OTT-VID: OPTIMAL TRANSPORT TEMPORAL TOKEN COMPRESSION FOR VIDEO-LLMS

Anonymous authors
Paper under double-blind review

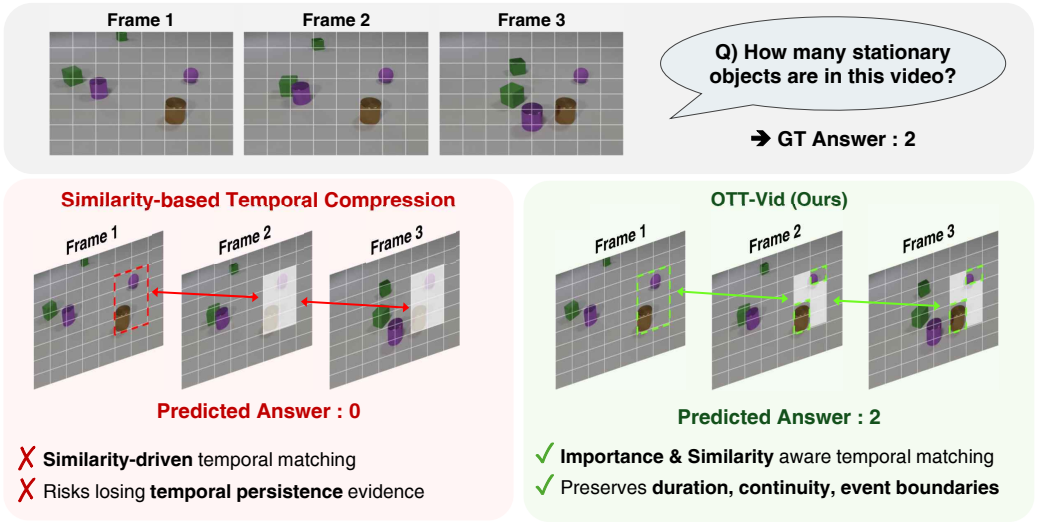

Figure 1: Illustration of temporal compression strategies, where white-shaded regions denote compressed tokens. For simplicity, only compression of stationary object regions is depicted. Similarity-driven compression merges stationary objects away due to high cross-frame similarity, losing persistence evidence. OTT-Vid jointly considers token importance and similarity, preserving semantically representative tokens across frames.

ABSTRACT

As Video Large Language Models (Video-LLMs) scale to longer and more complex videos, their inference cost grows rapidly due to visual tokens accumulated across frames. Training-free token compression reduces this overhead in part by exploiting repetition across frames, often using cross-frame similarity to identify redundant tokens or group similar frames for compression. However, cross-frame similarity and segmentation alone do not capture each token's importance within its frame or directly determine how much compression to apply to each frame pair. In this work, we propose **OTT-Vid**, a training-free outer-LLM compression framework that incorporates intra-frame token importance into temporal matching and derives frame-pair budgets from the resulting transport difficulty. After spatial pruning retains representative tokens, we formulate optimal transport (OT) between neighboring frames using importance-aware masses and a locality-aware cost combining feature dissimilarity with spatial distance. Smaller masses limit the coupling values of important tokens, reducing their priority in compression candidate ranking. The transport plan guides token-level compression, while its total cost serves as a proxy for compression difficulty, assigning larger compression budgets to lower-cost frame pairs. Experiments on six benchmarks spanning video question answering and temporal grounding across four backbones demonstrate strong performance at multiple token retention ratios, with gains in temporal grounding supporting the preservation of temporal evidence under compression.

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.

1 INTRODUCTION

Recent advances in Video-LLMs (Zhang et al., 2024; Li et al., 2024a; Wang et al., 2024; Bai et al., 2025b) have achieved remarkable progress in video understanding, enabling complex tasks such as video question answering, temporal grounding, and long video reasoning. As these capabilities extend to longer and more complex videos, a practical bottleneck has become increasingly apparent: processing many frames requires a large number of visual tokens, which quickly drives up sequence length and computational cost. In practice, the resulting visual token overhead often limits how many frames a Video-LLM can handle effectively.

To mitigate this bottleneck, training-free token compression has emerged as a practical direction that preserves compatibility with pretrained Video-LLMs without retraining. Such methods can be broadly categorized into two regimes according to where compression is applied in the inference pipeline: inner-LLM methods and outer-LLM methods. Inner-LLM methods (Chen et al., 2024; Fu et al., 2025b) operate within the transformer layers of the LLM to prune or merge tokens at intermediate depths. While they benefit from richer intermediate representations, layers before the compression point still process the full visual token sequence, and suitable pruning configurations may vary across LLM architectures. Outer-LLM methods (Shen et al., 2026; Fan et al., 2026), by contrast, compress visual tokens before LLM prefilling, directly shortening the sequence processed by the decoder. They can also be combined with inner-LLM methods for further token reduction (Shao et al., 2026).

Within the outer-LLM regime, video token compression targets two sources of redundancy under a fixed token budget: spatial redundancy, where patches within each individual frame encode overlapping content, and temporal redundancy, where similar content recurs across frames. The vision encoder provides token features and self-attention weights that support spatial compression through attention-based saliency (Yang et al., 2025), diversity (Alvar et al., 2025), coverage (Deng et al., 2026), and intra-frame feature similarity. Temporal compression, by contrast, requires cross-frame correspondences that are often not explicitly provided by the encoder. Many existing methods (Hyun et al., 2025; Shen et al., 2026) therefore rely on cross-frame feature similarity as a key signal for identifying temporal redundancy.

High cross-frame similarity, however, does not necessarily indicate removable redundancy. Tokens with similar features across frames may represent uninformative background, but they may also encode the persistence of semantically important objects or actions. Compressing solely according to similarity can therefore discard evidence of duration, continuity, and event boundaries. The challenge is to distinguish tokens that can be compressed from those whose presence across frames provides meaningful temporal evidence. Intra-frame signals offer a useful complementary cue by indicating a token's contribution to its frame representation. Although these signals do not directly measure temporal importance, they can help assess what information may be lost through compression. Combining this frame-level context with cross-frame similarity can thus help preserve meaningful temporal evidence.

Beyond deciding which tokens to compress, a video-level compressor must also determine how much compression to apply to each neighboring frame pair under a limited token budget. The amount of redundancy that can be removed while preserving meaningful evidence varies over time, making some frame pairs more amenable to compression than others. Existing approaches use cross-frame token similarity thresholds (Hyun et al., 2025; Shao et al., 2026) or temporal segmentation (Fan et al., 2026; Shen et al., 2026) to regulate temporal compression. These mechanisms help identify compression candidates or restrict where compression is applied, but do not by themselves establish how the total budget should be distributed across frame pairs. The remaining challenge is to allocate the budget according to pairwise compression difficulty, as assessed through the token correspondences used for compression.

To address these challenges, we propose **OTT-Vid**, a training-free outer-LLM video token compression framework that first applies spatial pruning to retain representative semantic content within each frame, and then performs adaptive temporal compression between neighboring frames. Our method considers two complementary factors when determining temporal compression: **(1)** each token's importance to its frame representation, and **(2)** the ease of transferring its information across neighboring frames. Optimal transport (OT) provides a principled framework for bringing these considerations together through its two primitives, **mass** and **cost**. We use mass to encode compression priority, assigning smaller mass to tokens with higher intra-frame importance, and cost to measure

cross-frame transfer difficulty. OT jointly determines how this mass is transported, considering both the suitability of each correspondence and competition among tokens for shared counterparts. In this way, the frame-level importance established by spatial pruning informs a coordinated compression decision across the two token sets. The resulting transport plan guides which tokens to compress, while its associated cost estimates the compression difficulty of each neighboring frame pair. We compare these costs across the video to distribute a fixed temporal compression budget according to relative difficulty, assigning more compression to easier pairs and less to harder ones. OTT-Vid thus connects token-level compression decisions with pair-level budget allocation through a unified transport formulation.

We evaluate OTT-Vid on four video question answering benchmarks, MVBench, VideoMME, LongVideoBench, and MLVU, and two temporal grounding benchmarks, Charades-TimeLens and ActivityNet-TimeLens. Experiments span four VTG-capable backbones, Qwen2.5-VL, Qwen3-VL, InternVideo3, and InternVL3.5, at 10%, 20%, and 30% token retention. OTT-Vid achieves strong performance on both tasks, with gains over competitive compression baselines on temporal grounding supporting its ability to retain evidence needed for event localization. These benefits extend across different backbones and retention ratios, demonstrating the broad effectiveness of our approach.

2 Background and Related Work

2.1 Video-LLM Inference Pipeline

Video-LLMs (Li et al., 2024a; Zhang et al., 2024; Bai et al., 2025b; An et al., 2026; Bai et al., 2025a; Yan et al., 2026) combine a vision encoder, a projector, and a language decoder. Let T denote the number of frames or temporal groups after temporal patching, each represented by N_v visual tokens. Projecting these tokens into the language embedding space yields $H_v \in \mathbb{R}^{TN_v \times d}$. Together with text embeddings $H_q \in \mathbb{R}^{N_q \times d}$, the multimodal input $H = \mathrm{concat}(H_v, H_q)$ has length $n = TN_v + N_q$. As multiple visual tokens accumulate across frames or temporal groups, their total count TN_v typically far exceeds the text length N_q, making visual tokens the main contributor to sequence length and prefill overhead. Self-attention computation during prefill scales quadratically with n. Visual token compression reduces this overhead by shortening the sequence before or during decoder processing.

2.2 Token Compression for Video-LLMs

Training-free visual token compression reduces redundancy without retraining the backbone. Spatial methods use attention-based saliency, diversity, or coverage to retain representative visual content (Yang et al., 2025; Alvar et al., 2025; Deng et al., 2026). Spatiotemporal methods extend compression across frames through temporal segmentation (Shen et al., 2026; Fan et al., 2026), locality-constrained matching (Shao et al., 2026; Ju et al., 2026), or selection from the full spatiotemporal token pool (Cho et al., 2026; Du et al., 2026).

Recent methods differ in how they select and consolidate temporal information. ST-SimDiff (Luo et al., 2026) combines similarity-based representative selection with temporal differences to retain event tokens. SemVID (Li et al., 2026a) selects object, motion, and context tokens for temporal grounding, allocating budgets using query relevance and inter-frame variation. AOT (Li et al., 2026b) uses OT to aggregate information into preselected anchors and guide temporal token retention. OTT-Vid formulates temporal compression between spatially retained token sets as importance-aware transport, using its couplings to determine compression priorities and its cost to distribute the compression budget across frame pairs.

Hybrid methods, including HoliTom (Shao et al., 2026) and FlashVID (Fan et al., 2026), combine outer-LLM compression with pruning at intermediate decoder layers, whose configurations may vary across architectures. OTT-Vid requires no decoder-layer selection and uses fixed compression hyperparameters across four evaluated backbones. Its benefits also persist when combined with inner-LLM compression (Appendix E).

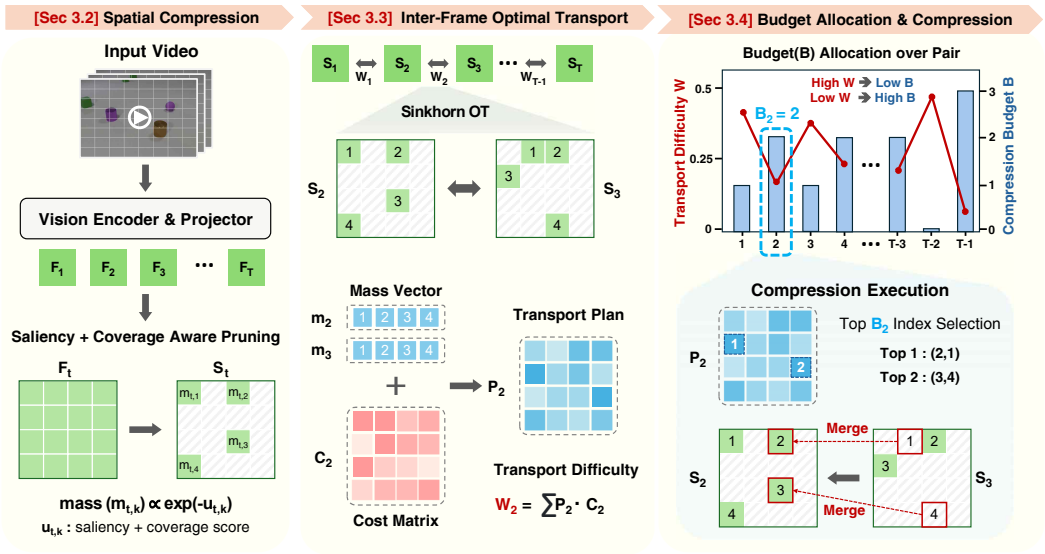

Figure 2: Overview of **OTT-Vid**. **(Sec 3.2)** Each frame t is encoded into visual tokens F_t, spatially pruned to retained tokens S_t, and converted into a mass vector m_t from per-token importance. **(Sec 3.3)** Sinkhorn OT computes a transport plan P_t and transport difficulty W_t for each neighboring frame pair. **(Sec 3.4)** Compression budgets B_t are allocated with larger budgets assigned to lower-difficulty pairs, and the transport plan determines which tokens to compress.

3 METHOD

3.1 OVERVIEW

An overview of **OTT-Vid** is provided in Figure 2. A video of T frames or temporal groups is encoded into N_v visual tokens per frame by the vision encoder and projector, yielding $T \cdot N_v$ tokens in total. Given a target retention ratio r, OTT-Vid compresses this set through two stages, governed by a temporal share parameter $\gamma \in [0, 1]$ that controls the balance between spatial and temporal compression. The total retention ratio r is decomposed as $r = r_s \cdot r_t$, where $r_s = r^{1-\gamma}$ and $r_t = r^{\gamma}$. In **Sec. 3.2**, we first prune each frame independently to remove spatial redundancy, retaining $K = N_v \cdot r_s$ tokens per frame and constructing a non-uniform mass for each retained token according to its frame-wise importance. In the temporal stage, we solve optimal transport between neighboring frame pairs to obtain two decision signals: a transport coupling that ranks compression candidates under importance-aware mass and matching cost (**Sec. 3.3**), and a transport difficulty that estimates pairwise compressibility. We then distribute a total budget of $B_{\text{tot}} = K \cdot T \cdot (1 - r_t)$ compression operations across all $T-1$ adjacent frame pairs according to this difficulty, and simultaneously execute the allocated compression on all pairs in parallel (**Sec. 3.4**).

3.2 SPATIAL COMPRESSION AND MASS ASSIGNMENT

The first stage of our framework compresses each frame independently and assigns a transport mass to each retained token. Prior spatial token selection methods (Deng et al., 2026; Fan et al., 2026) have explored semantic importance and representativeness as complementary criteria: importance prioritizes informative content, while representativeness encourages the selected tokens to collectively cover the frame's visual information. We instantiate this principle with a saliency-weighted coverage objective, where coverage is weighted by the saliency of the tokens being covered, and pass the selected tokens to the temporal OT stage.

Saliency-weighted representative token selection. For each frame t with N_v tokens $\{x_{t,i}\}_{i=1}^{N_v}$ where $x_{t,i} \in \mathbb{R}^d$, our goal is to retain K tokens. A saliency score $w_{t,i}$ is computed from the last layer of the vision encoder by averaging its self-attention weights over all heads and query positions,

normalized such that $\sum_i w_{t,i} = 1$. Starting from an empty set $\mathcal{S}_t$, we greedily select K tokens by maximizing the marginal gain in saliency-weighted coverage:

$$\text{gain}(j) = \sum_{i=1}^{N_v} w_{t,i} \max\big(0,\, \text{sim}(x_{t,i},\, x_{t,j}) - \mu_i\big), \tag{1}$$

Here, $\text{sim}(\cdot,\cdot)$ denotes cosine similarity, and $\mu_i = \max_{v \in \mathcal{S}t} \text{sim}(xt, i, v)$ measures how well the selected tokens cover $x_{t,i}$. We initialize μ_i to zero and update it after each selection.

Leave-one-out mass assignment. We assign transport mass from each retained token's leave-one-out contribution to the same saliency-weighted coverage objective. For each original token $x_{t,i}$, let $\hat{k}(i)$ identify its best-matching representative, and let $\sigma_{i,1}$ and $\sigma_{i,2}$ denote its highest and second-highest similarities to the retained tokens. Their gap estimates the coverage lost when the best match is removed. We aggregate these saliency-weighted gaps into $u_{t,k}$ and apply a negative softmax with temperature τ_m to obtain $m_{t,k}$:

$$u_{t,k} = \sum_{i:\hat{k}(i)=k} w_{t,i} \max\big(0,\, \sigma_{i,1} - \sigma_{i,2}\big), \qquad \hat{k}(i) = \arg\max_k \text{sim}(x_{t,i}, s_{t,k}), \tag{2}$$

$$\tilde{u}_{t,k} = u_{t,k} \big/ \max_{k'} u_{t,k'}, \qquad m_{t,k} = \frac{\exp(-\tilde{u}_{t,k}/\tau_m)}{\sum_{k'} \exp(-\tilde{u}_{t,k'}/\tau_m)}. \tag{3}$$

The negative softmax assigns smaller mass to tokens whose removal causes a larger estimated coverage loss. Smaller mass limits the transport assigned to their correspondences, reducing their priority in the coupling-based compression ranking described in Sec. 3.4. Normalizing each frame's total mass to one ensures compatible marginals for cross-frame transport.

3.3 Temporal Compression via Optimal Transport

Let $\mathcal{S}_t = \{s_{t,i}\}_{i=1}^K$ and $\mathcal{S}_{t+1} = \{s_{t+1,j}\}_{j=1}^K$ denote the retained token sets of two neighboring frames, with corresponding mass vectors $m_t, m_{t+1} \in \mathbb{R}_+^K$ constructed in Sec. 3.2. We formulate temporal compression as an optimal transport problem that jointly determines cross-frame correspondences under importance-aware mass constraints and matching cost. These constraints bound each coupling entry by the masses of its source and destination tokens, linking intra-frame importance to cross-frame matching.

Transport cost. We combine semantic dissimilarity with spatial distance to favor compatible cross-frame matches while discouraging confusion between visually similar but spatially distant regions. For retained tokens $s_{t,i}$ and $s_{t+1,j}$, we define

$$c_{\text{sem}}(i,j) = 1 - \text{sim}(s_{t,i}, s_{t+1,j}), \qquad c_{\text{loc}}(i,j) = \frac{\|p_{t,i} - p_{t+1,j}\|_2}{d_{\max}}, \tag{4}$$

where $p_{t,i}$ denotes the patch-grid position and $d_{\max}$ is the grid diagonal length. The transport cost is

$$C_{t,ij} = \alpha_t c_{\text{sem}}(i,j) + (1-\alpha_t) c_{\text{loc}}(i,j), \qquad \alpha_t = 1 - \frac{\bar{s}_t}{2}, \tag{5}$$

where $\bar{s}_t$ is the mean cosine similarity between co-located tokens in the original, pre-pruning frames, clipped to $[0,1]$. High co-located similarity increases the weight of spatial proximity, whereas low similarity reduces reliance on fixed locations to accommodate larger displacements. This rule keeps $\alpha_t \in [1/2, 1]$, so the coefficient on semantic dissimilarity is never smaller than that on spatial distance.

Transport plan and compression difficulty. Given the mass vectors and the cost matrix, we solve an entropically regularized OT problem via Sinkhorn iterations (Cuturi, 2013):

$$P_t = \underset{P \in \Pi(m_t, m_{t+1})}{\text{argmin}} \sum_{i,j} P_{ij}\, C_{t,ij} - \varepsilon H(P), \tag{6}$$

where $\Pi(m_t, m_{t+1})$ denotes all valid transport plans whose rows sum to m_t and columns sum to m_{t+1}, and $H(P) = -\sum_{i,j} P_{ij} \log P_{ij}$ is the entropic regularizer that smooths the solution.

Spatial selection evaluates the additional coverage provided by each token, while mass assignment estimates the coverage lost when the remaining representatives take its place. Both evaluate a token's contribution in the context of what other representatives cover. OT is well suited to this set-level view because it jointly optimizes all cross-frame correspondences under the mass constraints of both frames. The resulting coupling values account for competition over shared counterparts and the availability of alternative matches, allowing compression priorities to reflect how the representative sets align as a whole. The resulting coupling values$P_{t,ij}$ guide token-level compression. Beyond these token-level priorities, we quantify the transport difficulty of each frame pair as the total cost incurred by the plan:

$$W_t = \sum_{i,j} P_{t,ij}\, C_{t,ij}. \tag{7}$$

3.4 Transport-Derived Budget Allocation

Per-pair budget allocation. The transport difficulty W_t quantifies the cost of aligning the retained token sets of neighboring frames under their importance-aware mass constraints. Lower values indicate easier alignment, which we use as a proxy for greater temporal compressibility. We compare $\{W_t\}_{t=1}^{T-1}$ across frame pairs and distribute the total temporal compression budget via a negative softmax with temperature τ_b, assigning larger budgets to pairs with lower difficulty:

$$\beta_t = \exp(-W_t/\tau_b) \Big/ \sum_{t'=1}^{T-1} \exp(-W_{t'}/\tau_b), \qquad B_t = \mathrm{round}(\beta_t B_{\mathrm{tot}}), \tag{8}$$

Compression execution. Given P_t and B_t, we rank cross-frame token pairs by coupling values $P_{t,ij}$ in descending order and select B_t pairs with distinct source tokens from frame $t{+}1$, allowing multiple sources to share a destination in frame t. This budget-constrained selection may include high-cost pairs when low-cost correspondences are insufficient. Merging such pairs can distort the destination's representation by averaging poorly matched token features. We therefore merge a selected source into its destination only if $C_{t,ij} < \tau_c$, and otherwise mark the source for pruning.

All $T{-}1$ transport plans are computed independently from the initial spatially pruned sets $\{\mathcal{S}_t\}$. Only accepted merge edges enter the directed graph, where a token can serve as both a source and a destination across consecutive pairs. We resolve these chains using union-find. Each component follows its root's pruning decision: components with pruned roots are removed, while the original token features in each remaining component are uniformly averaged into a single representative.

4 Experimental Results

Benchmarks. We evaluate on four video question answering (VQA) benchmarks, MVBench (Li et al., 2024b), VideoMME (Fu et al., 2025a), LongVideoBench (Wu et al., 2024), and MLVU (Zhou et al., 2025), and two video temporal grounding (VTG) benchmarks, Charades-STA (Gao et al., 2017) and ActivityNet Captions (Caba Heilbron et al., 2015). For VTG, we use the refined TimeLens annotations (Zhang et al., 2026), referring to these benchmarks as Charades-TimeLens and ActivityNet-TimeLens.

Implementation details. We evaluate four VTG-capable Video-LLMs: Qwen2.5-VL-7B (Bai et al., 2025b), Qwen3-VL-8B (Bai et al., 2025a), InternVL3.5-8B (Wang et al., 2025), and InternVideo3-8B (Yan et al., 2026). For VQA, we uniformly sample up to 32 frames per video. For VTG, we sample at 2 FPS for Qwen2.5-VL and Qwen3-VL, and 1 FPS for InternVL3.5 and InternVideo3. We fix $\gamma = \tau_m = \tau_b = \tau_c = 0.3$ across all backbones, benchmarks, and retention ratios, evaluating 10%, 20%, and 30% visual-token retention. Detailed configurations are provided in Appendix A.

Compared methods. We compare **OTT-Vid** against four state-of-the-art training-free video token compression methods, focusing on outer-LLM compression. These methods offer different strategies for identifying redundancy and retaining informative visual evidence. **FastVID** (Shen et al., 2026) segments videos at scene transitions and applies density-based spatiotemporal pruning within each segment. **UniComp** (Yuan et al., 2026) uses feature dissimilarity to quantify information uniqueness,

Table 1: Main results on Qwen3-VL-8B and Qwen2.5-VL-7B. VQA scores (MVB, VMME, LVB, MLVU) are accuracy and VTG scores (Cha., ANet) are mIoU. Avg. is the mean benchmark-level performance retention (%) for each task. Best results within each backbone are in **bold**.

		Qwen3-VL-8B								**Qwen2.5-VL-7B**							
		VQA					VTG			VQA					VTG		
Method	**Ret.**	MVB	VMME	LVB	MLVU	**Avg.**	Cha.	ANet	**Avg.**	MVB	VMME	LVB	MLVU	**Avg.**	Cha.	ANet	**Avg.**
Vanilla	100%	65.8	63.7	60.6	64.2	100.0	47.4	46.1	100.0	69.0	61.4	58.7	62.8	100.0	36.7	30.7	100.0
FastVID	30%	64.4	62.5	60.5	62.9	98.5	28.6	23.5	55.6	65.4	59.5	57.9	61.9	97.2	25.8	21.6	70.4
UniComp	30%	63.4	62.3	60.1	61.9	97.5	43.0	40.5	89.2	65.8	**61.1**	58.3	62.1	98.3	35.3	25.8	90.1
ST-SimDiff	30%	63.2	59.6	58.4	58.1	94.1	43.3	40.5	89.6	65.4	60.1	57.0	61.8	97.1	**35.8**	26.3	91.5
SemVID	30%	63.6	61.8	**61.6**	63.2	98.4	44.9	44.9	96.0	65.8	59.9	**58.8**	61.1	97.6	35.2	26.5	91.3
OTT-Vid	30%	**65.0**	**63.0**	60.1	**63.3**	**98.9**	**46.6**	**45.3**	**98.3**	**66.9**	60.5	58.7	**62.1**	**98.6**	35.7	**26.9**	**92.4**
FastVID	20%	63.4	**61.9**	59.8	62.1	97.2	28.3	22.9	54.6	64.8	58.0	57.2	60.9	95.7	25.6	20.2	67.7
UniComp	20%	60.2	61.4	57.9	60.6	94.5	40.3	36.7	82.4	65.1	**60.9**	57.7	61.0	97.2	34.4	22.9	84.2
ST-SimDiff	20%	59.5	57.8	56.8	56.7	90.8	40.2	36.9	82.4	63.5	60.1	55.3	60.9	95.3	**35.2**	23.1	85.5
SemVID	20%	61.5	61.1	**61.2**	62.8	97.1	43.9	43.6	93.5	64.9	58.2	**57.7**	60.4	95.9	34.1	23.9	85.5
OTT-Vid	20%	**64.1**	61.4	60.1	**63.1**	**97.8**	**45.4**	**43.7**	**95.3**	**65.9**	60.8	57.6	**61.6**	**97.7**	34.8	**24.2**	**86.9**
FastVID	10%	58.7	59.9	57.5	59.7	92.8	27.0	22.1	52.4	63.2	56.1	55.2	59.7	93.0	24.4	15.9	59.2
UniComp	10%	54.8	58.1	53.3	56.3	87.6	34.4	29.0	67.7	62.3	**58.5**	53.9	57.5	92.2	31.8	16.0	69.4
ST-SimDiff	10%	55.1	54.7	53.9	54.7	85.9	32.2	28.3	64.6	61.7	57.2	52.6	58.4	91.3	31.8	17.9	72.1
SemVID	10%	57.8	57.9	58.5	60.9	92.6	40.3	**40.1**	85.9	63.2	56.2	**56.8**	58.8	93.4	31.4	18.2	72.5
OTT-Vid	10%	**59.9**	**60.4**	**59.5**	**61.1**	**94.8**	**43.0**	39.5	**88.2**	**64.1**	58.3	56.8	**60.9**	**95.4**	**32.2**	**18.4**	**73.9**

guiding frame fusion, token allocation, and spatial token selection. **ST-SimDiff** (Luo et al., 2026) selects representative tokens from a spatiotemporal similarity graph and preserves event tokens identified by temporal feature differences. **SemVID** (Li et al., 2026a) selects object, motion, and context tokens to improve temporal grounding, combining query relevance with temporal feature differences and contextual coverage. Together, these baselines allow us to assess whether OTT-Vid's importance-aware transport formulation better preserves evidence for video understanding under compression. We follow the authors' released implementations, preserving each method's complete compression procedure, and compare performance at matched visual-token retention ratios.

4.1 Comparisons with State-of-the-Art Methods

Main results on Qwen3-VL and Qwen2.5-VL. Table 1 shows that OTT-Vid achieves the highest average VQA and VTG retention at all three token retention ratios on both Qwen backbones. At 10% token retention, it preserves 94.8%/88.2% of VQA/VTG performance on Qwen3-VL and 95.4%/73.9% on Qwen2.5-VL, improving VQA retention over the strongest baseline by 2.0 percentage points on both.

The advantage extends to temporal grounding, where accurate localization requires preserving temporal evidence under compression. OTT-Vid outperforms SemVID (Li et al., 2026a), a method designed for temporal grounding, in average VTG retention at every ratio on both backbones. These results support the effectiveness of importance-aware temporal compression and transport-derived budget allocation in maintaining localization performance alongside video understanding.

Generalization across Video-LLM backbones. We extend evaluation to two additional VTG-capable backbones, InternVideo3-8B and InternVL3.5-8B, to assess generalization on both tasks. As shown in Table 2, OTT-Vid achieves the highest average performance retention in 10 of the 12 backbone-task-budget settings. At 10% token retention, it leads on both VQA and VTG across both backbones, retaining 92.6% and 86.4% of VTG performance on InternVideo3 and InternVL3.5, respectively. These results support OTT-Vid's broad applicability across Video-LLMs with different architectures and input configurations.

4.2 Ablation Studies

Ablation on core components. Table 3 evaluates OT matching, importance-aware mass, and difficulty-based budget allocation on Qwen3-VL-8B at 10% token retention. We isolate the value of OT's global coordination against independent assignments that also consider token importance

Table 2: Cross-backbone results on InternVideo3-8B and InternVL3.5-8B. VQA scores (MVB, VMME, LVB, MLVU) are accuracy and VTG scores (Cha., ANet) are mIoU. Avg. is the mean benchmark-level performance retention (%) for each task. Best results are in **bold**.

		InternVideo3-8B								InternVL3.5-8B							
		VQA					VTG			VQA					VTG		
Method	**Ret.**	MVB	VMME	LVB	MLVU	**Avg.**	Cha.	ANet	**Avg.**	MVB	VMME	LVB	MLVU	**Avg.**	Cha.	ANet	**Avg.**
Vanilla	100%	73.8	64.3	57.8	65.2	100.0	40.4	35.2	100.0	74.3	65.0	52.1	64.7	100.0	33.4	29.8	100.0
FastVID	30%	71.3	**63.5**	56.8	64.3	98.0	34.5	25.9	79.5	68.1	62.3	50.3	**63.4**	95.5	20.3	23.4	69.8
UniComp	30%	71.1	62.4	56.5	64.0	97.3	39.5	31.2	93.2	**71.3**	62.4	50.3	62.7	96.4	29.8	28.4	92.3
ST-SimDiff	30%	70.6	62.6	56.8	63.5	97.2	40.0	32.9	96.3	65.5	63.2	50.8	62.6	94.9	29.1	27.0	88.9
SemVID	30%	71.3	63.2	**58.0**	**65.5**	**98.9**	41.0	33.9	98.9	70.0	61.5	**51.2**	62.9	96.1	**31.5**	**30.0**	**97.4**
OTT-Vid	30%	**73.3**	63.3	57.2	64.2	98.8	**42.5**	**35.0**	**102.4**	69.5	**63.6**	50.9	63.0	**96.6**	31.3	29.6	96.7
FastVID	20%	71.3	62.4	56.4	63.2	97.1	33.9	25.1	77.7	67.5	61.1	50.0	62.7	94.5	19.6	23.1	68.1
UniComp	20%	69.9	62.2	55.0	62.3	95.5	37.4	28.5	86.8	**69.2**	60.7	47.3	59.9	92.5	27.1	25.6	83.5
ST-SimDiff	20%	69.0	61.7	55.8	61.7	95.2	37.2	30.0	88.7	63.8	62.1	50.0	61.1	93.0	27.2	25.3	83.3
SemVID	20%	68.9	61.7	**56.9**	**64.4**	96.6	39.6	32.4	95.1	67.7	60.6	**51.1**	61.6	94.4	**30.9**	28.8	94.6
OTT-Vid	20%	**73.2**	**62.5**	56.1	64.1	**98.0**	**42.2**	**34.0**	**100.6**	67.3	**62.3**	50.3	**63.2**	**95.2**	30.7	**29.0**	**94.7**
FastVID	10%	69.9	**61.0**	55.3	61.5	94.9	30.7	22.9	70.6	65.0	58.6	49.1	61.1	91.6	18.5	21.7	64.1
UniComp	10%	65.5	60.0	50.8	59.6	90.3	29.4	22.4	68.3	64.9	57.0	43.9	55.5	86.3	19.4	19.2	61.2
ST-SimDiff	10%	65.5	59.2	53.2	58.7	90.7	31.7	24.2	73.6	61.6	59.8	47.1	58.5	88.9	21.7	21.2	68.0
SemVID	10%	65.0	60.6	54.2	62.7	93.1	34.8	28.2	83.2	**65.3**	57.0	49.0	59.4	90.4	**27.5**	26.6	85.8
OTT-Vid	10%	**71.2**	60.6	**55.4**	**63.9**	**96.2**	**38.9**	**31.2**	**92.6**	65.3	**59.9**	**49.4**	**61.4**	**92.5**	27.1	**27.3**	**86.4**

Table 3: Component ablation on Qwen3-VL-8B at 10% token retention. Scores are mean performance retention (%).

	OT Match	Mass	Budget	VQA Avg.	VTG Avg.
(a)	×	✓	×	93.4	85.3
(b)	×	✓	✓	93.9	86.1
(c)	✓	×	×	93.1	85.9
(d)	✓	×	✓	93.7	86.6
(e)	✓	✓	×	94.3	87.3
(f)	✓	✓	✓	**94.8**	**88.2**

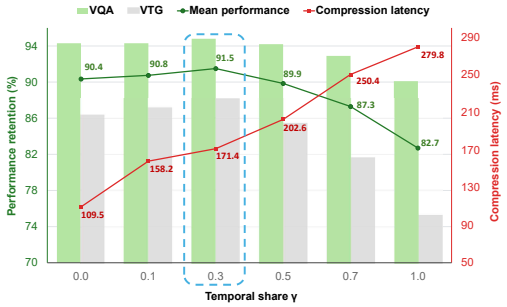

Figure 3: Ablation on temporal share γ on Qwen3-VL-8B at 10% retention.

and matching cost. ✓ indicates that the corresponding component is enabled, while × denotes Local Gibbs matching, uniform mass, or uniform allocation. Local Gibbs uses the same masses, costs, and temperature as OT but computes assignments independently for each source, preserving source marginals without enforcing destination marginals. Its difficulty-based allocation uses $W_t^{\text{local}} = \sum_{i,j} Q_{t,ij} C_{t,ij}$, where the local coupling is defined as

$$Q_{t,ij} = m_{t+1,j} \frac{m_{t,i} \exp(-C_{t,ij}/\varepsilon)}{\sum_{i'} m_{t,i'} \exp(-C_{t,i'j}/\varepsilon)}. \tag{9}$$

Under identical mass and uniform allocation, OT improves VQA and VTG retention over Local Gibbs by 0.9 and 2.0 percentage points, respectively, comparing (a) with (e). This supports globally coordinated matching beyond independent importance-aware assignments. Importance-aware mass improves both tasks under either budget scheme, while difficulty-based allocation benefits all three configurations. Their combination (f) achieves the highest retention, with larger gains on VTG.

Ablation on temporal share γ. Figure 3 examines the spatial–temporal balance at 10% retention. Increasing γ leaves more tokens for OT to match, raising compression latency from 109.5 to 279.8 ms. At $\gamma = 0.3$, mean VQA/VTG retention peaks at 91.5% with 171.4 ms overhead, compared with 90.4% for spatial-only and 82.7% for temporal-only compression. Spatial pruning before OT thus improves both performance and efficiency over temporal-only compression, with $\gamma = 0.3$ offering the best observed trade-off.

Additional ablations on τ_b, τ_m, τ_c, and the mixing coefficient α_t are provided in Appendix D.

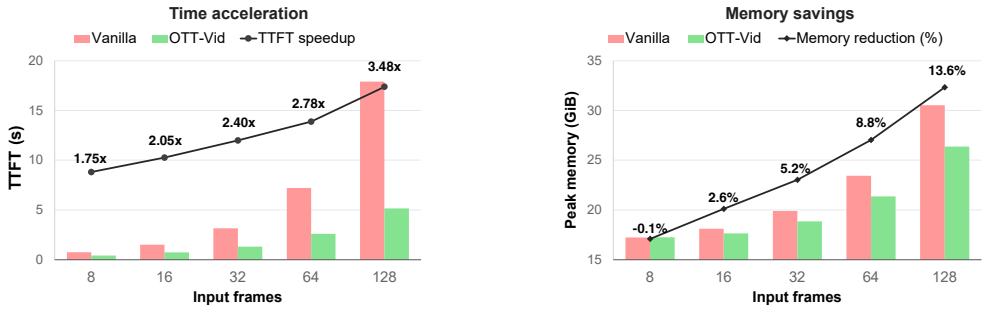

Figure 4: Efficiency scaling on Qwen3-VL-8B at 10% token retention. Bars compare Vanilla and OTT-Vid across 8–128 input frames. Lines show TTFT speedup (left) and peak-memory reduction relative to Vanilla (right). All measurements use the same video and hardware configuration.

Table 4: Efficiency with 32 input frames at 10% token retention on Qwen3-VL-8B. TTFT measures time to the first generated token, and E2E measures latency for generating 32 new tokens. Latencies are averaged over 10 measured runs after 3 warm-up runs.

Method	Latency (ms)					Peak Mem. (GiB)	TFLOPs	Perf. (%)	
	Vision	Comp.	LLM	TTFT	E2E			VQA	VTG
Vanilla	958.7	–	2193.2	3151.9 (1.0×)	5223.4 (1.0×)	19.89	252.00	100.0	100.0
FastVID	958.7	**84.9**	185.5	**1229.1 (2.6×)**	**3230.0 (1.6×)**	**18.85**	74.72	92.8	52.4
UniComp	958.7	669.3	186.2	1814.2 (1.7×)	3759.2 (1.4×)	**18.85**	**74.67**	87.6	67.7
ST-SimDiff	958.7	410.3	263.5	1632.5 (1.9×)	3654.6 (1.4×)	18.90	81.61	85.9	64.6
SemVID	958.7	271.4	185.3	1415.4 (2.2×)	3423.9 (1.5×)	**18.85**	74.71	92.6	85.9
OTT-Vid	958.7	171.4	**184.5**	1314.6 (2.4×)	3308.8 (1.6×)	**18.85**	74.84	**94.8**	**88.2**

4.3 Efficiency Analysis

At 32 input frames, Table 4 shows that OTT-Vid accelerates TTFT by $2.4\times$ and end-to-end generation by $1.6\times$ over uncompressed Qwen3-VL-8B. Among the compared methods, it achieves the second-lowest TTFT and E2E latency, trailing FastVID by only 85.5 and 78.8 ms, respectively, while delivering the highest VQA and VTG retention. These results demonstrate a favorable balance between inference efficiency and retained performance. Figure 4 compares OTT-Vid with uncompressed Qwen3-VL-8B as the input grows from 8 to 128 frames. Both TTFT speedup and memory savings increase with frame count, reaching $3.48\times$ acceleration and 13.6% peak-memory reduction at 128 frames. These results demonstrate the growing efficiency benefits of compression for longer visual sequences.

5 Conclusion

We presented OTT-Vid, a training-free video token compression framework that integrates intra-frame token importance and cross-frame correspondence through optimal transport. The transport plan guides token-level compression, while its associated difficulty informs budget allocation across neighboring frame pairs. Experiments on six benchmarks across four Video-LLM backbones demonstrate strong performance on video question answering and temporal grounding at multiple retention ratios. These results support the effectiveness of our transport formulation in preserving video understanding capabilities, including the temporal evidence needed for event localization.

Limitations and Future Work. OTT-Vid computes all pairwise transport plans in parallel from the initial spatially pruned representations, enabling single-pass budget allocation but overlooking representational changes induced by compression across consecutive pairs. Moreover, frame-level importance may underestimate query-relevant details, such as text or small objects, potentially discarding evidence needed to answer specific questions. Future work could explore efficient transport-plan updates and query-conditioned importance estimation to address these limitations.

AI USE STATEMENT

In this work, we used generative AI tools to provide feedback on the design of our ablation experiments. We have not used generative AI tools to develop core theoretical models or conceptual frameworks, formulate or prove mathematical claims, implement experimental methods, generate synthetic datasets, or interpret experimental results, and other mandatory disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for literature retrieval and discovery (e.g., identifying related prior work) and for editing the manuscript to improve readability and grammatical precision.

All primary research ideas, novel conceptual contributions, and overall experimental frameworks were conceived and designed independently by the authors without AI assistance. The AI tools were consulted solely as an auxiliary aid to obtain feedback on potential ablation configurations. The final selection of ablation studies was made entirely by the authors, and all experiments were implemented, executed, and evaluated by the authors without AI involvement. Furthermore, all candidate references identified through AI tools were manually verified against the original publications, and all AI-assisted text revisions were thoroughly reviewed and edited by the authors to ensure scientific accuracy and prevent misrepresentation. We take full responsibility for the final content of this work, including all text, claims, and results presented herein.

REPRODUCIBILITY STATEMENT

The method section specifies spatial token selection, importance-aware mass assignment, transport optimization, budget allocation, and compression execution. Appendix A documents model checkpoints, video sampling, inference and evaluation protocols, hyperparameters, and solver settings. Additional comparison settings, including inner-pruning schedules and the AOT adaptation to Qwen2.5-VL, are described in Appendix E.

REFERENCES

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. Divprune: Diversity-based visual token pruning for large multimodal models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 9392–9401, 2025.

Xiang An, Yin Xie, Feilong Tang, Yunyao Yan, Huajie Tan, Didi Zhu, Changrui Chen, Xiuwei Zhao, Bin Qin, Kaicheng Yang, et al. Llava-onevision-2: Towards next-generation perceptual intelligence. *arXiv preprint arXiv:2605.25979*, 2026.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025b.

Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pp. 961–970, 2015.

Qingqing Cao, Bhargavi Paranjape, and Hannaneh Hajishirzi. Pumer: Pruning and merging tokens for efficient vision language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 12736–12746, 2023.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision (ECCV)*, 2024.

Janghoon Cho, Jungsoo Lee, Munawar Hayat, Kyuwoong Hwang, Fatih Porikli, and Sungha Choi. Floc: Facility location-based efficient visual token compression for long video understanding. In *International Conference on Learning Representations*, volume 2026, pp. 102973–102996, 2026.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

Jinhong Deng, Wen Li, Joey Tianyi Zhou, and Yang He. Scope: Saliency-coverage oriented token pruning for efficient multimodel llms. *Advances in Neural Information Processing Systems*, 38: 161527–161552, 2026.

Mohamed Dhouib, Davide Buscaldi, Sonia Vanier, and Aymen Shabou. Pact: Pruning and clustering-based token reduction for faster visual language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14582–14592, 2025.

Junhao Du, Jialong Xue, Anqi Li, Jincheng Dai, and Guo Lu. Unified spatiotemporal token compression for video-llms at ultra-low retention. *arXiv preprint arXiv:2603.21957*, 2026.

Ziyang Fan, Keyu Chen, Ruilong Xing, Yulin Li, Li Jiang, and Zhuotao Tian. Flashvid: Efficient video large language models via training-free tree-based spatiotemporal token merging. *arXiv preprint arXiv:2602.08024*, 2026.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24108–24118, 2025a.

Tianyu Fu, Tengxuan Liu, Qinghao Han, Guohao Dai, Shengen Yan, Huazhong Yang, Xuefei Ning, and Yu Wang. Framefusion: Combining similarity and importance for video token reduction on large vision language models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22654–22663. IEEE, 2025b.

Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE international conference on computer vision*, pp. 5267–5275, 2017.

Xiaohu Huang, Hao Zhou, and Kai Han. Prunevid: Visual token pruning for efficient video large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 19959–19973, 2025.

Jeongseok Hyun, Sukjun Hwang, Su Ho Han, Taeoh Kim, Inwoong Lee, Dongyoon Wee, Joon-Young Lee, Seon Joo Kim, and Minho Shim. Multi-granular spatio-temporal token merging for training-free acceleration of video llms. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Shaobo Ju, Baiyang Song, Tao Chen, Jiapeng Zhang, Qiong Wu, Chao Chang, HuaiXi Wang, Yiyi Zhou, and Rongrong Ji. Forestprune: High-ratio visual token compression for video multimodal large language models via spatial-temporal forest modeling. *arXiv preprint arXiv:2603.22911*, 2026.

Minchul Kim, Shangqian Gao, Yen-Chang Hsu, Yilin Shen, and Hongxia Jin. Token fusion: Bridging the gap between token pruning and token merging. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1383–1392, 2024.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024a.

Jiaqi Li, Shuntian Zheng, Yixian Shen, Jia-Hong Huang, Xiaoman Lu, Minzhe Ni, and Yu Guan. Keeping the evidence chain: Semantic evidence allocation for training-free token pruning in video temporal grounding. In *European Conference on Computer Vision*, pp. 549–567. Springer, 2026a.

Jinlong Li, Liyuan Jiang, Haonan Zhang, and Nicu Sebe. Token reduction via local and global contexts optimization for efficient video large language models. *arXiv preprint arXiv:2603.01400*, 2026b.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22195–22206, 2024b.

Bingjun Luo, Tony Wang, Chaoqi Chen, and Xinpeng Ding. St-simdiff: Balancing spatiotemporal similarity and difference for efficient video understanding with mllms. *arXiv preprint arXiv:2605.22158*, 2026.

Kele Shao, Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. Holitom: Holistic token merging for fast video large language models. *Advances in Neural Information Processing Systems*, 38:135547–135570, 2026.

Leqi Shen, Guoqiang Gong, Tao He, Yifeng Zhang, Pengzhang Liu, Sicheng Zhao, et al. Fastvid: Dynamic density pruning for fast video large language models. *Advances in Neural Information Processing Systems*, 38:123553–123581, 2026.

Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. Dycoke: Dynamic compression of tokens for fast video large language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 18992–19001, 2025.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. *Advances in Neural Information Processing Systems*, 37:28828–28857, 2024.

Ziang Yan, Sheng Xia, Jiashuo Yu, Yue Wu, Tianxiang Jiang, Songze Li, Kanghui Tian, Yicheng Xu, Yinan He, Kai Chen, et al. Internvideo3: Agentify foundation models with multimodal contextual reasoning. *arXiv preprint arXiv:2606.12195*, 2026.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19792–19802, 2025.

Chao Yuan, Shimin Chen, Minliang Lin, Limeng Qiao, Guanglu Wan, and Lin Ma. Unicomp: Rethinking video compression through informational uniqueness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18609–18618, 2026.

Jun Zhang, Teng Wang, Yuying Ge, Yixiao Ge, Xinhao Li, and Limin Wang. Timelens: Rethinking video temporal grounding with multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10419–10429, 2026.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Llava-video: Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, et al. Mlvu: Benchmarking multi-task long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13691–13701, 2025.

A Additional Implementation Details

Models and inference environment. We evaluate four instruction-tuned models. The checkpoints are `Qwen2.5-VL-7B-Instruct`, `Qwen3-VL-8B-Instruct`, `InternVL3_5-8B-Instruct`, and `InternVideo3-8B-Instruct`. Experiments use BF16 inference with FlashAttention-2 and batch size 1 on NVIDIA RTX A6000 GPUs with 48 GiB memory. Each evaluation worker runs on a single GPU, with independent jobs optionally executed concurrently. The software environment uses Python 3.10.20, PyTorch 2.7.0 with CUDA 12.8, Transformers 4.57.3, FlashAttention 2.8.3, and Decord 0.6.0. All methods share the same evaluation pipeline within each backbone and benchmark.

Video sampling. Frame counts below refer to decoded RGB frames before temporal patching. For VQA, Qwen2.5-VL samples at most 32 frames uniformly over the full video, while Qwen3-VL uses nominal 2-FPS sampling capped at 32 frames. InternVL3.5 samples 32 uniform segment-midpoint frames, and InternVideo3 samples 32 uniformly spaced full-span frames. For VTG, both Qwen backbones use nominal 2-FPS sampling capped at 768 frames. InternVL3.5 and InternVideo3 use 1 FPS with caps of 32 and 128 frames, respectively. Sequences exceeding these caps are uniformly subsampled over the full duration. Qwen2.5-VL, Qwen3-VL, and InternVideo3 use temporal patch size 2, so 32 source frames normally yield 16 temporal groups. In our formulation, T denotes the number of temporal groups presented to the compressor.

OTT-Vid and Sinkhorn settings. Unless explicitly ablated, we fix $\gamma = \tau_m = \tau_b = \tau_c = 0.3$ across backbones, benchmarks, and retention ratios. For target retention r, spatial retention is $r_s = r^{1-\gamma}$ and temporal retention is $r_t = r/r_s$. At $r = 0.1$, these are approximately 0.1995 and 0.5012, respectively. We use saliency-weighted coverage selection, terminal leave-one-out mass scores, transport-difficulty-based budget allocation, and adaptive feature-position cost mixing. The entropically regularized OT problem in Eq. (6) is solved using log-domain Sinkhorn updates with $\varepsilon = 0.01$ and exactly 200 iterations, without early stopping. Compression candidates are ranked by raw coupling values $P_{t,ij}$ without marginal normalization. Each source is selected at most once, while multiple sources may share a destination. Merged components use the arithmetic mean of their original member features rather than a transport-mass-weighted average. We record both requested and realized retention because integer allocation and minimum-survivor constraints can produce deviations on short clips.

Baseline adaptation and retention matching. We preserve the baseline algorithms and their released numerical settings where available, adapting feature layouts, positional handling, and model hooks to each backbone. We distinguish these architecture-matched ports from author-native implementations. For the main comparison, original retention or compute-budget settings are replaced with target final visual-token retention ratios of 10%, 20%, and 30%.

FastVID and UniComp. FastVID (Shen et al., 2026) follows the authors' Qwen2.5-VL implementation with `DySeg_c`=8, `DySeg_tau`=0.84, `DySeg_ignore`=0.95, `STPrune_d`=0.4, `DTM_p`=4, and `DTM_beta`=0.6. The same settings are used in the other backbone adapters, with a top-k guard for short clips. UniComp (Yuan et al., 2026) uses frame and spatial uniqueness thresholds $U_f = 0.005$ and $U_c = 0.2$, `auto_compress=False`, and `min_interval`=$\lceil 1/r \rceil$. Its LLaVA implementation is adapted to projected post-merger features, replacing `vision_abstract` with the frame mean and omitting LLaVA-specific newline tokens and redundant spatial pooling. We use its feature-similarity fallback where the required vision-attention input is unavailable. Fused frames inherit the temporal position of the selected representative.

ST-SimDiff. ST-SimDiff (Luo et al., 2026) uses the released `new_topk` strategy-3 design with similarity threshold 0.8 and event upper bound 0.2. Graph pruning is applied before decoder layer 0. Decoder layers 0 and 1 process the graph-retained sequence, and attention from layer 1 guides additional pruning to reach the final token target. Layers 2 onward process the resulting reduced sequence. We replace the original decoder-compute budget with a final-token retention target, so the reported retention ratios do not denote identical decoder computation across methods.

SemVID. SemVID (Li et al., 2026a) retains query-aware allocation and structured object, motion, and context selection. Its query embedding is recomputed for each prompt, while projected video features are cached. Qwen3-VL uses the released shared Qwen3 profile with object ratio 0.6 and MMR coefficient 0.9. Qwen2.5-VL uses the VQA profile with object ratio 0.4 and MMR coefficient 0.3, and the VTG profile with `topk_segments`=2, object ratio 0.8, and MMR coefficient 0.5, retaining its temporal-anchor settings. InternVideo3 and InternVL3.5 use an architecture-matched port of the Qwen3 profile.

Generation and scoring. All evaluations use greedy decoding. Qwen3-VL explicitly disables thinking mode, and the other backbones use their standard instruction-following evaluation prompts without an added chain-of-thought instruction. Maximum generation lengths for VQA/VTG are 128/128 tokens for Qwen2.5-VL, 200/200 for Qwen3-VL, 16/50 for InternVL3.5, and 16/48 for InternVideo3. VQA reports multiple-choice accuracy or the benchmark's equivalent overall score, and VTG reports mean temporal intersection-over-union. The evaluated sets contain 1337 LongVideoBench questions, 2700 VideoMME questions, 4000 MVBench questions, 2174 MLVU-dev questions, 3363 Charades-STA queries, and 4500 ActivityNet-TimeLens queries. Mean performance retention is computed by normalizing each benchmark score by its corresponding uncompressed baseline before averaging within each task family.

B Spatial Compression Analysis

B.1 Comparison with Alternative Spatial Methods

We compare five spatial pruning methods within the OTT-Vid framework while keeping the OT-based temporal compression fixed. Each method combines a token selection algorithm with a corresponding mass function for the temporal stage. Transport mass is derived from the selection criterion through a negative softmax with $\tau_m = 0.3$, assigning smaller mass to higher-scoring tokens. This comparison evaluates each spatial selector together with its associated mass construction.

Top-K. Selection ranks tokens by their saliency w_j and retains the top K. Mass is assigned directly from saliency: $m_j \propto \exp(-w_j/\tau_m)$. Diversity or coverage are not considered, so selected tokens may concentrate within a few semantic clusters.

DivPrune (Alvar et al., 2025). Selection uses max-min farthest-point sampling on the cosine distance matrix without considering saliency: each subsequent token maximizes its minimum cosine distance to the already selected set. Mass is derived from the same isolation quantity that drives selection, $\text{iso}_j = \min_{k \in S \setminus \{j\}}(1 - \cos(j,k))$, with $m_j \propto \exp(-\text{iso}_j/\tau_m)$.

ADTS (Fan et al., 2026). The spatial pruning component of FlashVID combines diversity and saliency through saliency-scaled distances: each candidate's distance to the selected set is multiplied by its saliency before max-min selection. Mass uses the same saliency-scaled isolation, $\text{iso}_j = w_j \cdot \min_{k \in S \setminus \{j\}}(1 - \cos(j,k))$.

SCOPE (Deng et al., 2026). Selection performs greedy coverage maximization with a coverage gain weighted by the candidate's own saliency: $\text{gain}(j) = w_j \cdot \sum_i \max(0, \cos(i,j) - \text{cur_max}_i)$. Mass uses a leave-one-out form of the same expression, measuring how much coverage is lost when j is removed from the selected set, scaled by w_j.

OTT-Vid (Ours). Selection greedily picks tokens that maximize a coverage gain weighted by the saliency of the covered tokens rather than the candidate: $\text{gain}(j) = \sum_i w_i \cdot \max(0, \cos(i,j) - \text{cur_max}_i)$. While SCOPE (Deng et al., 2026) incorporates saliency explicitly via the candidate weight w_j, our formulation incorporates it implicitly through the covered tokens. As a result, the iterative selection naturally prioritizes salient regions in the early steps where their weighted gain is largest, and shifts toward under-covered regions in later steps as the salient regions saturate. Mass follows the same leave-one-out form, measuring the loss in coverage of salient tokens when j is removed.

Table A1: Comparison of spatial pruning methods within the OTT-Vid framework on Qwen3-VL-8B at 10% retention. Each method uses its corresponding mass assignment with the same OT-based temporal compression procedure. VQA scores are accuracy and VTG scores are mIoU. Avg. (%) denotes mean performance retention relative to the uncompressed baseline. Best results are in **bold**.

Spatial Pruning Method	Ret.	Video Question Answering					Video Temporal Grounding		
		MVBench	VideoMME	LVB	MLVU	Avg. (%)	Charades	ANet	Avg. (%)
Saliency	10%	**60.0**	59.4	58.6	60.2	93.8	41.9	36.2	83.5
DivPrune (Alvar et al., 2025) CVPR'25	10%	58.7	59.3	57.0	60.4	92.6	36.7	36.8	78.6
SCOPE (Deng et al., 2026) NeurIPS'25	10%	59.7	**61.0**	59.0	60.6	94.6	42.4	**39.8**	87.9
ADTS (Fan et al., 2026) ICLR'26	10%	59.9	59.9	57.8	60.5	93.7	41.7	37.0	84.1
OTT-Vid	10%	59.9	60.4	**59.5**	**61.1**	**94.8**	**43.0**	39.5	**88.2**

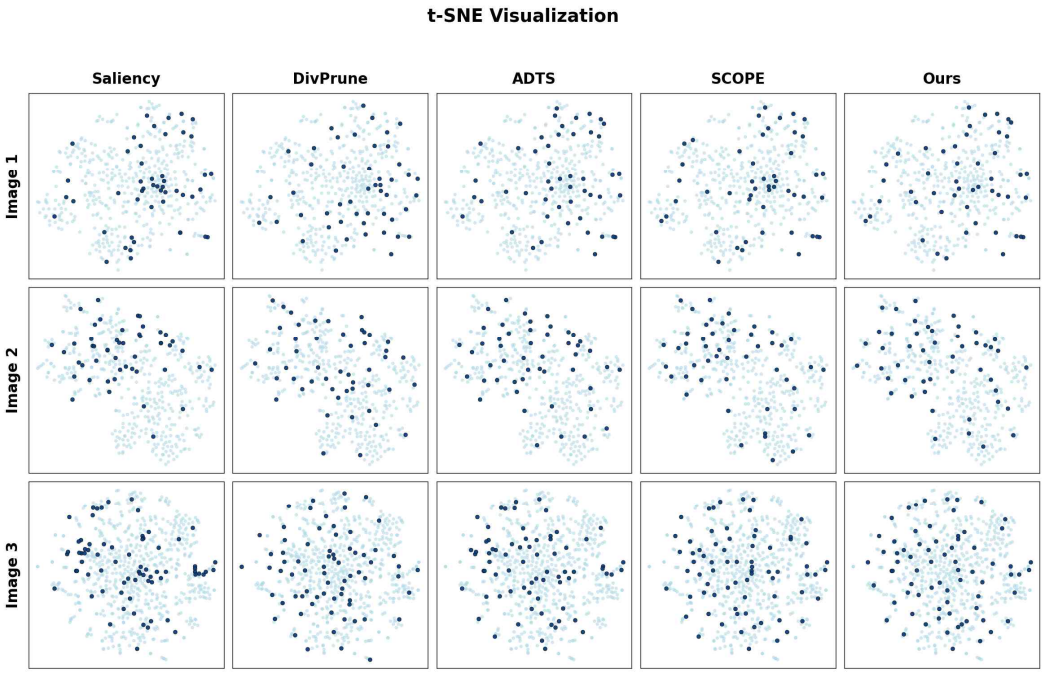

Figure A1: t-SNE visualization of selected tokens (dark blue) versus all tokens (light blue) for the five spatial pruning methods on three images at r=10% retention on Qwen3-VL. Top-K concentrates within a few feature-space clusters, DivPrune spreads uniformly without prioritization, and our saliency-and-coverage formulation produces a selection that spans the full distribution while maintaining density on salient regions.

Comparison results. Table A1 compares spatial selectors and their associated mass assignments within the same temporal OT framework on Qwen3-VL-8B at 10% final token retention. OTT-Vid achieves the highest mean performance retention on both VQA (94.8%) and VTG (88.2%). SCOPE performs closely, retaining 94.6% and 87.9%, respectively, and achieves the highest ActivityNet score. Differences across configurations are more pronounced on VTG, where retention ranges from 78.6% to 88.2%, compared with 92.6% to 94.8% on VQA. Saliency-only selection remains competitive on VQA but yields lower VTG retention (83.5%), while diversity-only selection performs worst on both task averages. These results suggest that combining saliency with coverage provides an effective intra-frame representation and importance signal for subsequent temporal compression. Since each selector uses its corresponding mass assignment, the comparison evaluates their combined effect.

B.2 Intra-frame Signals and Temporal Matching

Experimental setup. Table A2 examines whether the benefit of globally coordinated temporal matching depends on the intra-frame representation used for compression. On Qwen3-VL-8B at

Table A2: Spatial design and temporal matching ablation on Qwen3-VL-8B at 10% token retention with uniform temporal budgets. Spatial methods and their paired mass assignments follow Table A1. VQA scores are accuracy and VTG scores are mIoU. Avg. (%) denotes mean performance retention relative to the uncompressed baseline. Best results are in **bold**.

Spatial Pruning Method	Matching	Video Question Answering					Video Temporal Grounding		
		MVBench	VideoMME	LVB	MLVU	Avg. (%)	Charades	ANet	Avg. (%)
Saliency	Local Gibbs	58.9	59.2	**58.9**	59.7	93.2	41.5	36.1	82.9
Saliency	Global OT	**60.1**	59.7	57.9	60.0	93.5	41.6	35.9	82.8
OTT-Vid	Local Gibbs	59.6	**60.1**	56.9	60.7	93.4	41.5	38.3	85.3
OTT-Vid	Global OT	60.0	60.0	58.8	**60.7**	**94.3**	**42.6**	**39.1**	**87.3**

10% token retention, we cross the Saliency and OTT-Vid configurations from Table A1 with Local Gibbs and global OT matching. Saliency uses attention-based top-K selection and attention-derived mass, while OTT-Vid uses saliency-weighted coverage selection and leave-one-out mass. We fix temporal budget allocation to uniform to exclude the effect of difficulty-based allocation. Within each spatial configuration, both matching schemes use the same retained tokens, masses, cost matrix, and temperature. Local Gibbs preserves source marginals but computes assignments independently across sources, whereas OT also enforces destination marginals. The experiment therefore tests the interaction between temporal matching and the combined spatial-selection and mass design.

Results. With Saliency, replacing Local Gibbs with OT changes VQA retention from 93.2% to 93.5% and VTG retention from 82.9% to 82.8%. With the OTT-Vid spatial and mass design, the same replacement increases VQA retention from 93.4% to 94.3% and VTG retention from 85.3% to 87.3%. The larger OT gains under the coverage-and-LOO configuration support a positive interaction between this intra-frame design and globally coordinated matching, particularly for temporal grounding. This pattern is consistent with our motivation: coverage selection and LOO mass characterize representatives relative to the other tokens in a frame, while OT coordinates their cross-frame correspondences under both sets of marginal constraints. The combined configuration achieves the highest average retention on both task families.

C Adaptive Behavior Analysis

C.1 Budget Feasibility and Overflow Analysis

The pairwise budget allocation distributes the global budget B_{tot} across $T-1$ adjacent frame pairs via a softmax over transport difficulties, with no inherent upper bound on the per-pair budget B_t. However, each pair $(t, t+1)$ supports at most K compression operations under the constraint that each source token in S_{t+1} is selected at most once, giving $B_t \leq K$. We refer to the condition $B_t > K$ as *budget overflow*, in which the allocation exceeds the per-pair feasibility cap and the excess operations cannot be executed. In this subsection, we describe the mechanism handling overflow, derive its theoretical condition, and empirically verify its frequency under our default configuration.

Theoretical overflow condition. The uniform per-pair budget is $B_{\text{uniform}} = B_{\text{tot}}/(T-1) = KT(1-r_t)/(T-1)$. Defining the budget ratio $\rho_t := B_t/B_{\text{uniform}}$, the overflow condition $B_t > K$ becomes

$$\rho_t > \rho^* := \frac{T-1}{T(1-r_t)}, \tag{A1}$$

which converges to $1/(1-r_t)$ as $T \to \infty$. For the 64-input-frame Qwen3-VL configuration used in our overflow analysis, temporal patching yields $T = 32$ groups. With $r = 0.10$ and $\gamma = 0.3$, $r_t = r^\gamma \approx 0.501$ gives $\rho^* \approx 1.94$, meaning a pair must receive nearly twice the uniform budget before exceeding the feasibility cap.

C.2 Per-Benchmark Budget Distribution

Per-benchmark budget variability. To characterize how non-uniformly OTT-Vid distributes the budget within a single video, we compute the coefficient of variation CV $= \sigma(B_t)/\mu(B_t)$,

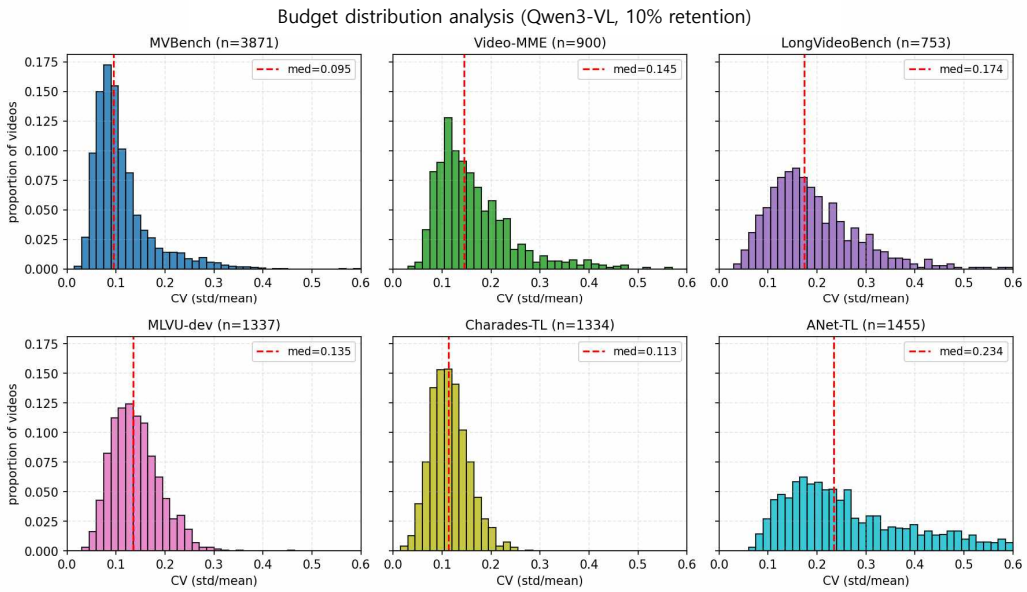

Figure A2: Per-video coefficient of variation CV $= \sigma(B_t)/\mu(B_t)$ of pairwise budgets across six benchmarks on Qwen3-VL at r=0.10 and τ_b=0.3. Red dashed lines mark medians.

where σ and μ denote the standard deviation and mean of the pairwise budgets $\{B_t\}_{t=1}^{T-1}$. A CV of 0 corresponds to a uniform allocation, while larger CV indicates that the adaptive mechanism concentrates budget on a smaller subset of frame pairs. Figure A2 reports the distribution of per-video CV across the six benchmarks on Qwen3-VL at r=0.10 and τ_b=0.3. The median CV ranges from 0.095 (MVBench) to 0.234 (ANet-TL), reflecting substantial differences in the temporal compressibility of each benchmark. MVBench and Charades-TL exhibit the lowest median CV (0.095 and 0.113), with most videos producing nearly uniform budget allocations. MLVU-dev (0.135) and Video-MME (0.145) form an intermediate group, while LongVideoBench (0.174) and ANet-TL (0.234) show substantially higher variability, with a long tail of videos exceeding CV 0.4.

C.3 Per-LLM Budget Distribution

Cross-backbone budget distribution and overflow. Figure A3 shows the pooled distribution of pairwise budget ratios $\rho_t = B_t/B_{\text{uniform}}$ across three Video-LLM backbones and four VQA benchmarks at $r = 0.10$ and $\tau_b = 0.3$. Here, $\rho_t = 1$ denotes uniform allocation, while smaller and larger values indicate less and more compression, respectively. For this analysis, we use 64 input frames per video. Temporal patching produces $T = 32$ groups for Qwen3-VL and InternVideo3, whereas InternVL3.5 retains $T = 64$. The corresponding overflow thresholds from Eq. (A1) are approximately 1.94 and 1.97, respectively, as indicated by the red dashed lines. Across all twelve backbone-benchmark combinations, the empirical overflow rate remains below 1%, reaching at most 0.52%. Most pairs fall within $\rho_t \in [0.5, 1.5]$, suggesting that the initial allocation is typically feasible and budget redistribution is infrequently required under this setting.

Handling overflow. During budget allocation process, if the initial softmax produces $B_t > K$ for some pair, the excess is redistributed through an iterative procedure that recomputes the softmax over the active set of pairs below their cap. The allocated amount is applied up to the headroom $K - B_t$ of each pair, with any excess carried to the next iteration. Pairs reaching K are frozen and excluded thereafter. This preserves the softmax-based prioritization within the feasible region and guarantees $\sum_t B_t = B_{\text{tot}}$ exactly. As Figure A3 shows, overflow is rare in practice, and most allocations terminate after the initial softmax step on the full set.

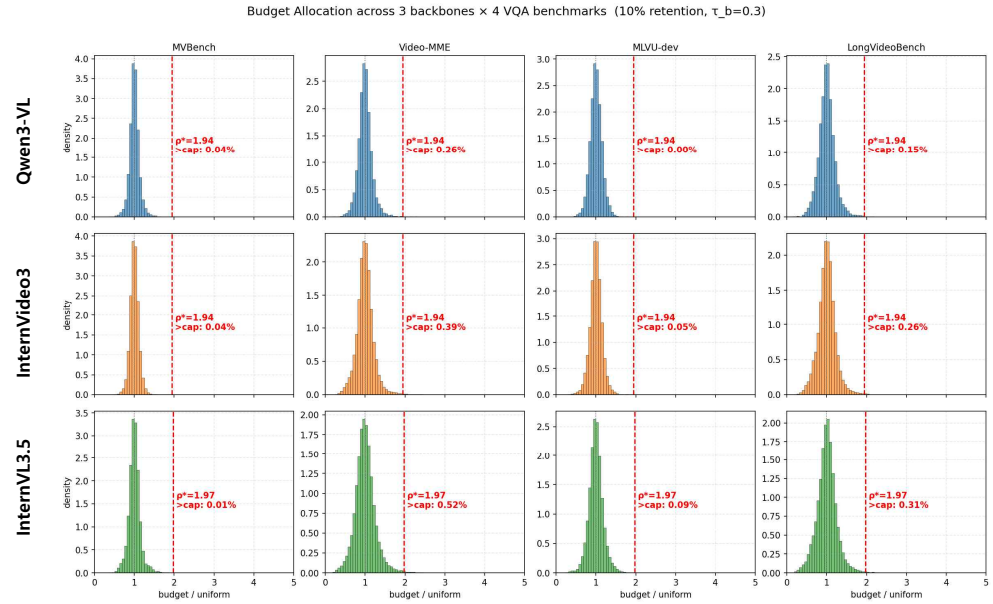

Figure A3: Pooled distribution of pair-level budget ratios $\rho_t = B_t / B_{\text{uniform}}$ across three Video-LLM backbones (rows) and four VQA benchmarks (columns) at r=0.10 and τ_b=0.3. Red dashed lines mark the overflow threshold ρ^*, with annotated overflow rates (>cap). All twelve combinations show overflow rates below 1%.

Table A3: Per-benchmark core component ablation on Qwen3-VL-8B at 10% token retention. ✓ denotes OT matching, importance-aware mass, and difficulty-based allocation, while × denotes Local Gibbs matching, uniform mass, and uniform allocation, respectively. VQA scores are accuracy and VTG scores are mIoU. Avg. (%) denotes mean performance retention relative to the uncompressed baseline.

Row	Components			Video Question Answering					Video Temporal Grounding		
	OT Match	Mass	Budget	MVBench	VideoMME	LVB	MLVU	Avg. (%)	Charades	ANet	Avg. (%)
(a)	×	✓	×	59.6	60.1	56.9	60.7	93.4	41.5	38.3	85.3
(b)	×	✓	✓	59.7	**60.8**	57.3	60.9	93.9	41.8	38.8	86.1
(c)	✓	×	×	58.9	59.8	57.8	60.1	93.1	41.6	38.8	85.9
(d)	✓	×	✓	59.3	60.0	58.1	60.7	93.7	42.0	39.0	86.6
(e)	✓	✓	×	**60.0**	60.0	58.8	60.7	94.3	42.6	39.1	87.3
(f)	✓	✓	✓	59.9	60.4	**59.5**	**61.1**	**94.8**	**43.0**	**39.5**	**88.2**

C.4 Budget Allocation Across Time

Figure A4 visualizes the per-pair transport difficulty W_t (red line) and the allocated budget B_t (blue bars) on two videos with contrasting temporal characteristics. In the dynamic video (top), where frame content changes substantially across pairs, W_t varies accordingly and the budget is allocated non-uniformly, concentrating on pairs with smaller content changes. In the static video (bottom), where consecutive frames depict nearly identical scenes, W_t remains uniformly low and the budget distribution approaches the uniform allocation. The adaptive mechanism thus produces non-uniform budgets when warranted by the video content and near-uniform budgets otherwise.

D Additional Ablation Study

Per-benchmark results of core component ablation. Table A3 provides the per-benchmark breakdown of the core component ablation summarized in Table 3. The results show that mass

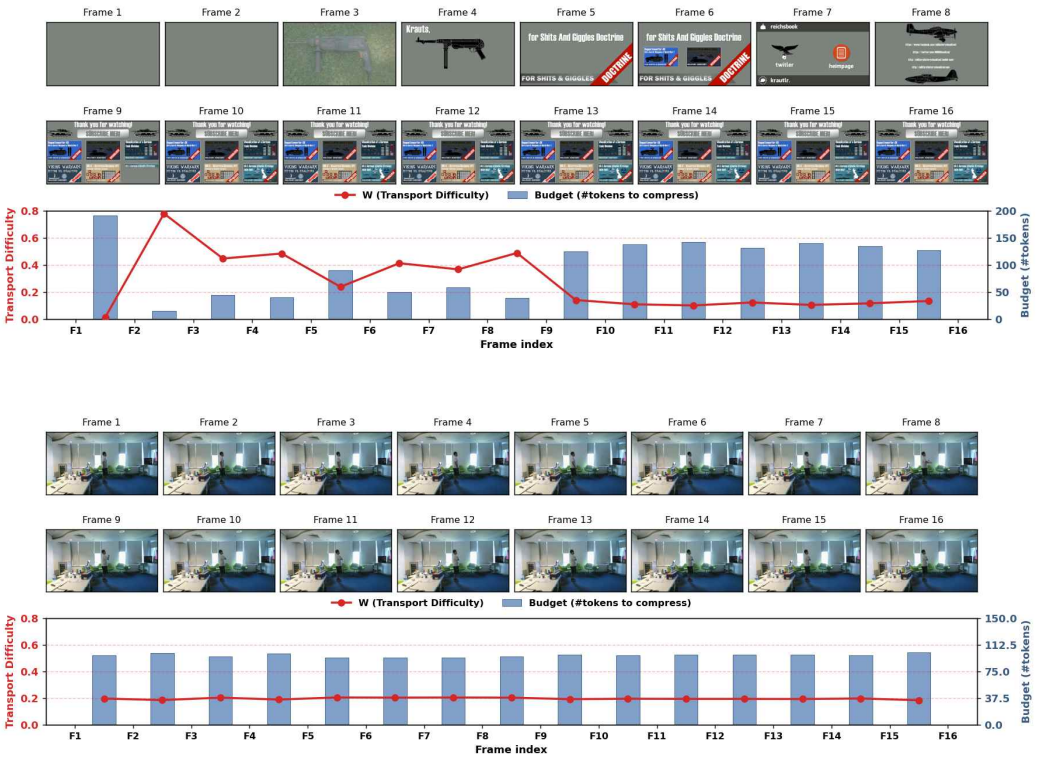

Figure A4: Per-pair transport difficulty W_t (red) and allocated budget B_t (blue) on 10% retention at Qwen3-VL with T=16. **Top:** dynamic video. **Bottom:** static video.

Table A4: Effect of the budget temperature τ_b on Qwen3-VL-8B at 10% retention. VQA scores are reported as accuracy and VTG scores as mIoU. Avg. (%) denotes the mean performance retention relative to the uncompressed baseline.

τ_b	Ret.	Video Question Answering					Video Temporal Grounding		
		MVBench	VideoMME	LVB	MLVU	Avg. (%)	Charades	ANet	Avg. (%)
0.1	10%	**60.2**	60.3	59.3	60.8	94.7	42.3	39.0	87.0
0.3	10%	59.9	**60.4**	59.5	**61.1**	**94.8**	**43.0**	**39.5**	**88.2**
1.0	10%	59.9	59.7	**59.6**	60.9	94.5	42.8	39.4	87.9
∞	10%	60.0	60.0	58.8	60.7	94.3	42.6	39.1	87.3

and budget contributions vary across benchmarks, consistent with the analyses of τ_m and τ_b in the following subsections.

D.1 Budget Temperature τ_b

The budget temperature τ_b controls the sharpness of the softmax distribution over transport difficulties $\{W_t\}$ in Eq. (8). As $\tau_b \to 0$, the allocation concentrates on the pair with the smallest W_t, amplifying the adaptive behavior but pushing the budget ratio ρ_t closer to the overflow threshold ρ^*. As $\tau_b \to \infty$, the allocation converges to the uniform baseline $B_t = B_{\text{uniform}}$, eliminating the data-dependent prioritization. Intermediate values balance these two regimes by modulating how strongly the relative differences in $\{W_t\}$ influence the budget distribution.

Table A4 evaluates budget temperature on Qwen3-VL-8B at 10% retention. Difficulty-based allocation improves VQA retention at all tested finite temperatures, while VTG benefits at $\tau_b = 0.3$ and 1.0 but declines slightly at $\tau_b = 0.1$. The default $\tau_b = 0.3$ achieves the highest VQA and VTG retention of 94.8% and 88.2%, improving over uniform allocation by 0.5 and 0.9 percentage points,

Table A5: Effect of the mass temperature τ_m on Qwen3-VL-8B at 10% retention. VQA scores are reported as accuracy and VTG scores as mIoU. Avg. (%) denotes the mean performance retention relative to the uncompressed baseline.

τ_m	Ret.	Video Question Answering					Video Temporal Grounding		
		MVBench	VideoMME	LVB	MLVU	Avg. (%)	Charades	ANet	Avg. (%)
0.1	10%	59.8	**61.1**	**59.8**	**61.2**	**95.2**	41.7	39.3	86.6
0.3	10%	59.9	60.4	59.5	61.1	94.8	**43.0**	**39.5**	**88.2**
1.0	10%	59.9	60.1	59.6	**61.2**	94.8	42.9	39.1	87.6
∞	10%	59.3	60.0	58.1	60.7	93.7	42.0	39.0	86.6

Table A6: Effect of the cost threshold τ_c on Qwen3-VL-8B at 10% retention. Disabled merges every selected match without a cost threshold. VQA scores are reported as accuracy and VTG scores as mIoU. Avg. (%) denotes the mean performance retention relative to the uncompressed baseline.

τ_c	Ret.	Video Question Answering					Video Temporal Grounding		
		MVBench	VideoMME	LVB	MLVU	Avg. (%)	Charades	ANet	Avg. (%)
0.1	10%	59.9	60.2	59.1	**61.5**	94.7	41.8	38.7	86.0
0.3	10%	59.9	60.4	**59.5**	61.1	**94.8**	**43.0**	**39.5**	**88.2**
0.5	10%	**60.3**	**60.6**	58.7	61.2	94.8	**43.0**	39.3	87.9
Disabled	10%	59.8	59.7	58.9	60.8	94.2	42.7	39.5	87.9

respectively. Both sharper ($\tau_b = 0.1$) and flatter ($\tau_b = 1.0$) allocations yield slightly lower retention, supporting an intermediate degree of budget adaptation.

D.2 Mass Temperature τ_m

The mass temperature τ_m controls the sharpness of the negative softmax in Eq. (3), which converts per-token contribution scores $\tilde{u}_{t,k}$ into the transport mass vector m_t. As $\tau_m \to 0$, the mass distribution concentrates on tokens with the smallest contribution scores, sharply prioritizing semantically representative tokens for preservation while making the transport plan strongly biased against compressing them. As $\tau_m \to \infty$, the mass distribution converges to a uniform vector $m_{t,k} = 1/K$, in which case the OT marginal constraint no longer encodes preservation priority and the transport reduces to similarity-driven matching. Intermediate values modulate how strongly intra-frame token importance influences the transport plan.

Table A5 evaluates mass temperature on Qwen3-VL-8B at 10% retention. Importance-aware mass improves VQA retention over uniform mass at all tested finite temperatures, while VTG gains are observed at $\tau_m = 0.3$ and 1.0. The default $\tau_m = 0.3$ raises VQA and VTG retention from 93.7% and 86.6% to 94.8% and 88.2%, respectively. A sharper distribution at $\tau_m = 0.1$ achieves the highest VQA retention of 95.2%, but lowers VTG retention to 86.6%, revealing a trade-off between the two tasks. We retain $\tau_m = 0.3$ as a shared setting that achieves the strongest VTG retention while maintaining competitive VQA performance.

D.3 Cost Threshold τ_c

The cost threshold τ_c determines whether each selected match in the transport plan is executed as a merge (uniform averaging) or as a prune (deletion of the source token), with matches satisfying $C_{t,ij} < \tau_c$ being merged and the remainder pruned. We introduced this dual-mode formulation following the well-established principle that merging is reliable when matched representations are close enough that uniform averaging preserves their information, while pruning is preferable when the matched representations differ substantially (Kim et al., 2024; Dhouib et al., 2025; Cao et al., 2023). Recent video token compression methods such as DyCoke (Tao et al., 2025) and PruneVid (Huang et al., 2025) adopt similar dual-mode strategies, and we incorporate τ_c as a principled safeguard against unreliable cross-frame correspondences.

Table A7: Effect of the mixing coefficient on Qwen3-VL-8B at 10% retention. Fixed α=0, 0.5, and 1 use position-only, equally weighted, and feature-only costs, respectively. Adaptive uses our frame-pair-dependent coefficient. VQA scores are accuracy and VTG scores are mIoU. Avg. (%) denotes mean performance retention relative to the uncompressed baseline.

Variant	Ret.	Video Question Answering					Video Temporal Grounding		
		MVBench	VideoMME	LVB	MLVU	Avg. (%)	Charades	ANet	Avg. (%)
α=0	10%	58.9	58.7	57.3	59.2	92.2	41.3	37.8	84.5
α=0.5	10%	**60.2**	60.6	58.6	60.6	94.5	42.9	**39.5**	88.0
α=1	10%	59.3	**61.2**	58.8	60.5	94.4	42.4	39.0	87.0
Adaptive (Ours)	10%	59.9	60.4	**59.5**	**61.1**	**94.8**	**43.0**	**39.5**	**88.2**

Table A6 evaluates the merge-cost threshold on Qwen3-VL-8B at 10% retention. The default $\tau_c = 0.3$ achieves 94.8% VQA and 88.2% VTG retention. A stricter threshold of 0.1 reduces VTG retention to 86.0%, suggesting that pruning too many selected matches can discard information that merging would retain. Increasing the threshold to 0.5 produces similar performance, while disabling it yields 94.2% VQA and 87.9% VTG retention. These results favor retaining a cost-based distinction between merging and pruning, although the gains over unrestricted merging are modest.

D.4 Mixing Weight α_t

We compare four options of the mixing weight α_t in Eq. (5): three fixed values (α=$0, 0.5, 1$), and our adaptive mixing coefficient. Table A7 compares transport-cost formulations on Qwen3-VL-8B at 10% retention. Position-only matching ($\alpha = 0$) yields the lowest VQA and VTG retention of 92.2% and 84.5%, highlighting the importance of feature similarity. Combining feature and position costs with $\alpha = 0.5$ improves upon feature-only matching ($\alpha = 1$), particularly on VTG, where retention increases from 87.0% to 88.0%. Adaptive mixing achieves the highest task averages of 94.8% and 88.2%, providing a modest additional gain over equal weighting. The results support combining semantic and spatial cues, with frame-pair-dependent weighting offering further refinement.

E Additional Architecture Comparisons

E.1 Outer-only and Hybrid Comparisons

Evaluation setup. We compare OTT-Vid with HoliTom (Shao et al., 2026) and FlashVID (Fan et al., 2026) on Qwen3-VL-8B and Qwen2.5-VL-7B using two protocols (Table A8). The outer-only comparison retains 10% of visual tokens before the first decoder layer and disables inner pruning for all methods. HoliTom and FlashVID use the released outer implementations with their retention settings adjusted to this common budget. The hybrid comparison replaces only the outer compressor with OTT-Vid while holding the outer token budget and the inner pruning layer, keep ratio, attention rule, and integer-budget rule fixed within each pair. Avg. reports the arithmetic mean of benchmark-level performance retention relative to the corresponding uncompressed backbone, using four benchmarks for VQA and two for VTG. Qwen2.5-VL VTG retention is computed using the reported vanilla mIoU scores of 36.7 and 30.7 for Charades-STA and ActivityNet-TimeLens, respectively.

Inner pruning settings. Both inner selectors rank visual tokens using decoder attention from the last query position, averaged over attention heads. For HoliTom, we retain 10% of tokens at the outer stage and keep 50% of the remaining tokens at decoder layer 18 on both backbones, giving approximately 5% final retention. This layer-18 rule is transferred from HoliTom's released LLaVA recipe rather than a native Qwen configuration. For FlashVID, we follow the released backbone-specific schedules. The outer stage retains 12.5% of tokens (a base rate of 10% with a 1.25 expansion factor), followed by 30% inner retention at layer 20 on Qwen2.5-VL and 10% at layer 28 on Qwen3-VL. The nominal final rates are therefore 3.75% and 1.25%, respectively. On Qwen3-VL, the HoliTom and FlashVID budgets use Python rounding and ceiling, respectively, identically within each comparison pair. Earlier decoder layers still process the outer-stage token count, so the two

hybrid families are evaluated as separate paired comparisons rather than at matched final retention or computational cost.

Table A8: Outer-only and hybrid comparisons on Qwen3-VL-8B and Qwen2.5-VL-7B. Method denotes the outer compressor, and Outer Ret. is its output token retention. The block heading specifies the fixed inner stage. Individual benchmark columns report accuracy (VQA) or mIoU (VTG), while shaded Avg. columns report mean performance retention (%). Best results within each block and backbone are in **bold**, determined before rounding.

		Qwen3-VL-8B								**Qwen2.5-VL-7B**							
		VQA					VTG			VQA					VTG		
Method	**Outer Ret.**	MVB	VMME	LVB	MLVU	**Avg.**	Cha.	ANet	**Avg.**	MVB	VMME	LVB	MLVU	**Avg.**	Cha.	ANet	**Avg.**
Vanilla	100%	65.8	63.7	60.6	64.2	100.0	47.4	46.1	100.0	69.0	61.4	58.7	62.8	100.0	36.7	30.7	100.0
Outer-only compression																	
HoliTom	10%	**61.4**	60.3	58.2	60.2	94.5	42.7	38.6	87.0	**64.1**	57.6	55.8	60.4	94.5	29.6	14.7	64.2
FlashVID	10%	60.2	**60.6**	59.2	60.3	94.6	42.3	39.1	87.0	63.7	55.5	**56.9**	59.4	93.6	28.0	9.7	54.0
OTT-Vid	10%	59.9	60.4	**59.5**	**61.1**	**94.8**	**43.0**	**39.5**	**88.2**	64.1	**58.3**	56.8	**60.9**	**95.4**	**32.2**	**18.4**	**73.9**
HoliTom inner pruning																	
HoliTom	10%	**61.6**	**60.6**	57.5	60.2	**94.3**	**42.7**	38.5	86.8	63.9	57.6	55.6	60.2	94.2	29.5	14.6	64.0
OTT-Vid	10%	59.4	60.4	**58.3**	**61.0**	94.1	42.6	**39.1**	**87.3**	**63.9**	**58.2**	**56.5**	**60.8**	**95.1**	**32.3**	**18.2**	**73.7**
FlashVID inner pruning																	
FlashVID	12.5%	60.6	60.5	**59.7**	62.0	95.6	43.5	40.8	90.1	64.1	59.6	56.8	60.3	95.7	30.8	19.1	73.0
OTT-Vid	12.5%	**61.3**	**61.3**	58.8	**62.6**	**96.0**	**43.6**	**40.9**	**90.4**	**64.7**	**59.6**	**57.1**	**61.2**	**96.4**	**33.6**	**20.1**	**78.4**

Results. At matched 10% outer retention, OTT-Vid achieves the highest average VQA and VTG retention on both backbones, demonstrating its effectiveness independently of inner-LLM pruning. Its strong performance also persists when the inner stage is retained and only the outer compressor is replaced. Under identical inner-pruning configurations, OTT-Vid improves VTG retention in all four paired comparisons and VQA retention in three, with a small 0.2-point VQA decrease for HoliTom on Qwen3-VL. On Qwen2.5-VL, VTG gains reach 9.7 and 5.4 percentage points with HoliTom and FlashVID inner pruning, respectively. These results show that OTT-Vid's benefits extend to hybrid compression pipelines and are not restricted to comparisons that disable inner pruning.

E.2 Comparison with AOT

Supported LLaVA backbones. We additionally compare with AOT (Li et al., 2026b), which also uses optimal transport for training-free video token reduction. Table A9 reports VQA results on LLaVA-OneVision-7B and LLaVA-Video-7B, the backbones supported by AOT's released implementation. These comparisons provide our primary evidence on AOT's supported architectures. OTT-Vid improves mean VQA performance retention from 96.8% to 97.7% on LLaVA-OneVision and from 94.1% to 94.5% on LLaVA-Video. The results are mixed across individual benchmarks on LLaVA-OneVision, while OTT-Vid scores higher on all four benchmarks on LLaVA-Video.

Table A9: VQA comparison with AOT on its supported LLaVA backbones on 10% retention. Individual columns report accuracy, and shaded Avg. columns report mean benchmark-level performance retention (%). Best results within each backbone are in **bold**.

	LLaVA-OneVision-7B					**LLaVA-Video-7B**				
Method	MVB	VMME	LVB	MLVU	**Avg.**	MVB	VMME	LVB	MLVU	**Avg.**
AOT	**57.1**	56.2	54.2	**61.6**	96.8	58.9	61.4	55.6	63.8	94.1
OTT-Vid	56.6	**57.1**	**56.1**	61.5	**97.7**	**59.2**	**61.5**	**55.8**	**64.4**	**94.5**

Architecture-aware adaptation to Qwen2.5-VL. Since AOT does not provide an official Qwen2.5-VL layer configuration, we additionally evaluate an architecture-aware port. AOT's released SigLIP configuration extracts shallow and deep attention features from blocks 6 and 25 of a 26-block vision encoder, at approximately 23% and 96% of its depth. Qwen2.5-VL instead has 32 vision blocks,

with full attention at the 8th, 16th, 24th, and 32nd blocks and window attention elsewhere. We select the 8th and 32nd blocks as the shallow and deep layers, respectively, preserving a similar relative depth while using global attention at both extraction points. This pair was selected before evaluation, fixed across all six benchmarks, and not tuned to individual benchmark results. Both methods use a nominal visual-token retention ratio of 10%.

Table A10: Supplementary comparison with our architecture-aware AOT port on Qwen2.5-VL-7B at 10% token retention. VQA scores are accuracy and VTG scores are mIoU. Shaded Avg. columns report mean performance retention (%). This port uses a fixed vision-layer configuration rather than an official AOT Qwen configuration.

	VQA					VTG		
Method	MVB	VMME	LVB	MLVU	**Avg.**	Cha.	ANet	**Avg.**
AOT (ported)	63.3	55.7	55.0	58.3	92.2	24.1	7.9	45.7
OTT-Vid	**64.1**	**58.3**	**56.8**	**60.9**	**95.4**	**32.2**	**18.4**	**73.9**

Scope of the comparison. OTT-Vid outperforms this AOT port on all six benchmarks, with a larger gap in temporal grounding (Table A10). This is a supplementary robustness check under the specified port, while the supported LLaVA comparisons remain the primary comparison with AOT. The results support the effectiveness of OTT-Vid across these settings.

F Additional Qualitative Results

We provide additional qualitative result on MVBench (Figure A5) and ActivityNet-TimeLens (Figure A6) at 10% retention. In both figures, bright patches indicate retained tokens and red circles mark the query-relevant regions.

Figure A5 shows a counting query asking how many purple objects exit the scene. The purple cylinder on the right gradually exits as the camera moves, providing key evidence for the answer. OTT-Vid retains tokens around the cylinder across the displayed frames as it leaves the scene and produces the correct count. This example illustrates how compression can retain visual evidence needed to track an object's presence and exit over time.

Figure A6 shows a temporal grounding query for cutting meat, with a ground-truth interval of 51.4–58.2 s. OTT-Vid retains tokens around the hand and knife across the displayed frames, including the middle of the action at 56 s. Its predicted interval of 51–58 s closely matches the ground truth, illustrating the retention of action-related representations throughout an event rather than only at isolated moments.

Effect of importance-aware mass. Figure A7 compares uniform and importance-aware mass at 10% token retention on an MVBench video of a person putting down a pen. With uniform mass, few retained tokens cover the hand and pen, and the model responds "Not Sure." Importance-aware mass retains more tokens around these action-relevant regions across the displayed frames, and the model correctly predicts "Putting." This example illustrates how intra-frame importance can guide temporal compression toward retaining evidence needed for action recognition, complementing the quantitative mass ablation.

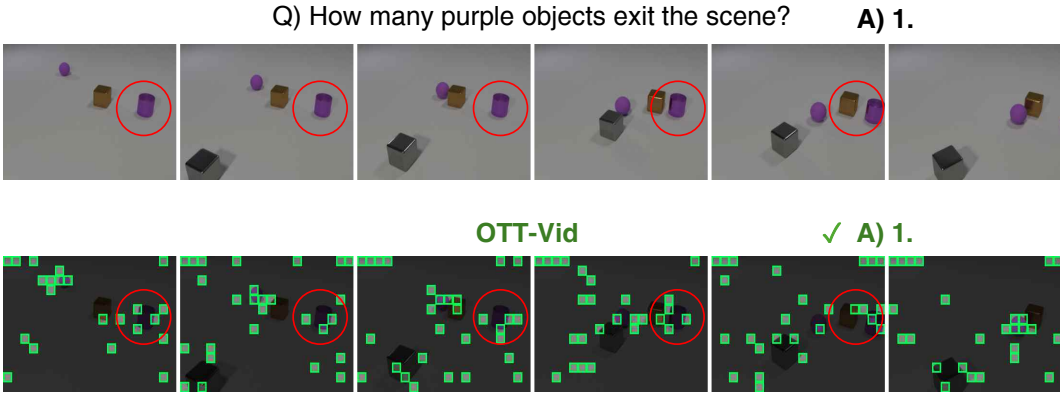

Figure A5: Qualitative comparison on MVBench at 10% retention. Bright patches indicate retained tokens. Red circles highlight the question-relevant region.

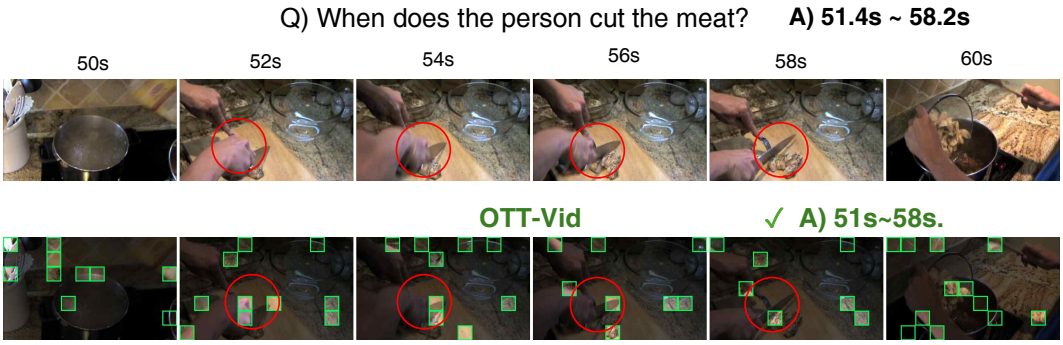

Figure A6: Qualitative comparison on ActivityNet-TimeLens at 10% retention. Bright patches indicate retained tokens. Red circles highlight the question-relevant region.

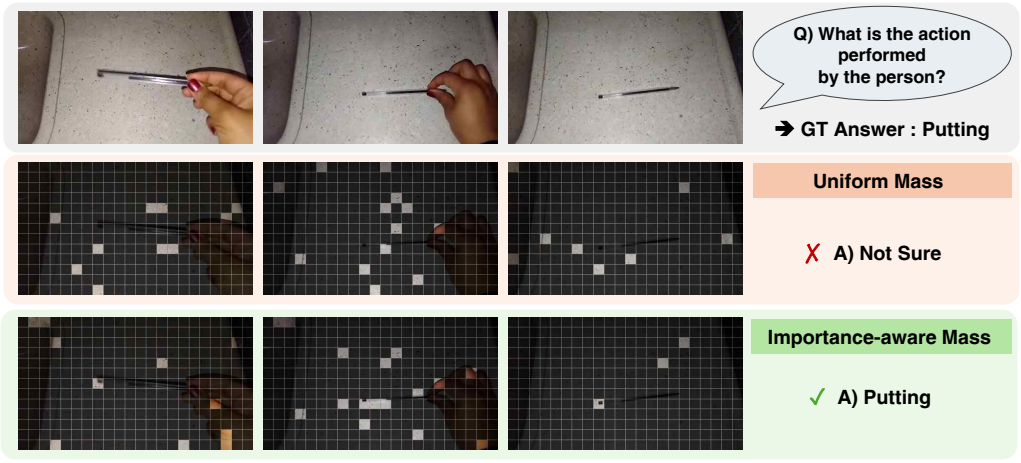

Figure A7: Effect of uniform versus importance-aware mass on retained tokens at 10% retention on MVBench. Bright patches indicate retained tokens. Importance-aware mass retains more visual evidence around the hand and pen in this example, accompanying the correct action prediction.