenes		NRGBD		ETH3D		
		CD↓	$F1\uparrow$	CD↓	$F1\uparrow$	CD↓	$F1\uparrow$	CD↓	$F1\uparrow$	CD↓	$F1\uparrow$	
CUT3R	N/A	0.027	0.851	0.026	0.851	0.135	0.431	0.247	0.322	0.785	0.536	25.38
Point3R		0.023	0.878	0.047	0.747	0.055	0.720	0.103	0.531	1.849	0.432	7.93
Stream3R		0.026	0.932	0.010	0.984	OOM	OOM	OOM	OOM	0.349	0.712	4.18
StreamVGGT		0.029	0.850	0.019	0.933	OOM	OOM	OOM	OOM	0.521	0.564	5.49
STAC	225K	0.030	0.840	0.026	0.882	0.041	0.780	0.080	0.589	0.501	0.554	**13.78**
InfiniteVGGT	1.2M	0.034	0.816	0.025	0.881	0.036	0.817	0.056	0.701	0.522	0.565	7.98
OVGGT	200K	0.022	0.899	0.017	0.935	0.028	0.867	0.042	0.766	0.503	0.562	11.65
DiaStream-S	30K	**0.016**	0.945	0.015	0.962	0.022	0.912	**0.025**	**0.894**	**0.416**	**0.643**	11.11
DiaStream	60K	**0.016**	**0.947**	**0.013**	**0.966**	**0.020**	**0.923**	0.026	0.877	0.417	0.631	10.83

Table 1. **3D reconstruction results on short and full sequences**. We report Chamfer distance (CD) and F1 score on short and full sequences, and FPS on short sequences. The best and second-best results among memory-bounded methods are highlighted in bold and underlined, respectively. OOM denotes out-of-memory, N/A indicates an unbounded cache and *DiaStream-S* denotes *DiaStream* with a small budget of 30K.

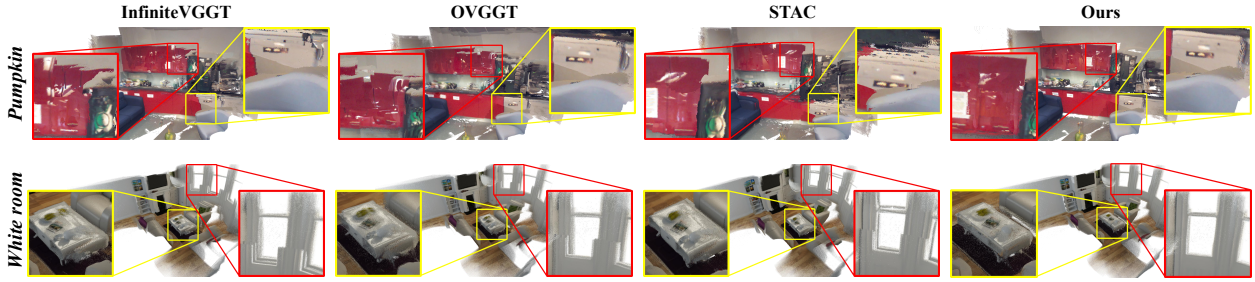

Figure 4. **Qualitative results on the *Pumpkin* sequence from 7-Scenes and the *White room* sequence from NRGBD.** Red and yellow boxes indicate regions enlarged for closer inspection.

Method	7-Scenes	Replica	TUM
	Short / Full	Short / Full	Short / Full
CUT3R	0.071 / 0.566	0.185 / 0.817	0.051 / 0.135
Point3R	0.095 / 0.348	0.128 / 0.582	0.074 / 0.156
Stream3R	0.034 / OOM	0.020 / OOM	0.024 / OOM
StreamVGGT	0.041 / OOM	0.020 / OOM	0.025 / OOM
STAC	0.042 / 0.129	0.024 / OOM	0.026 / 0.046
InfiniteVGGT	0.043 / 0.127	**0.020** / 0.244	0.028 / 0.054
OVGGT	0.036 / 0.152	0.028 / 0.357	0.027 / 0.067
DiaStream-S	0.035 / 0.127	0.022 / 0.176	**0.021** / 0.046
DiaStream	**0.034 / 0.104**	0.022 / **0.135**	0.023 / **0.039**

Table 2. **Trajectory error on short and full sequences**. We report ATE ↓ in meters on short and full sequences. Each entry is reported as *Short / Full* sequences, respectively.

4.3. Pose Estimation

Tab. 2 reports the quantitative camera pose estimation results. *DiaStream* achieves the lowest trajectory error among memory-bounded methods on all full-sequence benchmarks and remains competitive in the short-sequence setting, with more pronounced gains over longer horizons where drift accumulates. Despite using only a 60K-token cache, *DiaStream* achieves the lowest trajectory error among memory-bounded methods, requiring approximately 70% fewer cache budget than STAC and OVGGT and 95% fewer than InfiniteVGGT.

Long image streams inevitably accumulate localization errors, making effective long-term memory essential for stable camera tracking. Even with only a 30K-token cache, *DiaStream* consistently outperforms OVGGT across all evaluated datasets and surpasses all existing memory-bounded methods on Replica. Furthermore, *DiaStream* also achieves the best overall performance in the short-sequence setting, indicating that its benefits are not limited to long-horizon tracking. In contrast, StreamVGGT and Stream3R cannot complete full-sequence evaluation because their unbounded KV caches exhaust GPU memory. These results indicate that the proposed cache management strategy effectively preserves pose-relevant historical information while limiting long-term trajectory drift under strict memory constraints.

Method	Budget	7-Scenes		NRGBD	
		CD↓	F1↑	CD↓	F1↑
InfiniteVGGT	1.2M	0.036	0.817	0.056	0.701
	60K	0.102	0.627	0.096	0.559
OVGGT	200K	0.028	0.867	0.042	0.766
	60K	0.066	0.745	0.040	0.770
Ours	200K	0.022	0.900	0.029	0.828
	60K	**0.020**	**0.923**	0.026	0.877
	30K	0.022	0.912	**0.025**	**0.894**

Table 3. **Ablation study of cache budget**. We evaluate reconstruction performance under varying token budgets.

IDC	KABA	CD ↓	F1 ↑
✓	×	0.021	0.905
×	✓	0.029	0.850
✓	✓	**0.016**	**0.947**

Table 4. **Ablation study of the proposed components**. We evaluate each proposed component on the short sequences of 7-Scenes using a 60K-token budget.

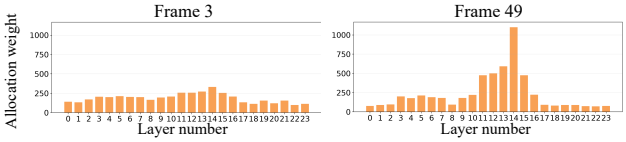

Figure 5. **Visualization of the layer-wise weights computed by KABA on 7-Scenes.**

4.4. Ablation study

We conduct ablation studies to analyze *DiaStream*.

Analysis of adaptive cache allocation. Fig. 5 visualizes the layer-wise weights w_t^l computed by KABA. Starting from a nearly uniform distribution, the weights gradually become more differentiated, indicating that KABA adaptively redistributes the cache budget across transformer layers as the sequence progresses.

Robustness under varying cache budgets. Tab. 3 demonstrates the robustness of *DiaStream* under varying KV cache budgets. Whereas competing methods show larger degradation on 7-Scenes and under the most constrained cache settings, *DiaStream* maintains stable reconstruction quality even with limited cache capacity. This robustness demonstrates the benefit of combining IDC for informative token retention with KABA for adaptive layer-wise cache allocation.

Effectiveness of the proposed components. Tab. 4 evaluates the individual and combined effects of IDC and KABA. Removing either component degrades performance, while combining both yields the best results, demonstrating that the two components are complementary. Among the individual components, IDC provides a larger performance gain than KABA, suggesting that jointly considering token distinctiveness and importance is particularly effective for filtering out redundant and uninformative tokens while retaining informative scene representations.

5. Conclusion

We presented *DiaStream*, a training-free distribution-aware KV cache management framework for memory-bounded streaming visual geometry transformers. Motivated by the observations that cached keys exhibit structured redundancy and layer-wise statistical heterogeneity, *DiaStream* jointly addresses token retention and layer-wise cache allocation through Important-Distinctive Caching (IDC) and Key Statistics-Aware Budget Allocation (KABA). Extensive experiments on multiple streaming 3D reconstruction benchmarks demonstrate that *DiaStream* consistently improves both 3D reconstruction and camera pose estimation while requiring substantially less memory than existing memory-bounded methods. Although *DiaStream* effectively exploits the statistical properties of cached keys for cache management, incorporating geometric cues, such as scene structure or camera motion, into the cache management process remains a promising direction for future work.

References

[1] Dejan Azinović, Ricardo Martin-Brualla, Dan B. Goldman, Matthias Nießner, and Justus Thies. Neural RGB-D surface reconstruction. In *CVPR*, 2022. 6

[2] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. BundleFusion: Real-time globally consistent 3D reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics*, 36(3): 1–18, 2017. Article 24. 2

[3] Kai Deng, Zexin Ti, Jiawei Xu, Jian Yang, and Jin Xie. VGGT-Long: Chunk it, loop it, align it – pushing VGGT's limits on kilometer-scale long RGB sequences. In *IEEE ICRA*, 2026. 2

[4] Alessio Devoto, Yu Zhao, Simone Scardapane, and Pasquale Minervini. A simple and effective l_2 norm-based strategy for KV cache compression. In *EMNLP*, 2024. 3, 5

[5] Petros Drineas, Malik Magdon-Ismail, Michael W. Mahoney, and David P. Woodruff. Fast approximation of matrix coherence and statistical leverage. *Journal of Machine Learning Research*, 13(111):3475–3506, 2012. 4

[6] Michael Grupp. evo: Python package for the evaluation of odometry and slam. https://github.com/MichaelGrupp/evo, 2017. 6

[7] Yushi Lan, Yihang Luo, Fangzhou Hong, Shangchen Zhou, Honghua Chen, Zhaoyang Lyu, Shuai Yang, Bo Dai, Chen Change Loy, and Xingang Pan. STream3R: Scalable sequential 3D reconstruction with causal transformer. In *ICLR*, 2026. 1, 3, 6

[8] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3D with MASt3R. In *ECCV*, 2024. 2

[9] Yijun Liu, Yixuan Wang, Yuzhuang Xu, Shiyu Ji, Yang Xu, Qingfu Zhu, and Wanxiang Che. Judge q: Trainable queries for optimized information retention in KV cache eviction. In *AAAI*, 2026. 3

[10] Si-Yu Lu, Po-Ting Chen, Hui-Che Hsu, Sin-Ye Jhong, Wen-Huang Cheng, and Yung-Yao Chen. OVGGT: $O(1)$ constant-cost streaming visual geometry transformer, 2026. 3, 6

[11] Dominic Maggio, Hyungtae Lim, and Luca Carlone. VGGT-SLAM: Dense RGB SLAM optimized on the SL(4) manifold. In *Advances in Neural Information Processing Systems*, 2025. 2

[12] Soroush Mahdi, Fardin Ayar, Ehsan Javanmardi, Manabu Tsukada, and Mahdi Javanmardi. Evict3R: Training-free token eviction for memory-bounded streaming visual geometry transformers, 2025. 1, 3

[13] Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. 2

[14] Richard A. Newcombe, Steven J. Lovegrove, and Andrew J. Davison. DTAM: Dense tracking and mapping in real-time. In *ICCV*, 2011. 2

[15] Stefano Recanatesi, Serena Bradde, Vijay Balasubramanian, Nicholas A. Steinmetz, and Eric Shea-Brown. A scale-dependent measure of system dimensionality. *Patterns*, 3(8): 100555, 2022. 5

[16] Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *CVPR*, 2017. 6

[17] You Shen, Zhipeng Zhang, Yansong Qu, Xiawu Zheng, Jiayi Ji, Shengchuan Zhang, and Liujuan Cao. FastVGGT: Fast visual geometry transformer. In *ICLR*, 2026. 2

[18] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew W. Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In *CVPR*, 2013. 6

[19] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces, 2019. 6

[20] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *IEEE IROS*, 2012. 6

[21] Hengyi Wang and Lourdes Agapito. 3D reconstruction with spatial memory. In *3DV*, 2025. 2

[22] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *CVPR*, 2025. 1, 2

[23] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3D perception model with persistent state. In *CVPR*, 2025. 2, 6

[24] Runze Wang, Yuxuan Song, Youcheng Cai, and Ligang Liu. STAC: Plug-and-play spatio-temporal aware cache compression for streaming 3D reconstruction. In *CVPR*, 2026. 1, 3, 6

[25] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jérôme Revaud. DUSt3R: Geometric 3D vision made easy. In *CVPR*, 2024. 1, 2

[26] Weitian Wang, Lukas Meiner, Rai Shubham, Cecilia De La Parra, and Akash Kumar. HTTM: Head-wise temporal token merging for faster VGGT. In *CVPR*, 2026. 2

[27] Yuqi Wu, Wenzhao Zheng, Jie Zhou, and Jiwen Lu. Point3R: Streaming 3D reconstruction with explicit spatial pointer memory. In *Advances in Neural Information Processing Systems*, 2025. 3, 6

[28] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *ICLR*, 2024. 3

[29] Tao Xie, Peishan Yang, Yudong Jin, Yingfeng Cai, Wei Yin, Weiqiang Ren, Qian Zhang, Wei Hua, Sida Peng, Xiaoyang Guo, and Xiaowei Zhou. Scal3R: Scalable test-time training for large-scale 3D reconstruction. In *CVPR*, 2026. 2

[30] Jianing Yang, Alexander Sax, Kevin J. Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3R: Towards 3D reconstruction of 1000+ images in one forward pass. In *CVPR*, 2025. 2

[31] Shuai Yuan, Yantai Yang, Xiaotian Yang, Xupeng Zhang, Zhonghao Zhao, Lingming Zhang, and Zhipeng Zhang. InfiniteVGGT: Visual geometry grounded transformer for endless streams, 2026. 1, 3, 6

[32] Hui Zeng, Daming Zhao, Pengfei Yang, WenXuan Hou, Tianyang Zheng, Hui Li, Weiye Ji, and Jidong Zhai. Lethe: Layer- and time-adaptive kv cache pruning for reasoning-intensive LLM serving. In *AAAI*, 2026. 3

[33] Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming 4d visual geometry transformer. In *ICLR*, 2026. 1, 3, 6