DIASTREAM: DISTRIBUTION-AWARE KV CACHE FOR STREAMING 3D RECONSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

In streaming visual geometry transformers that maintain a key-value (KV) cache, effectively preserving scene context under a fixed cache 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 multi-directional structured redundancy and that 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* 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

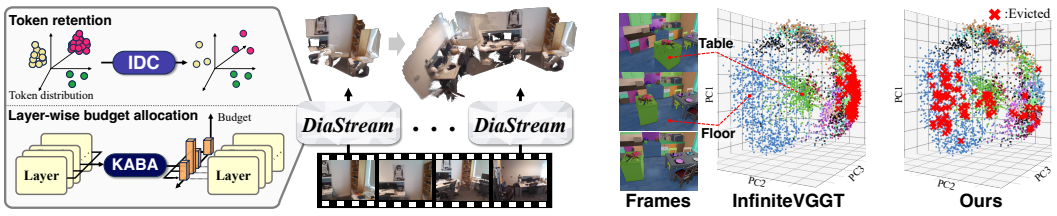
Recent geometric foundation models, such as DUSt3R (Wang et al., 2024) and VGGT (Wang et al., 2025a), have transformed multi-view 3D reconstruction by directly estimating camera parameters and dense scene geometry from multiple images in a single forward pass, eliminating the need for iterative geometric 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 (Zhuo et al., 2026) and STream3R (Lan et al., 2026) 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.

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 generates 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 a scene context over long input streams. Recent methods have addressed this problem using retention criteria based on accumulated attention weights (Mahdi et al., 2025), key-space diversity (Yuan et al., 2026), spatio-temporal cache compression (Wang et al., 2026a), and FFN-residual-based saliency (Lu et al., 2026). However, constructing a compact yet informative KV cache remains challenging because existing methods often evaluate individual tokens based on either importance or diversity, without jointly accounting for both. Consequently, retention based on importance alone may preserve redundant tokens, whereas diversity alone may favor less informative ones.

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

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) (b)

Figure 1: **(a) Overview of *DiaStream*.** *DiaStream* combines IDC for token retention and KABA for layer-wise cache allocation to achieve memory-efficient streaming 3D reconstruction under a limited cache budget. **(b) Comparison of evicted tokens in the key space distribution.** Red crosses denote evicted tokens projected onto the top three principal components. InfiniteVGGT, which uses a centroid-based diversity criterion, shows biased eviction concentrated in limited regions, whereas *DiaStream* selects eviction candidates across diverse regions by considering the underlying distribution.

in key space rather than concentrated around a single global centroid, as nearby views capture overlapping scene content and produce similar key representations (see Fig. 1b). 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. These observations suggest that an effective cache retention strategy should account for directional redundancy and layer-wise representation demand.

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. 1a). *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, together with accumulated attention weights. It thereby retains tokens that are both distinctive relative to the directional support of the cached-key distribution 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. 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.

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 (Mur-Artal et al., 2015; Newcombe et al.,

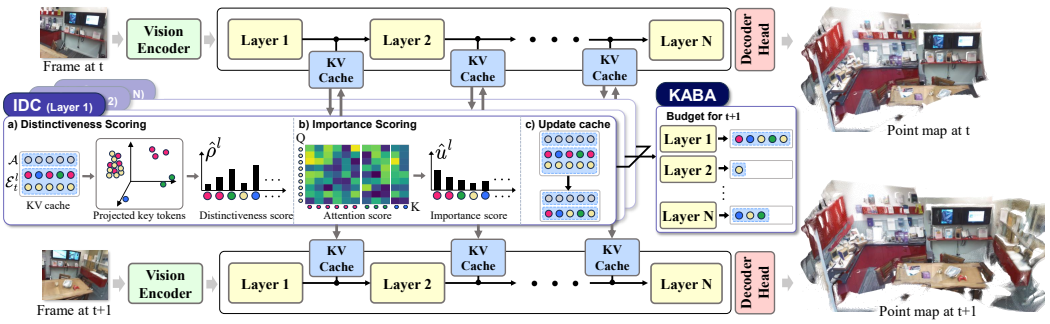


Figure 2: **Pipeline of *DiaStream*.** For each incoming frame, the corresponding KV representations are first appended to the layer-wise KV cache. IDC retains distinctive and important KV tokens, while KABA dynamically allocates the global cache budget across transformer layers. The resulting compact KV cache is then used for streaming visual geometry reconstruction.

2011; Dai et al., 2017). Recent feed-forward visual geometry models regress geometric quantities directly from uncalibrated images. DUSt3R (Wang et al., 2024) predicts dense, view-consistent pointmaps from image pairs, while MASt3R (Leroy et al., 2024) augments this formulation with dense local features for 3D-aware image matching. Fast3R (Yang et al., 2025) and VGGT (Wang et al., 2025a) 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 (Shen et al., 2026; Wang et al., 2026b), chunk-wise inference (Deng et al., 2026; Xie et al., 2026), or SLAM-style sequential processing (Maggio et al., 2025). 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 (Wang & Agapito, 2025) employs an external spatial memory to predict pointmaps in a common coordinate system, CUT3R (Wang et al., 2025b) recurrently updates a persistent latent state, and Point3R (Wu et al., 2025) maintains an explicit spatial pointer memory associated with 3D positions. These methods instantiate memory through recurrent or spatially structured modules. In particular, StreamVGGT (Zhuo et al., 2026) 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 (Xiao et al., 2024; Zeng et al., 2026; Liu et al., 2026), as well as combined leverage and non-causal attention scores (Chari & Van Durme, 2025). 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 (Mahdi et al., 2025) and STAC (Wang et al., 2026a), while diversity-based methods such as InfiniteVGGT (Yuan et al., 2026) preserve tokens based on key-space diversity. OVGGT (Lu et al., 2026) applies FFN-residual-based saliency to current-frame tokens and centroid-based key diversity to historical tokens. Both InfiniteVGGT and OVGGT also dynamically allocate layer budgets based on token diversity.

Unlike these streaming visual geometry methods, *DiaStream* combines token distinctiveness derived from the second-order distributional structure of cached keys with temporally aggregated attention-based importance for historical-token retention and dynamically allocates layer-wise cache budgets according to layer-wise key statistics.

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. 2, *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 (Lan et al., 2026; Zhuo et al., 2026) process image sequences incrementally by reusing historical KV tokens through a layer-wise cache. When the t-th frame arrives, the KV cache at layer l is updated 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. We therefore formulate fixed-budget cache retention as selecting a subset of cached KV tokens under a global budget B:

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

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

3.3 IMPORTANT-DISTINCTIVE CACHING

Motivated by the first observation, we use statistical leverage scores (Drineas et al., 2012) to measure token distinctiveness relative to the directional structure 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 tokens, we first define the evictable candidate token set while preserving the anchor tokens inherited from the first frame, as in InfiniteVGGT (Yuan et al., 2026). 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 $\tilde{\mathcal{M}}_t^l = \mathcal{A}^l \cup \mathcal{E}_t^l$. 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. Following Chari & Van Durme (2025), we project $\hat{\mathbf{K}}_t^l$ into a lower-dimensional subspace using Gaussian right-sketching to efficiently compute the approximate statistical leverage scores:

$$\mathbf{Z}_t^l = \hat{\mathbf{K}}_t^l \mathbf{R}^l, \tag{3}$$

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 $\mathbf{Z}_t^l$:

$$\mathbf{Z}_t^l = \mathbf{U}_t^l \mathbf{\Sigma}_t^l (\mathbf{W}_t^l)^\top, \qquad \rho_i^l = \|(\mathbf{U}_t^l)_{i,:}\|_2^2, \tag{4}$$

where ρ_i^l denotes the statistical leverage score of the i-th candidate token and $\|\cdot\|_2^2$ denotes the squared L2 norm. We interpret a larger leverage score as greater token distinctiveness relative to the aggregate directional support of the projected candidate keys.

To better understand how leverage scores account for directional redundancy, Eq. (4) can be equivalently expressed using the Gram matrix:

$$\mathbf{G}_t^l = (\mathbf{Z}_t^l)^\top \mathbf{Z}_t^l, \qquad \rho_i^l = \mathbf{z}_i^l (\mathbf{G}_t^l)^\dagger (\mathbf{z}_i^l)^\top, \tag{5}$$

where $\mathbf{z}_i^l \in \mathbb{R}^{1 \times r}$ denotes the i-th row of $\mathbf{Z}_t^l$ and $\dagger$ denotes the Moore–Penrose pseudoinverse. The pseudoinverse Gram matrix assigns smaller weights to directions that are strongly supported by many 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 and we compute the importance score as a decay-weighted average to obtain a robust importance score from multiple views:

$$u_i^l = \frac{\sum_{j=0}^{N_i-1} \gamma^{N_i-j-1} \bar{\alpha}_{t'+j,i}^l}{\sum_{j=0}^{N_i-1} \gamma^{N_i-j-1}}, \quad \bar{\alpha}_{t,i}^l = |\hat{\mathcal{M}}_t^l| \alpha_{t,i}^l, \tag{6}$$

where $\alpha_{t,i}^l$ denotes the attention probability assigned to the i-th candidate token by the frame query tokens at time step t, averaged over query tokens and attention heads, $\gamma \in (0, 1)$ is the decay factor, t' is the source frame time step of the i-th candidate token, and N_i is the number of frames processed starting from t', capped at T. The importance score is updated only during the first T frames after cache insertion and then kept fixed. This prevents a token's importance from decreasing solely because it receives less attention when the camera later observes a different scene region.

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

$$s_i^l = \hat{\rho}_i^l + \hat{u}_i^l. \tag{7}$$

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 $|\tilde{\mathcal{M}}_t^l| \leq b_t^l$, no eviction is performed. Please check the appendix for more details.

3.4 KEY STATISTICS-AWARE BUDGET ALLOCATION

Motivated by our second observation of layer-wise heterogeneity, 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.

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 (Recanatesi et al., 2022) over the leverage scores, yielding the leverage-score participation ratio:

$$\psi_t^l = \frac{\left(\sum_{i\in\mathcal{E}_t^l} \rho_i^l\right)^2}{\sum_{i\in\mathcal{E}_t^l} (\rho_i^l)^2}. \tag{8}$$

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-score participation ratio captures how leverage scores are distributed across candidate tokens, it does not reflect the absolute scale of the underlying key vectors. Key magnitude contributes to the scale of query-key interactions and can amplify attention responses to positively aligned queries. Motivated by this scale dependence, we incorporate key magnitude as a complementary statistic for estimating layer-wise representation demand alongside the leverage-score participation ratio:

$$\mu_t^l = \frac{1}{|\mathcal{E}_t^l|} \sum_{i\in\mathcal{E}_t^l} \|\mathbf{k}_i^l\|_2, \tag{9}$$

where $\|\cdot\|_2$ denotes the 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, which we treat as a signal of layer-wise representation demand.

Layer-wise budget allocation. The two statistics capture complementary characteristics of each transformer layer. To account for both the distribution of leverage scores and the scale of raw key representations, we combine them into a single allocation weight and compute the layer-wise budgets:

$$w_t^l = \left(\psi_t^l \mu_t^l\right)^\eta, \qquad b_{t+1}^l = \left\lfloor B \frac{w_t^l}{\sum_{\ell=1}^{L} w_t^\ell} \right\rfloor, \tag{10}$$

where $\eta > 0$ controls the sharpness of the allocation and $\lfloor\cdot\rfloor$ denotes the floor function. The resulting layer budgets are applied to cache retention for the next frame. In practice, we enforce a minimum layer-wise cache budget of $|A^l|$ to preserve anchor tokens, potentially relaxing the budget constraint.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Implementation details. We extend StreamVGGT (Zhuo et al., 2026) 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 set the random seed to 42 for all

experiments to ensure reproducibility. We use the CUDA column-sum operator from STAC (Wang et al., 2026a) to efficiently aggregate attention probabilities. All experiments are conducted on a single NVIDIA RTX A6000 GPU. Additional implementation details and additional experiments with different random seeds, frame strides, and hyperparameter settings are included in the appendix.

Evaluation settings. We evaluate 3D reconstruction on 7-Scenes (Shotton et al., 2013), NRGBD (Azinović et al., 2022), and ETH3D (Schöps et al., 2017), and camera pose estimation on 7-Scenes, Replica (Straub et al., 2019), and TUM-Dynamics (Sturm et al., 2012). Following OVGGT (Lu et al., 2026), 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. We use the first 100 input frames for short-sequence evaluation and the entire input sequence for full-sequence evaluation. The full input sequences in 7-Scenes and NRGBD range from 198 to 1,000 frames. ETH3D is excluded from the short-sequence setting because its sequences contain fewer than 100 frames. We report Chamfer Distance (CD) and F1 score for 3D reconstruction. The F1 score is computed using a threshold of $5\,\mathrm{cm}$ for 7-Scenes and NRGBD and $25\,\mathrm{cm}$ for ETH3D. For camera pose estimation, we report Absolute Trajectory Error (ATE) in meters, computed using the evo toolkit (Grupp, 2017) 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 (Wang et al., 2025b), Point3R (Wu et al., 2025), STream3R (Lan et al., 2026), and the VGGT-based StreamVGGT (Zhuo et al., 2026). The second group comprises StreamVGGT-based KV cache management methods for bounded-memory streaming, including InfiniteVGGT (Yuan et al., 2026), STAC (Wang et al., 2026a), and OVGGT (Lu et al., 2026). We use each method's default KV cache budget unless otherwise stated.

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 denotes methods without a fixed KV cache budget and *DiaStream-S* denotes *DiaStream* with a small budget of 30K.

Method	Budget	Short sequence				Full sequence						FPS↑
		7-Scenes		NRGBD		7-Scenes		NRGBD		ETH3D		
		CD↓	F1↑	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑	
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	<u>0.022</u>	0.899	0.017	0.935	0.028	0.867	0.042	0.766	0.503	0.562	<u>11.65</u>
DiaStream-S	30K	**0.016**	<u>0.945</u>	<u>0.015</u>	<u>0.962</u>	<u>0.022</u>	<u>0.912</u>	**0.025**	**0.894**	**0.416**	**0.643**	11.11
DiaStream	60K	**0.016**	**0.947**	**0.013**	**0.966**	**0.020**	**0.923**	<u>0.026</u>	<u>0.877</u>	<u>0.417</u>	<u>0.631</u>	10.83

4.2 3D Reconstruction

Tab. 1 summarizes the quantitative reconstruction results. *DiaStream* achieves the strongest reconstruction performance among memory-bounded methods across all evaluated datasets, under both short- and full-sequence settings. Compared with OVGGT, *DiaStream* achieves higher reconstruction accuracy across all evaluated datasets and maintains comparable throughput with only a 60K budget, approximately 70% smaller than OVGGT's 200K budget. The reconstruction advantage persists even with a 30K budget. StreamVGGT and STream3R maintain unbounded KV caches and therefore cannot complete full-sequence evaluation on 7-Scenes and NRGBD due to GPU memory exhaustion. These results demonstrate that the proposed method effectively preserves informative

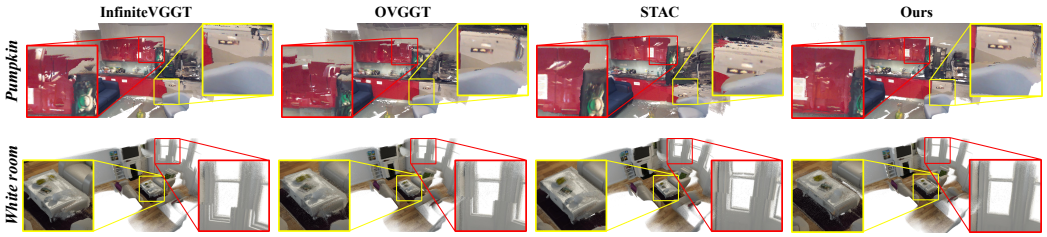


Figure 3: **Qualitative results on the *Pumpkin* sequence from 7-Scenes and the *White room* sequence from NRGBD.** The columns correspond to InfiniteVGGT, OVGGT, STAC, and DiaStream. Red and yellow boxes indicate regions enlarged for closer inspection.

scene representations, maintaining stable reconstruction quality even as the sequence length increases under strict memory constraints.

Qualitative results in Fig. 3 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.

Table 2: **Trajectory error on short and full sequences**. We report ATE ↓ in meters on short and full sequences across 7-Scenes, Replica, and TUM-Dynamics.

Method	Short sequence			Full sequence		
	7-Scenes	Replica	TUM	7-Scenes	Replica	TUM
CUT3R	0.071	0.185	0.051	0.566	0.817	0.135
Point3R	0.095	0.128	0.074	0.348	0.582	0.156
STream3R	0.034	0.020	0.024	OOM	OOM	OOM
StreamVGGT	0.041	0.020	0.025	OOM	OOM	OOM
STAC	0.042	0.024	0.026	0.129	OOM	0.046
InfiniteVGGT	0.043	**0.020**	0.028	0.127	0.244	0.054
OVGGT	0.036	0.028	0.027	0.152	0.357	0.067
DiaStream-S	0.035	0.022	**0.021**	0.127	0.176	0.046
DiaStream	**0.034**	0.022	0.023	**0.104**	**0.135**	**0.039**

4.3 Camera Pose Estimation

Long image streams inevitably accumulate localization errors, making effective long-term memory essential for stable camera tracking. We report the quantitative camera pose estimation results in Tab. 2. Despite using only a 60K-token cache, *DiaStream* achieves the lowest trajectory error among memory-bounded methods on all full-sequence benchmarks and remains competitive in the short-sequence setting among memory-bounded methods, requiring a roughly 70% smaller cache budget than STAC and OVGGT and 95% smaller than InfiniteVGGT.

Even with a 30K-token cache, *DiaStream* outperforms OVGGT across all evaluated datasets in both sequence settings and surpasses existing memory-bounded baselines on full-sequence Replica. The improvement on TUM-Dynamics further shows the robustness of our design in dynamic scenes. In contrast, StreamVGGT and STream3R cannot complete full-sequence evaluation due to GPU memory exhaustion, highlighting the need for bounded-cache inference over long image streams. STAC, which employs an additional spatial cache, also encounters OOM on Replica.

4.4 Ablation Study

Analysis of adaptive cache allocation. Fig. 4 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.

Table 3: **Ablation studies.** (a) Reconstruction performance under varying token budgets. (b) Effect of each proposed component on the short sequences of NRGBD using a 60K-token budget.

(a)

Method	Budget	7-Scenes	
		CD↓	F1↑
InfiniteVGGT	1.2M	0.036	0.817
	60K	0.102	0.627
OVGGT	200K	0.028	0.867
	60K	0.066	0.745
Ours	1.2M	0.026	0.884
	200K	0.022	0.900
	60K	**0.020**	**0.923**
	30K	0.022	0.912

(b)

IDC		KABA		CD↓	F1↑
ρ	u	ψ	μ		
✓				0.024	0.873
	✓			0.022	0.904
✓	✓			0.017	0.953
✓	✓	✓		0.016	0.949
✓	✓		✓	0.016	0.953
✓		✓	✓	0.016	0.953
	✓	✓	✓	0.015	0.956
✓	✓	✓	✓	**0.013**	**0.966**

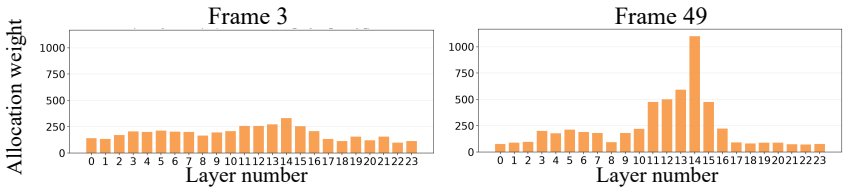


Figure 4: **Visualization of the layer-wise weights computed by KABA on 7-Scenes.**

Robustness under varying cache budgets. Tab. 3a demonstrates the robustness of *DiaStream* under varying KV cache budgets. Whereas competing methods show larger degradation on 7-Scenes under the 60K-token budget, *DiaStream* maintains stable reconstruction quality. *DiaStream* further surpasses OVGGT at a matched 200K-token budget and InfiniteVGGT at a matched 1.2M-token budget. These results highlight the importance of preserving informative scene representations within the available budget.

Effectiveness of the proposed components. Tab. 3b examines the contributions of the scoring terms in IDC and the allocation statistics in KABA. For IDC, combining distinctiveness and importance scoring improves reconstruction performance over using either score alone, both with and without KABA. For KABA, removing either the leverage-score participation ratio or the key-magnitude scale degrades performance relative to the full model, as does replacing adaptive allocation with uniform layer-wise budgets. The full model achieves the best performance, supporting the complementary roles of the IDC scoring terms and KABA allocation statistics.

4.5 Discussion

DiaStream focuses on preserving useful context in a compact KV cache. IDC removes tokens from textureless or dominant regions such as walls and floors (Fig. 5). Retaining more redundant context from these regions at larger budgets may contribute to the lower accuracy observed in Tab. 3a. Using additional cache capacity to further improve reconstruction is a direction for future work.

5 Conclusion

We presented *DiaStream*, a training-free, distribution-aware KV cache management framework for streaming 3D reconstruction. IDC combines leverage-based distinctiveness with attention-based importance for token retention, while KABA allocates the cache budget according to layer-wise key statistics. Experiments on multiple benchmarks show that *DiaStream* improves 3D reconstruction and camera pose estimation while using substantially smaller KV cache budgets than existing memory-bounded methods. Incorporating geometric cues, such as scene structure and camera motion, into cache management remains to be explored.

AI USE STATEMENT

Generative AI tools were used to assist with code implementation, language editing, translation, and literature discovery, and to provide feedback on research methodology. Problem identification, analysis, method design, and experiments were all carried out and reviewed by the authors. The authors reviewed the LLM-generated text and code, directly examined the identified literature, and take full responsibility for the final manuscript, implementation, and reported results.

REPRODUCIBILITY STATEMENT

We provide a detailed description of our method in Sec. 3 and hyperparameter configurations and implementation details in Sec. 4.1 and Appendix A.1 with pseudocode. The datasets, evaluation settings, and hardware are specified in Sec. 4.1 and Appendix A.2.

REFERENCES

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

Vivek Chari and Benjamin Van Durme. Compactor: Calibrated query-agnostic KV cache compression with approximate leverage scores. arXiv preprint arXiv: 2507.08143, 2025.

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 TOG*, 36(3):1–18, 2017. Article 24.

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.

Petros Drineas, Malik Magdon-Ismail, Michael W. Mahoney, and David P. Woodruff. Fast approximation of matrix coherence and statistical leverage. *JMLR*, 13(111):3475–3506, 2012.

David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NeurIPS*, 2014.

Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3D object reconstruction from a single image. In *CVPR*, 2017.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *CVPR*, 2012.

Michael Grupp. evo: Python Package for the Evaluation of Odometry and SLAM, 2017.

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.

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

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.

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. In *ECCV*, 2026.

Dominic Maggio, Hyungtae Lim, and Luca Carlone. VGGT-SLAM: Dense RGB SLAM optimized on the SL(4) manifold. In *NeurIPS*, 2025.

Soroush Mahdi, Fardin Ayar, Ehsan Javanmardi, Manabu Tsukada, and Mahdi Javanmardi. Evict3R: Training-free token eviction for memory-bounded streaming visual geometry transformers. arXiv preprint arXiv: 2509.17650, 2025.

Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE TRO*, 31(5):1147–1163, 2015.

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

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.

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.

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

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.

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. arXiv preprint arXiv: 1906.05797, 2019.

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.

Maxim Tatarchenko, Stephan R. Richter, Rene Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3D reconstruction networks learn? In *CVPR*, 2019.

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

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

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

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

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

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

Yuqi Wu, Wenzhao Zheng, Jie Zhou, and Jiwen Lu. Point3R: Streaming 3D reconstruction with explicit spatial pointer memory. In *NeurIPS*, 2025.

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

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.

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.

Shuai Yuan, Yantai Yang, Xiaotian Yang, Xupeng Zhang, Zhonghao Zhao, Lingming Zhang, and Zhipeng Zhang. InfiniteVGGT: Visual geometry grounded transformer for endless streams. arXiv preprint arXiv: 2601.02281, 2026.

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.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Forrester Cole, Deqing Sun, Ming-Hsuan Yang, et al. MonST3R: A simple approach for estimating geometry in the presence of motion. In *ICLR*, 2025.

Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming visual geometry transformer. In *ICLR*, 2026.

A Appendix

A.1 Implementation Details

Streaming Pipeline. Algorithm 1 summarizes the complete online inference pipeline of *DiaStream*. Given an image stream $\{I_t\}_{t=1}^{N}$, key-value pairs generated for each incoming frame are appended to the retained cache, and IDC is invoked only when the pre-eviction cache exceeds the current layer budget. The current queries then attend to the post-eviction cache via $\mathrm{CausalAttention}(\mathcal{M}_t^l)$, which performs temporal causal self-attention. The resulting attention statistics are collected for subsequent importance-score updates. Once the required candidate statistics are available, KABA determines the layer-wise budgets for the next frame. Finally, $\mathrm{Decode3DGeometry}()$ applies the prediction heads to the current-frame features to produce the frame-level 3D geometry predictions such as a point map, a depth map, and a camera pose.

Algorithm 1 *DiaStream* streaming pipeline

Input: $\{I_t\}_{t=1}^{N}$, B, $\{R^l\}_{l=1}^{L}$, η

Initialize $\mathcal{M}_0^l \leftarrow \emptyset$ for all layers
Initialize $\mathcal{A}^l \leftarrow \emptyset$ for all layers
Initialize $b_1^l \leftarrow \lfloor B/L \rfloor$ for all layers.
for each frame t in the input sequence **do**
  for $l = 1, \ldots, L$ **do**
    $\tilde{\mathcal{M}}_t^l = \mathcal{M}_{t-1}^l \cup (\mathbf{K}_t^l, \mathbf{V}_t^l)$ # Eq. (1)
    if $\mathcal{A}^l = \emptyset$ **then**
      $\mathcal{A}^l = (\mathbf{K}_t^l, \mathbf{V}_t^l)$
    end if
    $\mathcal{E}_t^l \leftarrow \tilde{\mathcal{M}}_t^l \setminus \mathcal{A}^l$
    if $|\tilde{\mathcal{M}}_t^l| > b_t^l$ **then**
      # Retain tokens under the layer budget using IDC.
      $(\hat{\mathcal{M}}_t^l, \{\rho_i^l\}_{i \in \mathcal{E}_t^l}) = \mathrm{IDC}(\mathcal{E}_t^l, \mathcal{A}^l, \{\bar{\alpha}_{t-1,i}^l\}_{i \in \mathcal{E}_t^l}, b_t^l)$
    else
      $\hat{\mathcal{M}}_t^l = \tilde{\mathcal{M}}_t^l$
    end if
    $\mathrm{CausalAttention}\left(\hat{\mathcal{M}}_t^l\right)$
    # Normalize the averaged attention weights by cache size.
    $\{\bar{\alpha}_{t,i}^l = |\hat{\mathcal{M}}_t^l| \alpha_{t,i}^l\}_{i \in \hat{\mathcal{M}}_t^l}$ # Eq. (6)
    $\mathcal{M}_t^l = \hat{\mathcal{M}}_t^l$
  end for
  # Update layer-wise budgets using KABA.
  $\{b_{t+1}^l\}_{l=1}^{L} = \mathrm{KABA}\left(\{\mathcal{E}_t^l, \{\rho_i^l\}_{i \in \mathcal{E}_t^l}\}_{l=1}^{L}, B, \eta, |A^l|\right)$
  $\mathrm{Decode3DGeometry}()$ # Predict per-frame 3D geometry.
end for

Output: Per-frame 3D geometry predictions

Algorithm 2 Important-Distinctive Caching

Input: $\mathcal{E}_t^l, \mathcal{A}^l, \{\bar{\alpha}_{t-1,i}^l\}_{i\in\mathcal{E}_t^l}, b_t^l$

$\hat{\mathbf{K}}_t^l = \mathrm{NormalizeKeys}\left(\{\mathbf{k}_i^l\}_{i\in\mathcal{E}_t^l}\right)$
$\mathbf{Z}_t^l = \hat{\mathbf{K}}_t^l \mathbf{R}^l$ # Eq. (3)
$\mathbf{G}_t^l = \left(\mathbf{Z}_t^l\right)^\top \mathbf{Z}_t^l$ # Eq. (5)
for $i \in \mathcal{E}_t^l$ **do**
  # Compute distinctiveness score and importance score.
  $\rho_i^l = \mathbf{z}_i^l \left(\mathbf{G}_t^l\right)^\dagger \left(\mathbf{z}_i^l\right)^\top$ # Eq. (5)
  if i belongs to frame t **then**
    $u_i^l = \frac{\sum_{j=0}^{N_i-1} \gamma^{N_i-j-1} \bar{\alpha}_{t'+j,i}^l}{\sum_{j=0}^{N_i-1} \gamma^{N_i-j-1}}$ # Eq. (6)
  end if
end for
for $i \in \mathcal{E}_t^l$ **do**
  # Combine the mean-normalized $\hat{u}_i^l$ and $\hat{\rho}_i^l$.
  if i belongs to frame t **then**
    $s_i^l = 2\hat{\rho}_i^l$
  else
    $s_i^l = \hat{\rho}_i^l + \hat{u}_i^l$ # Eq. (7)
  end if
end for
  # Retain anchor tokens and the highest-scoring tokens.
$\hat{\mathcal{M}}_t^l = \mathcal{A}^l \cup \{(\mathbf{k}_i^l, \mathbf{v}_i^l) \mid i \in \mathrm{Top}_{b_t^l - |\mathcal{A}^l|}\{s_i^l\}_{i\in\mathcal{E}_t^l}\}$

Output: $\hat{\mathcal{M}}_t^l, \{\rho_i^l\}_{i\in\mathcal{E}_t^l}$

Important-Distinctive Caching. Algorithm 2 details how IDC updates the KV cache. $\mathrm{NormalizeKeys}(\cdot)$ independently normalizes each head-wise post-RoPE key vector to a unit vector, concatenates the normalized keys across heads for each token, and stacks the resulting token representations into a matrix.

LEVERAGE SCORE COMPUTATION. Eq. (4) characterizes the statistical leverage scores through a compact SVD. In our implementation, we instead use the equivalent Gram-matrix formulation in Eq. (5), avoiding an explicit compact SVD.

ATTENTION COLUMN SUM AND DECAY. IDC requires the total attention received by each cached key from the current queries. Materializing the full attention tensor solely to compute this statistic would introduce unnecessary memory overhead. We therefore use the custom CUDA column-sum operator from STAC (Wang et al., 2026a). Following STAC, we set the decay factor to $\gamma = 0.9$. Unlike STAC's decayed cumulative score, *DiaStream* computes a normalized decay-weighted average over the first $T = 5$ observations, starting from each token's source frame and then fixes the resulting importance score.

COMBINED SCORE. When processing the t-th frame, newly appended tokens have not yet received attention from subsequent frames and therefore do not have an attention-based importance score. For these tokens, we use twice the mean-normalized leverage score as the combined retention score:

$$s_i^l = \begin{cases} 2\hat{\rho}_i^l, & \text{if } i \text{ belongs to the current frame,} \\ \hat{\rho}_i^l + \hat{u}_i^l, & \text{otherwise.} \end{cases}$$

Since the leverage and importance scores are independently mean-normalized, this adjustment keeps the score scale of newly appended tokens comparable to that of previously cached tokens.

Key Statistics-Aware Budget Allocation. Algorithm 3 details the KABA procedure.

INITIAL BUDGET. The layer-wise cache budgets are initialized uniformly under the global budget. Once statistics for the non-anchor candidates become available, KABA replaces the uniform initialization with adaptive layer-wise budgets computed from these statistics.

STATISTICS USED FOR ALLOCATION. KABA computes the leverage-score participation ratio using the raw leverage scores before mean normalization. For layers in which eviction is not triggered, leverage scores are not recomputed for the current candidate set. We therefore use the most recently computed leverage-score participation ratio for those layers. The key-magnitude statistic is computed from the raw key vectors before head-wise normalization and random projection.

BUDGET FLOOR. When an allocated budget falls below a minimum layer-wise capacity of $|A^l|$, only the anchor tokens are retained in that layer. This anchor-preserving adjustment may cause the total number of retained KV token pairs across layers to exceed the configured global budget B.

Algorithm 3 Key Statistics-Aware Budget Allocation

Input: $\{\mathcal{E}_t^l, \{\rho_i^l\}_{i\in\mathcal{E}_t^l}\}_{l=1}^L, B, \eta, |\mathcal{A}^l|$

for $l = 1, \ldots, L$ **do**
  # Compute leverage-score participation ratio.
  $\psi_t^l = \dfrac{\left(\sum_{i\in\mathcal{E}_t^l} \rho_i^l\right)^2}{\sum_{i\in\mathcal{E}_t^l} (\rho_i^l)^2}$ # Eq. (8)
  # Measure key-magnitude scale.
  $\mu_t^l = \dfrac{1}{|\mathcal{E}_t^l|} \sum_{i\in\mathcal{E}_t^l} \|\mathbf{k}_i^l\|_2$ # Eq. (9)
  # Form a single allocation weight from two statistics.
  $w_t^l = \left(\psi_t^l \mu_t^l\right)^\eta$ # Eq. (10)
end for
for $l = 1, \ldots, L$ **do**
  # Allocate layer-wise budgets for frame at $t+1$.
  $b_{t+1}^l = \left\lfloor B \dfrac{w_t^l}{\sum_{\ell=1}^L w_t^\ell} \right\rfloor$ # Eq. (10)
  if $b_{t+1}^l < |\mathcal{A}^l|$ **then**
    $b_{t+1}^l = |\mathcal{A}^l|$
  end if
end for

Output: $\{b_{t+1}^l\}_{l=1}^L$

A.2 EVALUATION METRICS

3D Reconstruction and Camera Pose Estimation. We follow prior works for streaming reconstruction evaluations (Wang et al., 2026a; Zhuo et al., 2026) and adopt standard 3D reconstruction metrics, including Chamfer distance (CD) (Fan et al., 2017) and F1 score (Tatarchenko et al., 2019). Chamfer distance measures the geometric discrepancy between the reconstructed and ground-truth point clouds, while F1 score evaluates reconstruction accuracy by combining precision and recall under a predefined distance threshold. Camera trajectories are evaluated using the evo toolkit (Grupp, 2017) with Umeyama alignment applied between estimated and ground-truth trajectories.

A.3 ADDITIONAL EXPERIMENTS

We provide additional experiments and ablation studies to further analyze the effectiveness and robustness of *DiaStream*. We investigate the impact of key design choice such as allocation exponent η of KABA and the sensitivity to random seeds. Furthermore, we evaluate *DiaStream* on an outdoor depth estimation benchmark and analyze its robustness under different temporal sampling strides.

Allocation exponent. The allocation exponent η controls the allocation sharpness induced by the KABA weights $w_t^l = (\psi_t^l \mu_t^l)^\eta$. In Tab. 4a, we conduct an ablation study on the allocation exponent η of KABA to investigate its impact on cache budget allocation. With $\eta = 0$, KABA becomes equivalent to uniform allocation, while larger values increasingly bias the allocation toward layers with high KABA weight. The best performance is achieved with $\eta = 0.7$, indicating that a moderate degree of adaptation to layer statistics yields the most effective budget allocation. Smaller values of η produce smoother allocations that do not sufficiently reflect layer-specific representation demands.

Table 4: **Quantitative results.** (a) Effect of the KABA allocation exponent η on full-sequence NRGBD. (b) Sensitivity to the random seed on full-sequence 7-Scenes and NRGBD.

(a)

η	CD↓	F1↑
1.0	0.048	0.793
0.7 (default)	**0.026**	**0.877**
0.5	0.027	0.850
0.3	0.032	0.815
0.0	0.037	0.811

(b)

Seed	7-Scenes		NRGBD	
	CD↓	F1↑	CD↓	F1↑
46	0.020	0.924	0.027	0.875
45	0.020	0.923	0.026	0.879
44	0.020	0.923	0.026	0.869
43	0.020	0.922	0.027	0.882
42	0.020	0.923	0.026	0.877

Table 5: **Video depth estimation results on four KITTI sequences.** We report Abs Rel ↓, RMSE ↓, and $\delta < 1.25$ ↑.

Method	Sequence 0002			Sequence 0020			Sequence 0113			Sequence 0026		
	Abs Rel	RMSE	$\delta < 1.25$	Abs Rel	RMSE	$\delta < 1.25$	Abs Rel	RMSE	$\delta < 1.25$	Abs Rel	RMSE	$\delta < 1.25$
STAC	0.184	4.088	0.637	0.099	4.006	0.906	0.140	5.281	0.814	0.071	2.682	0.978
InfiniteVGGT	0.161	3.603	0.737	<u>0.098</u>	<u>3.897</u>	<u>0.918</u>	0.120	5.342	0.899	<u>0.046</u>	<u>2.380</u>	<u>0.980</u>
OVGGT	<u>0.142</u>	<u>3.446</u>	<u>0.787</u>	0.113	3.944	0.883	<u>0.113</u>	5.188	**0.943**	0.049	2.390	<u>0.980</u>
DiaStream	**0.119**	**2.995**	**0.863**	**0.087**	**3.706**	**0.931**	**0.108**	**5.164**	<u>0.936</u>	**0.044**	**2.336**	**0.982**

In contrast, $\eta = 1.0$ makes the allocation overly concentrated, assigning most tokens to a few layers while allocating too few tokens to the remaining layers.

Random-seed sensitivity. *DiaStream* is training-free and directly operates on a pretrained model, introducing limited optimization-related randomness. However, the leverage scores rely on a random projection, which may introduce variation across different random seeds. Therefore, we evaluate the sensitivity of *DiaStream* to random seed changes in Tab. 4b. The results show that the performance variation across seeds remains small, demonstrating the robustness of our method to the randomness introduced by the projection step. The reported statistics are computed over five independent runs.

Video Depth Estimation on Outdoor Dataset. We evaluate the depth estimation performance of *DiaStream* on long outdoor driving sequences from the KITTI benchmark (Geiger et al., 2012). As shown in Tab. 5, we report the absolute relative error (Abs Rel), root mean squared error (RMSE), and threshold accuracy at $\delta < 1.25$, following the standard depth estimation evaluation protocol (Eigen et al., 2014). We compare *DiaStream* against InfiniteVGGT, OVGGT, and STAC on the KITTI dataset following the experimental setup of MonST3R (Zhang et al., 2025), aligning the predicted depth maps to the ground truth using a per-sequence scale. The results show lower Abs Rel and RMSE for *DiaStream* than for the comparison methods on all four evaluated sequences, with higher δ_1 on three of them, demonstrating that our proposed KV cache effectively preserves scene representations during outdoor sequences and leads to improved depth estimation accuracy.

Temporal-Stride Variation. While the main experiments use a fixed stride of 2 for fair comparison with existing methods, we further evaluate the robustness of *DiaStream* under different keyframe sampling intervals in Tab. 6. We use a 60K cache budget for our method and vary the sampling stride. The results demonstrate that the proposed method maintains stable performance under different temporal sampling rates.

A.4 Additional Qualitative Results

Token-Eviction Visualization. We further check which tokens are evicted by the IDC. Fig. 5 visualizes the token-eviction decisions made by IDC at layer 12 on sequences from 7-Scenes and NRGBD. We observe that IDC evicts tokens across multiple common scene elements, such as floors, walls, and table surfaces. This indicates our IDC module successfully evicts redundant scene features so that it can preserve an informative scene representation with a compact budget.

Reconstruction Comparison. Fig. 6 presents reconstruction examples from 7-Scenes and NRGBD. *DiaStream* produces cleaner and more structurally consistent reconstructions in the high-

Table 6: **3D reconstruction results at stride 3 and 5**. We evaluate 3D reconstruction on 7-Scenes and NRGBD using frame strides of 3 and 5, and compare *DiaStream* with baselines.

Method	Stride 3				Stride 5			
	7-Scenes		NRGBD		7-Scenes		NRGBD	
	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑
STAC	0.049	0.754	0.081	0.602	0.041	0.780	0.085	0.610
InfiniteVGGT	0.036	0.819	0.059	0.681	0.037	0.814	<u>0.064</u>	0.677
OVGGT	<u>0.027</u>	<u>0.873</u>	<u>0.040</u>	<u>0.774</u>	<u>0.028</u>	<u>0.871</u>	0.045	<u>0.772</u>
DiaStream	**0.020**	**0.922**	**0.029**	**0.857**	**0.022**	**0.916**	**0.031**	**0.846**

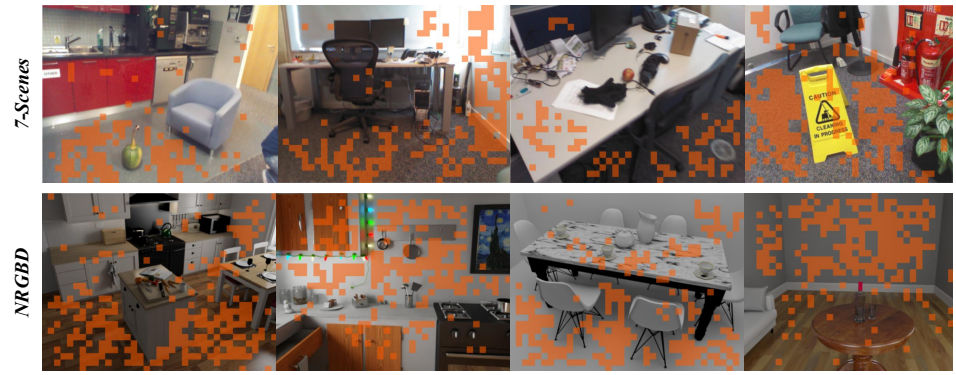


Figure 5: **Token-eviction visualization on 7-Scenes and NRGBD.** Orange patches indicate evicted tokens at layer 12 when processing the frame at time step t.

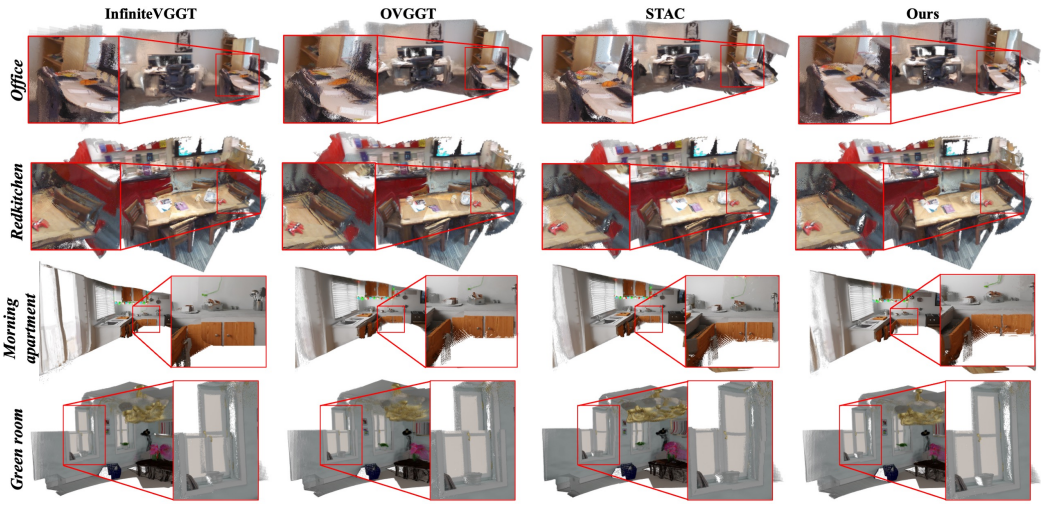


Figure 6: **Additional reconstruction results on 7-Scenes and NRGBD.** The columns correspond to InfiniteVGGT, OVGGT, STAC, and *DiaStream*.

lighted regions. In office and redkitchen, it better preserves the boundaries of the table and surrounding objects while reducing surface artifacts. In morning apartment, *DiaStream* reconstructs the cabinet region with less geometric distortion, preserving straighter boundaries and a more regular planar structure. In green room, it retains a cleaner and more continuous window-frame structure. These examples qualitatively support that *DiaStream* preserves reconstruction-relevant scene information under a compact cache budget.