SEMANTICALLY ALIGNED KV CACHE COMPRESSION FOR EFFICIENT AGENTIC AUTOREGRESSIVE VIDEO GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Agentic autoregressive video generation offers a promising approach to controllable video synthesis through semantic planning, sequential generation, and visual verification. However, growing visual histories enlarge the key–value (KV) cache, increasing memory use and attention costs across candidate rollouts. KV cache compression reduces this overhead, but existing methods primarily optimize individual generators without explicitly accounting for the planner's semantic constraints. Applying them directly can assign low importance to states associated with particular requested attributes or actions, potentially weakening semantic fidelity. In this paper, we introduce SEMANTICKV, a training-free method for semantically aligned KV cache compression. To our knowledge, SEMANTICKV is the first KV cache compression method for agentic autoregressive video generation that explicitly uses planner-extracted semantic constraints to guide retention. Text cross-attention identifies constraint-related video positions, whose self-attention queries are collected into separate groups. Attention weights are averaged within each group and maximized across groups, giving each cached entry a high score when it is strongly used by any represented constraint. A single top-k selection retains high-scoring entries under the memory budget, leaving their keys, values, and positions unchanged. For fixed query groups and equal entry costs, this selection minimizes an additive upper bound on worst-group discarded attention. The method operates independently within each candidate without retraining or additional video generations. Our evaluation spans four AR video backbones and three benchmarks, covering 5-, 30-, and 60-second generation; our method achieves strong performance across these settings. On Causal Forcing at a compression ratio of 0.7, SEMANTICKV obtains a VBench Total Score of 84.10, compared with 84.05 for the strongest compressed baseline and 83.99 for Full KV. Meanwhile, it achieves a speedup of $1.22\times$ over Full KV and reduces KV-cache memory from 6.04 GB to 1.81 GB, corresponding to a $3.35\times$ reduction.

1 INTRODUCTION

Agentic autoregressive (AR) video generation offers a promising approach to producing controllable videos from complex user requests (Zeng et al., 2025; Feng et al., 2026; Huang et al., 2026b). A semantic planner clarifies the requested content and refines the prompt (Yang et al., 2026d; Cudlenco et al., 2026; Huang et al., 2025b), an AR generator synthesizes frames or latent chunks sequentially from previously generated content (Henschel et al., 2025; Gao et al., 2024), and a visual critic assesses generated candidates against the request (Long et al., 2026). These complementary roles address the need to both specify intended content and check whether the resulting video follows it. This workflow, however, incurs substantial inference overhead. Each rollout stores historical keys and values in a KV cache to reuse information from earlier chunks (Xing et al., 2024; Gao et al., 2025). Longer visual histories increase cache storage and attention computation, while generating multiple candidates repeats these costs even when their caches are not resident simultaneously.

KV cache compression alleviates this bottleneck by reducing the historical states stored and accessed during attention. Representative methods include **R-KV** (Cai et al., 2025), which selects cached states using importance and redundancy; **Dummy Forcing** (Guo et al., 2026), which allo-

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.

cates context according to attention-head locality; and **Forcing-KV** (Ji et al., 2026), which applies different retention policies to static and dynamic heads. Other approaches exploit temporal redundancy (Chen et al., 2026a), future-query statistics (Luo et al., 2026), historical coverage (Zhao et al., 2026), or instance-specific context representations (Fu et al., 2026). These methods primarily optimize *context use within an individual generator*, rather than explicitly connecting cache retention to the semantic constraints produced by an agentic planner.

Directly applying such policies within an agentic workflow leaves two closely related limitations. First, the planner distinguishes user-specified subjects, attributes, relations, and actions, but cache selection need not identify which states are used by computations associated with each constraint. A detailed plan can therefore guide prompt construction without guiding the history retained during generation. Second, when attention observations are pooled into a global importance score, a state strongly used for one attribute or action can be assigned a low average score because unrelated queries give it little attention. Discarding such states may weaken the information supporting the requested content. The resulting deviations can propagate through subsequent chunks (Liu et al., 2026; Guo et al., 2025) and compromise long-horizon consistency (Xie et al., 2025; Wang et al., 2025). The missing capability is thus not merely stronger compression, but *semantically aligned retention that carries the planner's explicit constraints into cache selection*.

In this paper, we introduce SEMANTICKV, a training-free method for semantically aligned KV cache compression in agentic autoregressive video generation. To our knowledge, SEMANTICKV is the first KV cache compression method in this setting to explicitly use planner-extracted semantic constraints to guide retention. Its central idea is that *the semantics that guide video generation should also guide the history its generator retains*. The existing semantic constraints $\mathcal{R}$ determine which generator queries are observed and how their cache-attention weights are grouped. This connects retention to the requested content while preventing low scores from unrelated groups from diluting an entry's importance to a particular constraint. The planner, generator, and visual-verification workflow remain unchanged.

Specifically, SEMANTICKV first matches each semantic constraint to its words in the candidate prompt. Text cross-attention identifies video positions that attend strongly to these words, and their actual self-attention queries form a constraint-specific group. For each KV entry, the method averages attention weights within each group and takes the maximum across groups to obtain one semantic importance score. Thus, an entry can receive high priority even when it is primarily used by only one represented constraint. At each completed-chunk cache update, a single top-k selection retains the highest-scoring eligible entries within the byte budget, alongside states required by the backbone. Retained keys, values, and positions are packed unchanged, and each candidate uses its own observations and cache. Our local analysis relates discarded attention to groupwise attention-output changes and shows that, for fixed query groups and equal entry costs, top-k selection minimizes an additive upper bound on worst-group discarded attention (Appendix D). This provides a theoretical basis for the scoring rule without adding another optimization stage.

To evaluate SEMANTICKV, we conduct experiments on four AR video backbones: LongLive (Yang et al., 2026c), LongLive 2.0 (Chen et al., 2026b), Self Forcing (Huang et al., 2025a), and Causal Forcing (Zhu et al., 2026). We use three benchmarks—VBench (Huang et al., 2024), VBench-Long (Huang et al., 2025c), and VisionReward (Xu et al., 2026)—covering short-video quality, long-horizon consistency, and preference-based assessment at 5-, 30-, and 60-second durations. Comparisons include three KV cache compression baselines and Full KV under matched generation settings and realized cache budgets. Two targeted ablations isolate constraint-conditioned query selection and groupwise score aggregation, while efficiency analysis reports cache memory and generator throughput. On Causal Forcing at a compression ratio of 0.7, SEMANTICKV obtains a VBench Total Score of 84.10, compared with 84.05 for the strongest compressed baseline and 83.99 for Full KV. Meanwhile, it achieves a speedup of $1.22\times$ over Full KV and reduces KV-cache memory from 6.04 GB to 1.81 GB, corresponding to a $3.35\times$ reduction.

2 RELATED WORK

Agentic autoregressive video generation. Autoregressive video models generate causally ordered frames or latent chunks for streaming and long-horizon synthesis (Chen et al., 2024; 2025; ai et al., 2025; Deng et al., 2025; Wang et al., 2024b), with advances in rollout stability, efficiency,

and memory modeling (Liu et al., 2026; Guo et al., 2025; Lin et al., 2025; Yu et al., 2025). Agentic systems add planning, generation, and feedback (Yuan et al., 2024; Liang et al., 2025; Lin et al., 2024; Xie et al., 2024; Hu et al., 2024) through staged production, role-based collaboration, and hierarchical planning (Zhuang et al., 2024; Xu et al., 2025; Wu et al., 2025), as well as experience- or feedback-driven refinement (Wang et al., 2024a; Soni et al., 2024). Longer visual histories increase KV-cache costs within each rollout, and candidate generation repeats this overhead. SEMANTICKV keeps the planning–generation–verification workflow and AR backbone unchanged, using the planner's semantic constraints to guide cache retention within each candidate.

KV cache compression. KV cache compression reduces inference costs by selecting or compacting cached keys and values. Approaches include merging and salience-based retention (Samuel et al., 2026; Chen et al., 2026a), compact representations and structured selection (Xi et al., 2026; Ma et al., 2026; Mao et al., 2026; Wang et al., 2026; Luo et al., 2026), context management (Li et al., 2026a; Meng et al., 2026; Huang et al., 2026c; Zhang et al., 2026), and response-aware compression (Feng et al., 2025b; Gu et al., 2025; Zweiger et al., 2026; Li et al., 2026c; Yang et al., 2026b). Representative methods include R-KV (Cai et al., 2025), which selects states by importance and redundancy; Dummy Forcing (Guo et al., 2026), which limits context by head locality; and Forcing-KV (Ji et al., 2026), which preserves transition anchors for static heads and prunes similar segments for dynamic heads. These methods do not explicitly use planner-extracted semantic constraints, potentially overlooking task-relevant states. SEMANTICKV instead uses these constraints to identify and retain relevant KV states within the memory budget.

3 PROBLEM FORMULATION

3.1 AGENTIC AR VIDEO GENERATION

Autoregressive (AR) video diffusion models generate video chunks sequentially from a text prompt and previously generated content (Yin et al., 2025; Huang et al., 2025a; Yang et al., 2026c). Given prompt $\mathbf{q}$, the video $\mathbf{X}_{1:T} = (\mathbf{X}_1, \ldots, \mathbf{X}_T)$ is generated as

$$\mathbf{X}_t = \mathcal{G}_{\boldsymbol{\theta}} \left(\mathbf{K}_{t-1}, \mathbf{V}_{t-1}, \mathbf{q}; \boldsymbol{\rho}_t \right), \tag{1}$$

where $\mathcal{G}_{\boldsymbol{\theta}}$ is the pretrained generator, $(\mathbf{K}_{t-1}, \mathbf{V}_{t-1})$ is the historical KV cache, and $\boldsymbol{\rho}_t$ denotes generation randomness. Although this formulation supports streaming and long-horizon generation (Yin et al., 2025; Huang et al., 2025a; Yang et al., 2026c), early errors can propagate through the generated history (Liu et al., 2026; Li et al., 2026b). We therefore adopt a planning–generation–verification workflow (Huang et al., 2026a; Zeng et al., 2025):

$$\mathcal{A} = (\mathcal{P}_{\boldsymbol{\phi}}, \mathcal{G}_{\boldsymbol{\theta}}, \mathcal{C}_{\boldsymbol{\psi}}, \pi_{\delta,\tau}), \tag{2}$$

where $\mathcal{P}_{\boldsymbol{\phi}}$ is the semantic planner, $\mathcal{G}_{\boldsymbol{\theta}}$ generates candidate videos, $\mathcal{C}_{\boldsymbol{\psi}}$ is the vision–language critic, and $\pi_{\delta,\tau}$ performs conservative candidate selection.

3.2 CONSERVATIVE SEMANTIC PLANNING

The planner refines the input while preserving the original request (Cheng et al., 2025; Yang et al., 2026a):

$$(\mathcal{R}, \mathcal{N}, \mathbf{q}^1) = \mathcal{P}_{\boldsymbol{\phi}}(\mathbf{q}), \qquad \mathbf{q}^0 = \mathbf{q}, \tag{3}$$

where $\mathcal{R}$ contains semantic constraints such as subjects, attributes, relations, actions, and motion, and $\mathcal{N}$ contains contradictions or unrequested content to avoid. The refined prompt $\mathbf{q}^1$ organizes the information in $\mathcal{R}$ while respecting $\mathcal{N}$, whereas $\mathbf{q}^0$ keeps the original prompt unchanged. If planning or validation fails, we set $\mathbf{q}^1 = \mathbf{q}^0$. More details are provided in Appendix C.

3.3 DUAL-CANDIDATE AUTOREGRESSIVE GENERATION

Using $\mathbf{q}^0$ and $\mathbf{q}^1$, the same generator produces two candidates:

$$\mathcal{V}(\mathbf{q}) = \left\{ \mathbf{X}^0_{1:T}, \mathbf{X}^1_{1:T} \right\}, \qquad \mathbf{X}^i_t = \mathcal{G}_{\boldsymbol{\theta}} \left(\mathbf{K}^i_{t-1}, \mathbf{V}^i_{t-1}, \mathbf{q}^i; \boldsymbol{\rho}_{i,t} \right), \quad i \in \{0, 1\}, \tag{4}$$

where $i = 0$ and $i = 1$ denote the original-prompt and planned candidates, respectively. Each candidate maintains an independent KV cache and random-input sequence, and no cached states are shared between them. The original-prompt candidate serves as a fallback (Long et al., 2026; Liu et al., 2025).

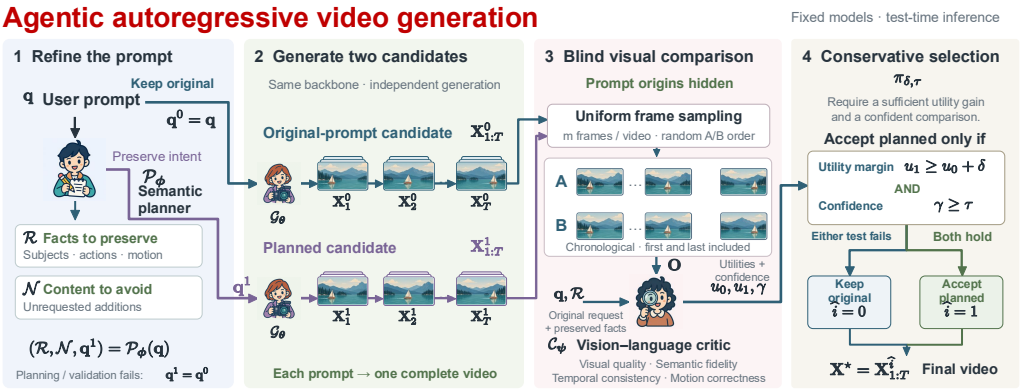


Figure 1: Agentic AR video generation. Semantic constraints $\mathcal{R}$ and prompt-specific exclusions $\mathcal{N}$ guide prompt refinement. The original-prompt and planned candidates from the same generator undergo randomized, anonymous visual verification, followed by conservative selection using Eq. 7.

3.4 Blind Pairwise Visual Verification

After generation, the critic compares both candidates against the original request $\mathbf{q}$ and semantic constraints $\mathcal{R}$ (Long et al., 2026). Their display order is randomized and anonymized to reduce order bias (Zheng et al., 2023). The visual input is

$$\mathbf{O} = \mathcal{U}_m\left(\mathbf{X}^{\kappa(0)}_{1:T}, \mathbf{X}^{\kappa(1)}_{1:T}\right), \tag{5}$$

where $\mathcal{U}_m$ samples m evenly spaced frames from each candidate and κ denotes the randomized display order.

The critic evaluates visual quality, semantic fidelity, temporal consistency, and motion correctness (He et al., 2024). The utility of candidate i is

$$u_i = w_{\mathrm{v}} z^i_{\mathrm{v}} + w_{\mathrm{s}} z^i_{\mathrm{s}} + w_{\mathrm{t}} z^i_{\mathrm{t}} + w_{\mathrm{m}} z^i_{\mathrm{m}}, \tag{6}$$

where the nonnegative weights sum to one. The critic also outputs confidence $\gamma \in [0, 1]$, which is used together with u_0 and u_1 for candidate selection.

3.5 Conservative Candidate Selection

The planned candidate is selected only when both the utility gain and confidence exceed their thresholds (Zheng et al., 2023):

$$\widehat{i} = \pi_{\delta,\tau}(u_0, u_1, \gamma) = \begin{cases} 1, & u_1 \geq u_0 + \delta \ \wedge \ \gamma \geq \tau, \\ 0, & \text{otherwise}, \end{cases} \tag{7}$$

where $\delta > 0$ is the minimum utility gain and τ is the confidence threshold. The final output is

$$\mathbf{X}^\star = \mathbf{X}^{\widehat{i}}_{1:T}. \tag{8}$$

Figure 1 summarizes the complete planning–generation–verification workflow.

3.6 KV Cache Compression Problem

We compress each candidate's KV cache during AR generation (Ji et al., 2026; Guo et al., 2026) without changing planning, verification, or selection. Omitting candidate indices, the compression policy is

$$(\widetilde{\mathbf{K}}_t, \widetilde{\mathbf{V}}_t) = \mathcal{C}_\eta(\mathbf{K}_t, \mathbf{V}_t; \mathcal{R}, \mathbf{q}^i), \tag{9}$$

where η denotes the policy settings. The next chunk is generated using the compressed history:

$$\widetilde{\mathbf{X}}^i_{t+1} = \mathcal{G}_{\boldsymbol{\theta}}\left(\widetilde{\mathbf{K}}_t, \widetilde{\mathbf{V}}_t, \mathbf{q}^i; \boldsymbol{\rho}_{i,t+1}\right). \tag{10}$$

Let $(\mathbf{K}_t^{\text{full}}, \mathbf{V}_t^{\text{full}})$ denote the uncompressed reference under the same native schedule. We aim to maximize video quality under a bounded cache budget:

$$\max_{\eta} \mathbb{E}\Big[\mathcal{Q}\Big(\widetilde{\mathbf{X}}_{1:T}, \mathbf{q}\Big)\Big] \quad \text{s.t.} \quad \max_{t} \frac{M(\widetilde{\mathbf{K}}_t, \widetilde{\mathbf{V}}_t)}{M(\mathbf{K}_t^{\text{full}}, \mathbf{V}_t^{\text{full}})} \le 1 - r, \quad 0 \le r < 1, \tag{11}$$

where $\mathcal{Q}$ measures video quality, r is the compression ratio (the fraction of cache removed), and M measures KV cache storage.

4 METHODOLOGY

SEMANTICKV implements the compression policy $\mathcal{C}_\eta$ in Eq. 9 by retaining cached information used by queries associated with the planner's semantic constraints $\mathcal{R}$. We first identify these queries, then aggregate their attention into one score per KV entry and apply a single top-k selection under the memory constraint in Eq. 11.

4.1 CONSTRAINT-CONDITIONED QUERY SELECTION

Connecting semantic constraints to video positions. For each layer and KV head, let $\mathcal{H}_t$ index the available entries in $(\mathbf{K}_t, \mathbf{V}_t)$ after chunk t of candidate i, with $\mathcal{A}_t \subseteq \mathcal{H}_t$ containing entries required by the backbone. We match the content words of each semantic constraint $r_a \in \mathcal{R}$ to positions $\mathcal{T}_a \subseteq \mathcal{T}_{\text{text}}$ in the candidate prompt $\mathbf{q}^i$, where $\mathcal{T}_{\text{text}}$ contains the text positions visible under the native cross-attention mask.

For video-token position x in the completed chunk and text position $b \in \mathcal{T}_{\text{text}}$, let $\mathbf{z}_{x,h}^{\text{cross}}$ and $\mathbf{k}_{b,h}^{\text{text}}$ denote the cross-attention query and text key in routing head h, including native normalization. With H_c routing heads and query/key dimension d_c, we compute

$$\alpha_{xb}^{\text{text}} = \frac{1}{H_c} \sum_{h=1}^{H_c} \frac{\exp\Big((\mathbf{z}_{x,h}^{\text{cross}})^\top \mathbf{k}_{b,h}^{\text{text}} / \sqrt{d_c}\Big)}{\sum_{b' \in \mathcal{T}_{\text{text}}} \exp\Big((\mathbf{z}_{x,h}^{\text{cross}})^\top \mathbf{k}_{b',h}^{\text{text}} / \sqrt{d_c}\Big)}, \qquad g_a(x) = \sum_{b \in \mathcal{T}_a} \alpha_{xb}^{\text{text}}. \tag{12}$$

Here, $\alpha_{xb}^{\text{text}}$ is the head-averaged attention from video position x to text token b. A larger routing score $g_a(x)$ indicates stronger attention to the words matched to constraint r_a.

Collecting the corresponding queries. For each reliably matched constraint, we select up to n_q positions with the largest $g_a(x)$, where n_q is the per-group query limit. Let $\mathbf{z}_p$ denote an existing self-attention query collected at a selected position, with p indexing the query observation, and let $\mathcal{P}_a$ contain the query indices for constraint r_a. Unlike $\mathbf{z}_{x,h}^{\text{cross}}$, which attends to text keys, $\mathbf{z}_p$ attends to video keys in $\mathbf{K}_t$. Thus, cross-attention selects where to collect queries, while self-attention measures which cache entries those positions use.

4.2 SEMANTICALLY ALIGNED KV RETENTION

From selected queries to cache-attention weights. Let $\mathcal{I} \neq \varnothing$ index the nonempty query groups used for scoring. For entry $j \in \mathcal{H}_t$, let $\mathbf{k}_j$ and $\mathbf{v}_j$ denote its key and value, and let $\mathcal{H}_{t,p} \subseteq \mathcal{H}_t$ contain the entries visible to query $\mathbf{z}_p$ under the native attention mask. With query/key dimension d_k, we compute

$$\alpha_{pj} = \frac{\exp(\mathbf{z}_p^\top \mathbf{k}_j / \sqrt{d_k})}{\sum_{j' \in \mathcal{H}_{t,p}} \exp(\mathbf{z}_p^\top \mathbf{k}_{j'} / \sqrt{d_k})}, \quad j \in \mathcal{H}_{t,p}, \qquad \alpha_{pj} = 0 \text{ otherwise.} \tag{13}$$

Unlike $\alpha_{xb}^{\text{text}}$ in Eq. 12, these weights describe attention to cached video entries rather than prompt tokens. We average them within each group and take the maximum across groups:

$$w_{aj} = \frac{1}{|\mathcal{P}_a|} \sum_{p \in \mathcal{P}_a} \alpha_{pj}, \qquad s_j = \max_{a \in \mathcal{I}} w_{aj}. \tag{14}$$

Here, w_{aj} is group a's mean attention to entry j, and s_j is its semantic importance. The mean avoids favoring a group merely because it contains more queries; the maximum gives an entry high priority when strongly used by any represented constraint.

Why discarded attention matters. For a retained set $\mathcal{A}_t \subseteq \mathcal{S} \subseteq \mathcal{H}_t$, define the average attention discarded from group a and the largest such loss as

$$\ell_a(\mathcal{S}) = \sum_{j \in \mathcal{H}_t \setminus \mathcal{S}} w_{aj}, \qquad \mathcal{L}(\mathcal{S}) = \max_{a \in \mathcal{I}} \ell_a(\mathcal{S}). \tag{15}$$

We compare attention before and after retention with the observed queries, retained keys, values, positions, and native masks held fixed. Let $\mathbf{o}_p$ and $\mathbf{o}_p^{\mathcal{S}}$ denote the corresponding value-weighted outputs before the output projection; retained attention is renormalized over the remaining visible entries.

Theorem 4.1 (Discarded attention controls local output error) *Suppose $\mathcal{S}$ retains at least one visible entry for every selected query. Let $D_{\mathrm{V}} = \max_{j,j' \in \mathcal{H}_t} \|\mathbf{v}_j - \mathbf{v}_{j'}\|_2$ be the largest distance between available value vectors. Then the largest group-average attention-output error satisfies*

$$\mathcal{E}(\mathcal{S}; \mathbf{V}_t) := \max_{a \in \mathcal{I}} \frac{1}{|\mathcal{P}_a|} \sum_{p \in \mathcal{P}_a} \|\mathbf{o}_p - \mathbf{o}_p^{\mathcal{S}}\|_2 \le D_{\mathrm{V}}\, \mathcal{L}(\mathcal{S}). \tag{16}$$

Theorem 4.1 connects attention retention to a local output guarantee: limiting the discarded mass in every observed group limits its average output change. The result includes attention renormalization and does not require the discarded mass to be small. Its worst-case sharpness is established in Appendix D.1.

Why use the groupwise maximum. Directly minimizing $\mathcal{L}$ couples the retention decisions because its worst-affected group can change with the retained set. The score s_j instead gives each entry a fixed contribution to an upper bound.

Theorem 4.2 (Smallest universal additive bound) *For every $\mathcal{A}_t \subseteq \mathcal{S} \subseteq \mathcal{H}_t$,*

$$\mathcal{L}(\mathcal{S}) \le \mathcal{U}(\mathcal{S}) := \sum_{j \in \mathcal{H}_t \setminus \mathcal{S}} s_j \le |\mathcal{I}|\, \mathcal{L}(\mathcal{S}). \tag{17}$$

Among bounds of the form $\sum_{j \in \mathcal{H}_t \setminus \mathcal{S}} b_j$ with fixed nonnegative coefficients that hold for every deletion of optional entries, the smallest valid coefficients are $b_j = s_j$ for $j \in \mathcal{H}_t \setminus \mathcal{A}_t$.

Thus, the maximum is not an arbitrary aggregation: it gives the smallest entrywise penalties that upper-bound every group's discarded attention for all deletion sets. In contrast, pooling queries controls a query-count-weighted average and can dilute an entry's importance to a particular constraint (Appendix D.3). This does not imply that every constraint receives reserved memory or is satisfied in the generated video.

From the memory constraint to retained entries. We allocate the total byte budget implied by compression ratio r in Eq. 11 across layer/KV-head units in proportion to their uncompressed reference sizes. Let B_t denote the current unit's budget. Mandatory entries $\mathcal{A}_t$ count toward B_t; SEMANTICKV adds no further protected states. Since entries within a unit have equal native-precision KV costs, let k_t be the largest number of additional available entries that fit after accounting for mandatory states, position records, metadata, and allocation granularity. We retain

$$\mathcal{S}_t = \mathcal{A}_t \cup \mathrm{TopK}_{j \in \mathcal{H}_t \setminus \mathcal{A}_t}(s_j, k_t), \qquad M(\mathbf{K}_{\mathcal{S}_t}, \mathbf{V}_{\mathcal{S}_t}) \le B_t. \tag{18}$$

Here, TopK returns the k_t highest-scoring eligible indices. The selected keys and values form the compressed cache in Eq. 9; M includes persistent position records and metadata. Budgets that cannot hold the mandatory states and a usable attention context are reported as infeasible.

Theorem 4.3 (Top-k optimality for the bound) *Suppose optional entries have equal storage costs and budget feasibility depends only on their count. Let $\mathcal{F}_t$ contain all retained sets that include $\mathcal{A}_t$ and at most k_t optional entries. The set $\mathcal{S}_t$ in Eq. 18 satisfies*

$$\mathcal{U}(\mathcal{S}_t) = \min_{\mathcal{S} \in \mathcal{F}_t} \mathcal{U}(\mathcal{S}), \qquad \mathcal{L}(\mathcal{S}_t) \le |\mathcal{I}| \min_{\mathcal{S} \in \mathcal{F}_t} \mathcal{L}(\mathcal{S}). \tag{19}$$

Theorem 4.3 explains why one ranking is sufficient to minimize the additive bound. The second inequality compares its worst-group discarded mass with the best possible value under the same count budget. For a usable $\mathcal{S}_t$, Theorem 4.1 also gives $\mathcal{E}(\mathcal{S}_t; \mathbf{V}_t) \le D_{\mathrm{V}} \min_{\mathcal{S} \in \mathcal{F}_t} \mathcal{U}(\mathcal{S})$. These are guarantees for fixed observed queries, not exact minimization of actual output error or final video-quality loss.

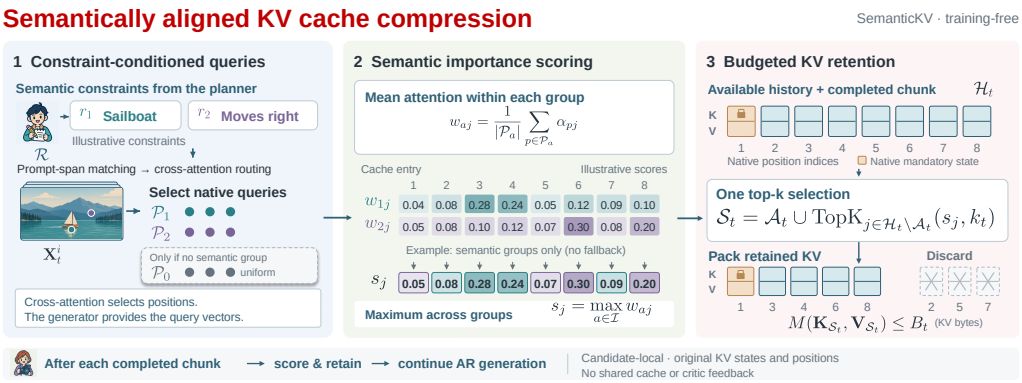


Figure 2: Overview of SEMANTICKV. Semantic constraints $\mathcal{R}$ are matched to prompt tokens. Text cross-attention identifies related video positions, from which we collect the generator's self-attention queries. Their attention to each KV entry is averaged within groups and maximized across groups. A single top-k selection retains high-scoring entries within the byte budget, together with states required by the backbone. Retained keys, values, and positions remain unchanged.

Returning to autoregressive generation. If all available entries fit, the cache remains unchanged. Otherwise, we score once, select once, and pack the retained keys, values, and position records in their original logical order, breaking score ties by original position index. Retained vectors and positions remain unchanged, with no rescoring after deletion. The compressed cache generates the next chunk as in Eq. 10; subsequent updates consider retained history together with new KV states. Both candidates use the same policy and fixed $\mathcal{R}$, but their own prompts, queries, and caches, without critic feedback during generation. Query collection, scoring, selection, and packing count toward runtime, and their temporary buffers count toward peak memory. Figure 2 illustrates the complete procedure, and Algorithm 1 summarizes the cache update.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Models. We evaluate four autoregressive video diffusion models spanning different model designs and parameter scales: LongLive (Yang et al., 2026c), LongLive 2.0 (Chen et al., 2026b), Self Forcing (Huang et al., 2025a), and Causal Forcing (Zhu et al., 2026). The evaluated models cover a broad range of model sizes, with parameter counts ranging from 1.3B to 5B.

Baselines. We compare SEMANTICKV with representative KV-cache compression methods that provide publicly available implementations and can be reliably integrated into autoregressive video diffusion models. Specifically, we compare with **Full KV**, **R-KV** (Cai et al., 2025), **Dummy Forcing** (Guo et al., 2026), and **Forcing-KV** (Ji et al., 2026). For all compressed methods, we follow their official implementations and apply the same KV-cache compression ratio whenever possible to ensure a fair comparison. Appendix B.2 provides further details.

Benchmarks. We evaluate video generation quality using three complementary benchmarks: VBench (Huang et al., 2024), VBenchLong (Huang et al., 2025c), and VisionReward (Xu et al., 2026). For short-video evaluation, we generate 5-second videos using all 946 official VBench prompts. For long-video evaluation, we use 128 prompts from MovieGen (Polyak et al., 2024) to generate 30-second and 60-second videos, with five random seeds for each prompt. We additionally employ VisionReward on the 100-prompt evaluation set used by Causal Forcing. We evaluate all 16 VBench dimensions. Detailed metric definitions appear in Appendix B.4.

Implementation Details. All experiments are run on a server equipped with 8× NVIDIA H20 96GB GPUs. Our implementation uses PyTorch for video generation and serves Qwen3.5-9B through vLLM for semantic planning and pairwise visual verification. The default launcher assigns

Table 1: Five-second VBench at compression ratio 0.7: Total Score, temporal quality, and six shared semantic diagnostics. Table 3 gives all 16 dimensions. Full KV is the uncompressed reference; efficiency is DiT-only. ↑/↓: higher/lower is better. **Bold**: best compressed result for Total Score.

Backbone	Method	Efficiency		KV Cache		Temporal Quality			VBench: Overall and Semantic Fidelity						
		FPS↑	Speedup↑	Memory (GB)↓	Reduction (×)↑	Disc.↓	Dynamic↑	Smooth.↑	Total Score↑	Object Class↑	Multiple Objects↑	Human Action↑	Color↑	Spatial Relation↑	Scene↑
Self Forcing	Full KV	6.95	1.00×	6.04	1.00×	2.25	91.67	97.77	82.96	89.83	75.55	92.00	90.54	71.67	48.05
	R-KV	7.36	1.06×	1.81	3.35×	2.23	92.78	97.77	82.90	88.29	74.15	93.00	91.40	71.66	47.53
	Dummy Forcing	8.16	1.17×	1.81	3.35×	2.23	91.39	97.81	82.87	89.26	77.07	92.00	91.12	71.01	47.43
	Forcing-KV	7.92	1.14×	1.81	3.35×	2.62	92.78	97.55	82.69	88.21	75.88	92.60	90.64	71.44	47.72
	SEMANTICKV	7.95	1.14×	1.81	3.35×	2.26	93.33	97.80	**83.03**	89.43	75.47	92.00	91.30	72.46	48.65
Causal Forcing	Full KV	7.24	1.00×	6.04	1.00×	2.39	83.61	97.87	83.99	91.11	75.88	90.00	88.72	77.57	46.25
	R-KV	8.14	1.13×	1.81	3.35×	2.54	86.39	97.72	84.00	92.88	76.77	91.40	87.89	75.60	45.20
	Dummy Forcing	9.20	1.27×	1.81	3.35×	2.53	86.39	97.78	84.05	91.72	76.34	91.60	88.12	77.57	43.88
	Forcing-KV	8.91	1.23×	1.81	3.35×	2.97	90.00	97.39	84.04	92.74	79.19	89.80	89.27	78.18	43.18
	SEMANTICKV	8.83	1.22×	1.81	3.35×	2.52	86.67	97.72	**84.10**	91.85	77.09	91.60	89.09	78.01	45.35

seven GPUs to independent generation workers and one GPU to planning and verification. All four video backbones use BF16 precision and four denoising steps; model-specific resolutions, frame rates, and chunk sizes are detailed in Appendix B.1. We evaluate five base seeds, 42–46, generating two candidates sequentially per run. The critic samples $m = 6$ frames per candidate, and candidate selection uses $\delta = 1.0$, $\tau = 0.55$, and utility weights $(w_{\mathrm{v}}, w_{\mathrm{s}}, w_{\mathrm{t}}, w_{\mathrm{m}}) = (0.4, 0.2, 0.3, 0.1)$. Both the planner and critic use temperature zero.

5.2 Main Results

Short-video quality and efficiency. Table 1 compares 5-second video generation on Self Forcing and Causal Forcing. SemanticKV achieves the highest Total Score among all compressed methods on both backbones, reaching 83.03 on Self Forcing and 84.10 on Causal Forcing, outperforming the strongest compressed baselines by 0.13 and 0.05 points, respectively. Meanwhile, SemanticKV increases DiT-only throughput from 6.95 to 7.95 FPS on Self Forcing and from 7.24 to 8.83 FPS on Causal Forcing, corresponding to 1.14× and 1.22× speedups over Full KV. It also reduces KV-cache memory from 6.04 GB to 1.81 GB, corresponding to a 3.35× reduction. These results demonstrate that SemanticKV provides a favorable balance between generation quality, memory efficiency, and inference speed. The complete results across all 16 VBench dimensions are reported in Table 3 in Appendix E.

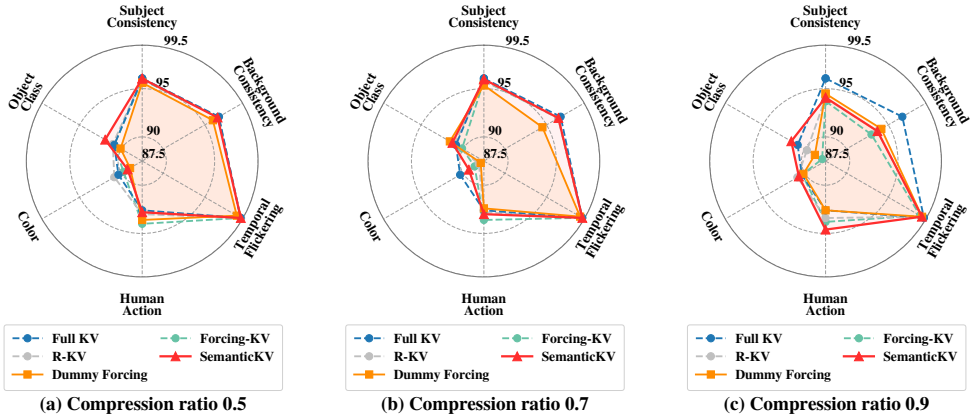
Quality across compression ratios. Figure 3 shows that SemanticKV maintains strong semantic quality across compression ratios of 0.5, 0.7, and 0.9. It achieves the best compressed result on several semantic dimensions, with especially clear gains at 0.9, where Object Class improves by up to 3.69 points over competing methods and Human Action reaches 94.60. Meanwhile, throughput increases from 8.00 to 10.00 FPS, reaching a 1.52× speedup over Full KV at 0.9. These results demonstrate a favorable quality–efficiency trade-off across different compression ratios.

Long-horizon generation. On LongLive 2.0, SemanticKV achieves 1.27× and 1.29× DiT-only speedups for 30- and 60-second generation, while the Total Score decreases by only 0.78 and 1.28 points relative to Full KV, respectively. These results show that SemanticKV extends its efficiency gains to substantially longer autoregressive generation. Complete VBenchLong results are provided in Appendix E.3.

Preference-based evaluation. On the VisionReward benchmark, SemanticKV achieves an instruction-following score of 98.000, matching the best compressed methods and exceeding Full KV and Dummy Forcing by 2.0 points. Its overall VisionReward score is also higher than R-KV and Forcing-KV. Detailed preference-based results are reported in Appendix E.4.

5.3 Ablation Studies

Component ablations. We ablate constraint-conditioned query selection and groupwise aggregation under the same cache budget. SemanticKV improves the Semantic Score from 76.94 to 77.18 over uniform queries and from 76.78 to 77.18 over pooled constraints, while maintaining



(a) Compression ratio 0.5 (b) Compression ratio 0.7 (c) Compression ratio 0.9

Figure 3: Five-second VBench across compression ratios $0.5/0.7/0.9$, with the same five methods and six dimensions; higher is better. Table 4 gives exact scores and DiT-only efficiency.

similar throughput. These results support the contributions of both semantic query selection and constraint-wise aggregation. Complete results are reported in Appendix F.1.

Query limit. With the cache budget fixed, $n_q = 32$ achieves the highest Total Score (81.68) among $n_q \in \{16, 32, 64\}$ while reaching 10.00 FPS. We therefore use $n_q = 32$ as the default setting; detailed results are provided in Appendix F.2.

5.4 Additional Ablation Studies

Paired semantic analysis. On matched prompt–seed outputs, SemanticKV shows positive gains in 15 of 18 semantic dimension–baseline comparisons. The six-dimension macro improvements over R-KV, Dummy Forcing, and Forcing-KV are 1.00, 0.86, and 1.25 points, respectively, indicating a consistent advantage in semantic fidelity. Detailed paired results and bootstrap intervals are provided in Appendix G.1.

Semantic complexity. We further analyze performance with respect to the number of planner-extracted semantic constraints. The gains of SemanticKV become more pronounced on multi-constraint prompts, reaching macro improvements of 4.62, 3.26, and 7.24 points over R-KV, Dummy Forcing, and Forcing-KV for two-constraint prompts. Results for different semantic-complexity groups are reported in Appendix G.2.

Temporal semantic drift. We also evaluate within-video semantic drift using CLIP-based prompt alignment and visual similarity. The compressed methods show only small differences, with all paired confidence intervals between SemanticKV and the compressed baselines including zero. This suggests that the semantic gains of SemanticKV do not introduce detectable additional temporal drift on this subset. Detailed results are provided in Appendix G.3.

6 Conclusion

We introduce SemanticKV, a training-free method for semantically aligned KV cache compression in agentic autoregressive video generation. Its central idea is to use the planner's semantic constraints not only to guide generation, but also to guide cache retention. Constraint-conditioned queries provide groupwise attention scores, and one top-k selection retains high-scoring KV entries under the byte budget. The method preserves native keys, values, and positions and leaves the outer agentic workflow unchanged. Its practical benefit is assessed through matched semantic-quality, cache-memory, and throughput measurements rather than inferred from attention scores alone.

Statement on LLM Usage

LLMs contributed to the research workflow by supporting idea exploration, experimental planning, literature discovery, manuscript drafting and revision, and mathematical reasoning. They were additionally used for language refinement and code-related assistance. All resulting suggestions, derivations, references, implementations, and interpretations were carefully examined and independently validated by the authors. The authors remain solely responsible for the scientific content, methodological choices, reported results, and conclusions of this paper.

Ethics Statement

The research presented in this paper does not involve human participants, personally identifiable information, or sensitive private data. We have conducted this work in accordance with the ICLR Code of Ethics and are not aware of any particular ethical risks arising from the proposed method or the reported experiments.

Reproducibility Statement

The experimental configuration, including the evaluated models, datasets, baselines, implementation details, and evaluation procedures, is described in the Experiments section. We additionally provide our code as part of the supplementary materials to support reproducibility. The complete implementation will be released publicly after acceptance.

References

Sand. ai, Hansi Teng, Hongyu Jia, Lei Sun, Lingzhi Li, Maolin Li, Mingqiu Tang, Shuai Han, Tianning Zhang, W. Q. Zhang, Weifeng Luo, Xiaoyang Kang, Yuchen Sun, Yue Cao, Yunpeng Huang, Yutong Lin, Yuxin Fang, Zewei Tao, Zheng Zhang, Zhongshu Wang, Zixun Liu, Dai Shi, Guoli Su, Hanwen Sun, Hong Pan, Jie Wang, Jiexin Sheng, Min Cui, Min Hu, Ming Yan, Shucheng Yin, Siran Zhang, Tingting Liu, Xianping Yin, Xiaoyu Yang, Xin Song, Xuan Hu, Yankai Zhang, and Yuqiao Li. Magi-1: Autoregressive video generation at scale, 2025. URL https://arxiv.org/abs/2505.13211.

Zefan Cai, Wen Xiao, Hanshi Sun, Cheng Luo, Yikai Zhang, Ke Wan, Yucheng Li, Yeyang Zhou, Li-Wen Chang, Jiuxiang Gu, Zhen Dong, Animashree Anandkumar, Abedelkadir Asi, and Junjie Hu. R-kv: Redundancy-aware kv cache compression for reasoning models. *Advances in Neural Information Processing Systems*, 38:60980–61005, 2025.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

Guibin Chen, Dixuan Lin, Jiangping Yang, Chunze Lin, Junchen Zhu, Mingyuan Fan, Hao Zhang, Sheng Chen, Zheng Chen, Chengcheng Ma, et al. Skyreels-v2: Infinite-length film generative model. *arXiv preprint arXiv:2504.13074*, 2025.

Hanmo Chen, Chenghao Xu, Xu Yang, Xuan Chen, and Cheng Deng. Past-and future-informed kv cache policy with salience estimation in autoregressive video diffusion. *arXiv preprint arXiv:2601.21896*, 2026a.

Yukang Chen, Luozhou Wang, Wei Huang, Shuai Yang, Bohan Zhang, Yicheng Xiao, Ruihang Chu, Weian Mao, Qixin Hu, Shaoteng Liu, et al. Longlive-2.0: An nvfp4 parallel infrastructure for long video generation. *arXiv preprint arXiv:2605.18739*, 2026b.

Jiale Cheng, Ruiliang Lyu, Xiaotao Gu, Xiao Liu, Jiazheng Xu, Yida Lu, Jiayan Teng, Zhuoyi Yang, Yuxiao Dong, Jie Tang, et al. Vpo: Aligning text-to-video generation models with prompt optimization. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15636–15645. IEEE, 2025.

Nicolae Cudlenco, Mihai Masala, and Marius Leordeanu. Authoring for living worlds: Tool-constrained llm agents for executable multi-actor scenarios, 2026.

Haoge Deng, Ting Pan, Haiwen Diao, Zhengxiong Luo, Yufeng Cui, Huchuan Lu, Shiguang Shan, Yonggang Qi, and Xinlong Wang. Autoregressive video generation without vector quantization. In *International Conference on Learning Representations*, volume 2025, pp. 44730–44745, 2025.

Yuan Feng, Junlin Lv, Yukun Cao, Xike Xie, and S Kevin Zhou. Ada-kv: Optimizing kv cache eviction by adaptive budget allocation for efficient llm inference. *Advances in Neural Information Processing Systems*, 38:113152–113188, 2025a.

Yuan Feng, Junlin Lv, Haoyu Guo, Yukun Cao, S Kevin Zhou, and Xike Xie. Criticalkv: Optimizing kv cache eviction from an output perturbation perspective. *arXiv preprint arXiv:2502.03805*, 2025b.

Yuxiang Feng, Juncheng Wang, Chao Xu, Yijie Qian, Huihan Wang, Wenlong Hou, Yang Liu, Baigui Sun, Yong Liu, and Shujun Wang. Newton: Agentic planning for physically grounded video generation. *arXiv preprint arXiv:2605.18396*, 2026.

Xiaomeng Fu, Jia Li, Yiming Hu, Yong Wang, Hayden Kwok-Hay So, Jiao Dai, Xiangxiang Chu, and Jizhong Han. Towards memory-efficient autoregressive video generation via instance-specific parametric absorption. *arXiv preprint arXiv:2607.00712*, 2026.

Kaifeng Gao, Jiaxin Shi, Hanwang Zhang, Chunping Wang, and Jun Xiao. Vid-gpt: Introducing gpt-style autoregressive generation in video diffusion models. *arXiv preprint arXiv:2406.10981*, 2024.

Kaifeng Gao, Jiaxin Shi, Hanwang Zhang, Chunping Wang, Jun Xiao, and Long Chen. Ca2-VDM: Efficient autoregressive video diffusion model with causal generation and cache sharing. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 18550–18565. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/gao25m.html.

Yuzhe Gu, Xiyu Liang, Jiaojiao Zhao, and Enmao Diao. Obcache: Optimal brain kv cache pruning for efficient long-context llm inference. *arXiv preprint arXiv:2510.07651*, 2025.

Hang Guo, Zhaoyang Jia, Jiahao Li, Bin Li, Yuanhao Cai, Jiangshan Wang, Yawei Li, and Yan Lu. Efficient autoregressive video diffusion with dummy head. *arXiv preprint arXiv:2601.20499*, 2026.

Yuwei Guo, Ceyuan Yang, Hao He, Yang Zhao, Meng Wei, Zhenheng Yang, Weilin Huang, and Dahua Lin. End-to-end training for autoregressive video diffusion via self-resampling. *arXiv preprint arXiv:2512.15702*, 2025.

Jitai Hao, Qiang Huang, Yaowei Wang, Min Zhang, and Jun Yu. Deltakv: Residual-based kv cache compression via long-range similarity. *arXiv preprint arXiv:2602.08005*, 2026.

Xuan He, Dongfu Jiang, Ge Zhang, Max Ku, Achint Soni, Sherman Siu, Haonan Chen, Abhranil Chandra, Ziyan Jiang, Aaran Arulraj, et al. Videoscore: Building automatic metrics to simulate fine-grained human feedback for video generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 2105–2123, 2024.

Roberto Henschel, Levon Khachatryan, Hayk Poghosyan, Daniil Hayrapetyan, Vahram Tadevosyan, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Streamingt2v: Consistent, dynamic, and extendable long video generation from text. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2568–2577. IEEE, 2025.

Panwen Hu, Jin Jiang, Jianqi Chen, Mingfei Han, Shengcai Liao, Xiaojun Chang, and Xiaodan Liang. Storyagent: Customized storytelling video generation via multi-agent collaboration. *arXiv preprint arXiv:2411.04925*, 2024.

Kaiyi Huang, Yukun Huang, Xuefei Ning, Zinan Lin, Yu Wang, and Xihui Liu. Genmac: compositional text-to-video generation with multi-agent collaboration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 5049–5057, 2026a.

Lingxuan Huang, Sizhe He, Hengji Zhou, Liqiang Nie, Lianghao Xia, and Chao Huang. Vimax: Agentic video generation. *arXiv preprint arXiv:2606.07649*, 2026b.

Weizhong Huang, Jinchao Zhang, and Xiawu Zheng. Commitkv: Lifecycle-aware kv cache compression via commit transitions for multi-turn agents. *arXiv preprint arXiv:2608.07855*, 2026c.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *Advances in Neural Information Processing Systems*, 38:167283–167308, 2025a.

Yidong Huang, Zun Wang, Han Lin, Dong-Ki Kim, Shayegan Omidshafiei, Jaehong Yoon, Yue Zhang, and Mohit Bansal. Planning with sketch-guided verification for physics-aware video generation. *arXiv preprint arXiv:2511.17450*, 2025b.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818. IEEE, 2024.

Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, et al. Vbench++: Comprehensive and versatile benchmark suite for video generative models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025c.

Yicheng Ji, Zhizhou Zhong, Jun Zhang, Qin Yang, XiTai Jin, Ying Qin, Wenhan Luo, Shuiyang Mao, Wei Liu, and Huan Li. Forcing-kv: Hybrid kv cache compression for efficient autoregressive video diffusion models. *arXiv preprint arXiv:2605.09681*, 2026.

Kunyang Li, Mubarak Shah, and Yuzhang Shang. Packcache: A training-free acceleration method for unified autoregressive video generation via compact kv-cache. *arXiv preprint arXiv:2601.04359*, 2026a.

Wuyang Li, Wentao Pan, Po-Chien Luan, Yang Gao, and Alexandre Alahi. Stable video infinity: Infinite-length video generation with error recycling. In *International Conference on Learning Representations*, volume 2026, pp. 23406–23432, 2026b.

Yu Li, Binxu Li, and Tian Lan. Momentkv: Closing the directional gap in kv cache eviction for long-context inference. *arXiv preprint arXiv:2606.01563*, 2026c.

Zhengyang Liang, Daoan Zhang, Huichi Zhou, Rui Huang, Bobo Li, Yuechen Zhang, Shengqiong Wu, Xiaohan Wang, Jiebo Luo, Lizi Liao, et al. Univa: Universal video agent towards open-source next-generation video generalist. *arXiv preprint arXiv:2511.08521*, 2025.

Han Lin, Abhay Zala, Jaemin Cho, and Mohit Bansal. Videodirectorgpt: Consistent multi-scene video generation via llm-guided planning. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=sKNIjS2brr.

Shanchuan Lin, Ceyuan Yang, Hao He, Jianwen Jiang, Yuxi Ren, Xin Xia, Yang Zhao, Xuefeng Xiao, and Lu Jiang. Autoregressive adversarial post-training for real-time interactive video generation. *Advances in Neural Information Processing Systems*, 38:41061–41086, 2025.

Fangfu Liu, Hanyang Wang, Yimo Cai, Kaiyan Zhang, Xiaohang Zhan, and Yueqi Duan. Video-t1: Test-time scaling for video generation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18671–18681. IEEE, 2025.

Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. Rolling forcing: Autoregressive long video diffusion in real time. In *International Conference on Learning Representations*, volume 2026, pp. 91177–91196, 2026.

Do Xuan Long, Xingchen Wan, Hootan Nakhost, Chen-Yu Lee, Tomas Pfister, and Sercan Ö Arik. Vista: A test-time self-improving video generation agent. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6021–6032, 2026.

Jiayi Luo, Qiyan Liu, Tengyang Wang, JunHao Liu, Jiayu Chen, Cong Wang, Hanxin Zhu, Chen Gao, Xiaobin Hu, Qingyun Sun, et al. Future forcing: Future-aware training-free kv cache policy for autoregressive video generation. *arXiv preprint arXiv:2605.30083*, 2026.

Chengtao Lv, Yumeng Shi, Yushi Huang, Ruihao Gong, Shen Ren, and Wenya Wang. Light forcing: Accelerating autoregressive video diffusion via sparse attention. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=NixeX51WQA.

Yuexiao Ma, Xuzhe Zheng, Jing Xu, Xiwei Xu, Feng Ling, Xiawu Zheng, Huafeng Kuang, Huixia Li, XING WANG, Xuefeng Xiao, Fei Chao, and Rongrong Ji. Flow caching for autoregressive video generation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=vko4DuhKbh.

Xiaofeng Mao, Shaohao Rui, Kaining Ying, Bo Zheng, Chuanhao Li, Mingmin Chi, and Kaipeng Zhang. Packforcing: Short video training suffices for long video sampling and long context inference. *arXiv preprint arXiv:2603.25730*, 2026.

Yu Meng, Xiangyang Luo, Letian Li, Wenyuan Jiang, Chen Gao, Xinlei Chen, Yong Li, and Xiao-Ping Zhang. Tethercache: Stabilizing autoregressive long-form video generation with gated recall and trusted alignment. *arXiv preprint arXiv:2606.13035*, 2026.

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Dvir Samuel, Issar Tzachor, Matan Levy, Michael Green, Gal Chechik, and Rami Ben-Ari. Fast-ar: Fast autoregressive video diffusion and world models with temporal cache compression and sparse attention. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=UxgMJP0JKY.

Achint Soni, Sreyas Venkataraman, Abhranil Chandra, Sebastian Fischmeister, Percy Liang, Bo Dai, and Sherry Yang. Videoagent: Self-improving video generation. *arXiv preprint arXiv:2410.10076*, 2024.

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pp. 402–419. Springer, 2020.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Jiangshan Wang, Kang Zhao, Jiayi Guo, Jiayu Wang, Hang Guo, Chenyang Zhu, Xiu Li, and Xiangyu Yue. Precisecache: Precise feature caching for efficient and high-fidelity video generation. *arXiv preprint arXiv:2603.00976*, 2026.

Jing Wang, Fengzhuo Zhang, Xiaoli Li, Vincent YF Tan, Tianyu Pang, Chao Du, Aixin Sun, and Zhuoran Yang. Error analyses of auto-regressive video diffusion models: A unified framework. *arXiv preprint arXiv:2503.10704*, 2025.

Jiuniu Wang, Zehua Du, Yuyuan Zhao, Bo Yuan, Kexiang Wang, Jian Liang, Yaxi Zhao, Yihen Lu, Gengliang Li, Junlong Gao, et al. Aesopagent: Agent-driven evolutionary system on story-to-video production. *arXiv preprint arXiv:2403.07952*, 2024a.

Yuqing Wang, Tianwei Xiong, Daquan Zhou, Zhijie Lin, Yang Zhao, Bingyi Kang, Jiashi Feng, and Xihui Liu. Loong: Generating minute-level long videos with autoregressive language models. *arXiv preprint arXiv:2410.02757*, 2024b.

Weijia Wu, Zeyu Zhu, and Mike Zheng Shou. Automated movie generation via multi-agent cot planning. *arXiv preprint arXiv:2503.07314*, 2025.

Haocheng Xi, Shuo Yang, Yilong Zhao, Muyang Li, Han Cai, Xingyang Li, Yujun Lin, Zhuoyang Zhang, Jintao Zhang, Xiuyu Li, Zhiying Xu, Jun Wu, Chenfeng Xu, Ion Stoica, Song Han, and Kurt Keutzer. Quant videogen: Auto-regressive long video generation via 2-bit KV-cache quantization. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=r9UJgVH1Fx.

Desai Xie, Zhan Xu, Yicong Hong, Hao Tan, Difan Liu, Feng Liu, Arie Kaufman, and Yang Zhou. Progressive autoregressive video diffusion models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 6312–6322. IEEE, 2025.

Zhifei Xie, Daniel Tang, Dingwei Tan, Jacques Klein, Tegawend F Bissyand, and Saad Ezzini. Dreamfactory: Pioneering multi-scene long video generation with a multi-agent framework. *arXiv preprint arXiv:2408.11788*, 2024.

Zhening Xing, Gereon Fox, Yanhong Zeng, Xingang Pan, Mohamed Elgharib, Christian Theobalt, and Kai Chen. Live2diff: Live stream translation via uni-directional attention in video diffusion models. *arXiv preprint arXiv:2407.08701*, 2024.

Jiazheng Xu, Yu Huang, Jiale Cheng, Yuanming Yang, Jiajun Xu, Yuan Wang, Wenbo Duan, Shen Yang, Qunlin Jin, Shurun Li, et al. Visionreward: Fine-grained multi-dimensional human preference learning for image and video generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 11269–11277, 2026.

Zhenran Xu, Longyue Wang, Jifang Wang, Zhouyi Li, Senbao Shi, Xue Yang, Yiyu Wang, Baotian Hu, Jun Yu, and Min Zhang. Filmagent: A multi-agent framework for end-to-end film automation in virtual 3d spaces. *arXiv preprint arXiv:2501.12909*, 2025.

Chengyi Yang, Pengzhen Li, Jiayin Qi, Aimin Zhou, Ji Wu, and Ji Liu. Scmapr: Self-correcting multi-agent prompt refinement for complex-scenario text-to-video generation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 6942–6973, 2026a.

Jiaming Yang, Chenwei Tang, Liangli Zhen, Chenyang Zhang, and Jiancheng Lv. Jacap: Robust kv cache eviction via jacobian-based nonlinear information capacity preservation, 2026b.

Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, YINGCONG CHEN, Yao Lu, Song Han, and Yukang Chen. Longlive: Real-time interactive long video generation. In *International Conference on Learning Representations*, volume 2026, pp. 90037–90059, 2026c.

Yu Yang, Yue Liao, Jianbiao Mei, Baisen Wang, Xuemeng Yang, Licheng Wen, Jiangning Zhang, Xiangtai Li, Liang Lv, Hanlin Chen, et al. Spiral: Self-evolving action-conditioned video generation via reflective planning agents. *arXiv preprint arXiv:2603.08403*, 2026d.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22963–22974. IEEE, 2025.

Yifei Yu, Xiaoshan Wu, Xinting Hu, Tao Hu, Yangtian Sun, Xiaoyang Lyu, Bo Wang, Lin Ma, Yuewen Ma, Zhongrui Wang, et al. Videossm: Autoregressive long video generation with hybrid state-space memory. *arXiv preprint arXiv:2512.04519*, 2025.

Zhengqing Yuan, Yixin Liu, Yihan Cao, Weixiang Sun, Haolong Jia, Ruoxi Chen, Zhaoxu Li, Bin Lin, Li Yuan, Lifang He, et al. Mora: Enabling generalist video generation via a multi-agent framework. *arXiv preprint arXiv:2403.13248*, 2024.

Qinglin Zeng, Kaitong Cai, Ruiqi Chen, Qinhan Lv, and Keze Wang. Coagent: Collaborative planning and consistency agent for coherent video generation. *arXiv preprint arXiv:2512.22536*, 2025.

Ruicheng Zhang, Kaixi Cong, Jun Zhou, Zhizhou Zhong, Zunnan Xu, Shuiyang Mao, Wei Liu, and Xiu Li. Kvpo: Ode-native grpo for autoregressive video alignment via kv semantic exploration. *arXiv preprint arXiv:2605.14278*, 2026.

Wenqu Zhao, Xuemin Chi, Xin Zhang, Guoqing Ma, Baorun Li, Jianjie Fang, Peizhi Tang, Chen Gao, and Wei Wu. Densitykv: Density-guided kv cache compression for long video generation. *arXiv preprint arXiv:2608.27922*, 2026.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

Hongzhou Zhu, Min Zhao, Guande He, Hang Su, Chongxuan Li, and Jun Zhu. Causal forcing: Autoregressive diffusion distillation done right for high-quality real-time interactive video generation. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=BYInOck3gr.

Shaobin Zhuang, Kunchang Li, Xinyuan Chen, Yaohui Wang, Ziwei Liu, Yu Qiao, and Yali Wang. Vlogger: Make your dream a vlog. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8806–8817. IEEE, 2024.

Adam Zweiger, Xinghong Fu, Han Guo, and Yoon Kim. Fast kv compaction via attention matching. *arXiv preprint arXiv:2602.16284*, 2026.

APPENDIX

A Detailed Algorithm **16**

B Details of Models, Baselines, Metrics and Benchmarks **16**

B.1 Details of Models 16

B.2 Details of Baselines 17

B.3 Details of Benchmarks 17

B.4 Details of Evaluation Metrics 18

C Details of Semantic Constraints and Prompt Refinement **19**

D Theoretical Justification of Semantic KV Retention **19**

D.1 From Discarded Attention to Constraint-Wise Output Error 20

D.2 Why the Maximum Score Leads to One Top-k Selection 21

D.3 Why Keep Semantic Groups Separate? 22

E Detailed Benchmark Results **22**

E.1 VBench across Backbones 22

E.2 VBench across Compression Ratios 22

E.3 Long-Horizon Evaluation against Full KV 24

E.4 VisionReward Results 24

F Detailed Ablation Results **25**

F.1 Semantic Query Selection and Aggregation 25

F.2 Semantic Query Limit 25

G Additional Semantic and Temporal Analyses **26**

G.1 Paired Sample-Level Reliability 26

G.2 Performance versus Semantic Complexity 27

G.3 Within-Video Temporal Semantic Drift 27

H Qualitative Comparisons **29**

A Detailed Algorithm

Algorithm 1 summarizes the cache-update procedure of SemanticKV, combining constraint-conditioned query selection, groupwise attention scoring, and budgeted top-k retention. Attention weights are averaged within each semantic group and maximized across groups to rank KV entries. The procedure preserves backbone-required states and skips compression when all available entries fit within the budget. Otherwise, selected keys, values, and positions are packed in their original logical order without modification.

Algorithm 1 SemanticKV: core steps of a semantically aligned cache update

Require: Available cache $(\mathbf{K}_t, \mathbf{V}_t)$ indexed by $\mathcal{H}_t$; semantic constraints $\mathcal{R}$; candidate prompt $\mathbf{q}^i$
Require: Observed self-attention queries and text cross-attention; unit budget B_t determined by compression ratio r; query limit n_q
Ensure: Compressed cache $(\widetilde{\mathbf{K}}_t, \widetilde{\mathbf{V}}_t)$ within B_t
Identify mandatory entries $\mathcal{A}_t$ and check budget feasibility
if all available entries fit within B_t **then**
    return The unchanged cache $(\mathbf{K}_t, \mathbf{V}_t)$
end if
Match each constraint r_a to prompt-token positions $\mathcal{T}_a$
Compute routing scores $g_a(x)$ using Eq. 12
Collect self-attention queries at up to n_q top-ranked positions per reliably matched constraint; form index groups $\mathcal{P}_a$
Let $\mathcal{I}$ index all nonempty query groups used for scoring
Compute masked cache-attention weights α_{pj} using Eq. 13
Compute group means w_{aj} and semantic importance s_j using Eq. 14
Determine k_t from B_t, accounting for mandatory states, positions, metadata, and allocation granularity
Select retained indices $\mathcal{S}_t$ using Eq. 18
return $(\mathbf{K}_{\mathcal{S}_t}, \mathbf{V}_{\mathcal{S}_t})$ packed in original logical order with unchanged positions

B Details of Models, Baselines, Metrics and Benchmarks

B.1 Details of Models

Basic information. We evaluate four autoregressive video diffusion models spanning different model scales and generation designs.

- **LongLive** (Yang et al., 2026c). LongLive is a 1.3B-parameter autoregressive video diffusion model for real-time long-video generation. It adopts frame-level causal generation with KV caching and introduces a frame sink, short attention window, and KV recaching to support efficient long-horizon and interactive generation.

- **LongLive 2.0** (Chen et al., 2026b). LongLive 2.0 is a 5B-parameter autoregressive video diffusion model designed for scalable long-video generation. It extends the LongLive framework to a larger backbone and supports long-horizon generation with KV-cached causal attention and efficient sequence-parallel inference.
- **Self Forcing** (Huang et al., 2025a). Self Forcing is a 1.3B-parameter autoregressive video diffusion model based on Wan2.1-T2V-1.3B (Wan et al., 2025). It performs causal video generation with temporal KV caching and supports both frame-wise and chunk-wise autoregressive generation, making it a representative backbone for efficient short-video generation.
- **Causal Forcing** (Zhu et al., 2026). Causal Forcing is a 1.3B-parameter autoregressive video diffusion model derived from Wan2.1-T2V-1.3B (Wan et al., 2025). It performs few-step causal generation with temporal KV caching and provides an additional autoregressive formulation for evaluating the generalization of our method.

Model-specific inference configurations. We use the following inference configurations for the four backbones. Resolutions are reported as width $\times$ height, and chunk sizes are measured in latent frames. The video frame rates below describe playback rather than the generation throughput reported in our efficiency evaluation.

- **LongLive (v1).** We use BF16 precision, 832×480 resolution, a video frame rate of 16 FPS, four denoising steps, and three latent frames per autoregressive chunk.
- **LongLive 2.0.** We use BF16 precision, 1280×704 resolution, a video frame rate of 24 FPS, four denoising steps, and eight latent frames per autoregressive chunk.
- **Self Forcing.** We use BF16 precision, 832×480 resolution, a video frame rate of 16 FPS, four denoising steps, and three latent frames per autoregressive chunk.
- **Causal Forcing.** We use BF16 precision, 832×480 resolution, a video frame rate of 16 FPS, four denoising steps, and three latent frames per autoregressive chunk.

B.2 Details of Baselines

- **Full KV.** Full KV serves as the uncompressed reference and applies no additional cache eviction. Its configuration should specify the backbone's native attention mask, context window, sink states, and re-caching behavior, so that the reference cache footprint is unambiguous.
- **R-KV (Cai et al., 2025).** R-KV selects cached tokens using importance and redundancy. Its adaptation to video generation must specify the source of the importance scores, the treatment of video-token positions, and the cache-update schedule. Any changes required by chunk-wise denoising should be documented separately from the original selection rule.
- **Dummy Forcing (Guo et al., 2026).** Dummy Forcing allocates historical context according to attention-head locality. Reproduction requires the head-classification procedure, profiling inputs, retained context for each head category, and the frequency of classification. Profiling and cache-layout conversion costs belong in the measured inference time.
- **Forcing-KV (Ji et al., 2026).** Forcing-KV applies different retention policies to static and dynamic heads. The configuration should disclose the head-partition criterion, category-specific retention settings, protected context, and compression schedule. The realized cache footprint should be reported in addition to the method's nominal compression parameters.

B.3 Details of Benchmarks

We evaluate video generation quality using benchmarks covering short-video quality, long-horizon consistency, and human preference.

- **VBench** (Huang et al., 2024). VBench is a comprehensive benchmark for video generation that evaluates 16 fine-grained dimensions covering visual quality, temporal consistency, motion, and semantic alignment. We use all 946 official prompts for 5-second video evaluation.

- **VBenchLong** (Huang et al., 2025c). VBenchLong extends fine-grained video evaluation to long-duration generation, emphasizing temporal consistency, motion quality, and visual stability over extended sequences. We evaluate 30-second and 60-second videos using 128 MovieGen prompts with five random seeds per prompt.
- **VisionReward** (Xu et al., 2026). VisionReward is a fine-grained human-preference reward model for image and video generation. It evaluates multiple aspects of perceptual quality and instruction alignment and aggregates them into an overall preference score. We follow the Causal Forcing evaluation protocol and use its 100-prompt evaluation set.

B.4 Details of Evaluation Metrics

We evaluate both generation efficiency and video quality. The efficiency metrics include frames per second (FPS) and inference speedup, while the quality evaluation covers chunk continuity, temporal dynamics, visual quality, temporal consistency, and human-preference-aligned generation quality.

Frames per Second and Speedup. We measure generation throughput using frames per second (FPS). Following prior work, the runtime is measured within the diffusion transformer (DiT), excluding the VAE-related encoding and decoding overhead. Given a generated video containing F frames and a DiT inference time of T, FPS is computed as

$$\mathrm{FPS} = \frac{F}{T}. \tag{20}$$

We further report the inference speedup relative to the corresponding Full KV configuration:

$$\mathrm{Speedup} = \frac{\mathrm{FPS}_{\mathrm{method}}}{\mathrm{FPS}_{\mathrm{FullKV}}}. \tag{21}$$

Therefore, Full KV has a speedup of $1.00\times$ by definition.

Chunk Discontinuity. Autoregressive video diffusion models generate videos chunk by chunk, making visual continuity at chunk boundaries particularly important. We quantify abrupt transitions between consecutive chunks using the chunk discontinuity metric. Given a video with F frames generated in K autoregressive chunks, we first compute the optical-flow difference between every pair of adjacent frames using RAFT (Teed & Deng, 2020):

$$\delta_t = \Delta_{\mathrm{RAFT}}(I_t, I_{t+1}), \qquad t = 1, \ldots, F-1, \tag{22}$$

where I_t denotes the t-th frame. We then identify the $K-1$ largest optical-flow differences, corresponding to the strongest temporal discontinuities, and normalize their average by the mean optical-flow difference over the entire video:

$$\mathrm{ChunkDisc.} = \frac{\frac{1}{K-1}\sum_{\delta \in \mathrm{Top}_{K-1}(\{\delta_t\})} \delta}{\frac{1}{F-1}\sum_{t=1}^{F-1} \delta_t}. \tag{23}$$

A lower score indicates smoother transitions across autoregressive chunks and fewer abrupt visual changes at chunk boundaries.

Dynamic Degree. Dynamic Degree measures the amount of motion and temporal variation in the generated video. We obtain this score using the official VBench/VBenchLong evaluation pipeline. A higher Dynamic Degree indicates richer and more pronounced motion, whereas a lower value generally corresponds to more static video content. This metric is particularly relevant for KV-cache compression, since excessively removing historical context may preserve individual-frame quality while suppressing temporal dynamics.

VBench Metrics. For short-video generation, we evaluate 5-second videos using the official VBench (Huang et al., 2024) evaluation pipeline. VBench evaluates generated videos along 16 dimensions that characterize different aspects of visual and temporal quality. Table 1 reports the total score and representative visual and temporal metrics, complemented by the main-paper figures. Table 3 in Appendix E reports all 16 VBench dimensions, while Table 4 reports the six dimensions shown in Figure 3.

Imaging Quality evaluates the visual fidelity and perceptual quality of individual frames. *Subject Consistency* measures whether the appearance and identity of the primary subject remain stable throughout the video. *Background Consistency* evaluates the temporal stability of the background scene. *Motion Smoothness* measures whether motion evolves smoothly over time without abrupt or unnatural temporal changes. *Aesthetic Quality* evaluates the overall visual appeal of the generated content. The reported VBench scores are obtained directly from the official benchmark evaluators, and higher values indicate better performance.

VBenchLong Metrics. For long-video generation, we employ VBenchLong (Huang et al., 2025c) to evaluate 30-second and 60-second videos. The main-paper figure reports overall quality, dynamic degree, and chunk discontinuity; Appendix E also provides the available imaging quality, subject consistency, background consistency, motion smoothness, and aesthetic quality measurements. The overall score is aggregated using the standard VBench weighting coefficients. Compared with short-video evaluation, these metrics additionally reflect the ability of a model to maintain visual fidelity, subject identity, scene consistency, and temporal coherence over substantially longer autoregressive rollouts.

VisionReward. We further employ VisionReward (Xu et al., 2026) as a complementary human-preference-based evaluator. Following the evaluation protocol of Causal Forcing, we evaluate generated videos on a 100-prompt subset and report both the overall VisionReward score and the instruction-following score. The overall reward measures the preference-aligned quality of the generated video, while the instruction-following score evaluates how faithfully the generated content satisfies the input text instruction. Higher values indicate better performance for both metrics.

C Details of Semantic Constraints and Prompt Refinement

Plan construction. For each unique benchmark prompt, we prepare one plan and keep it fixed during candidate generation. Given the original prompt $\mathbf{q}^0 = \mathbf{q}$, the semantic planner returns the semantic constraints $\mathcal{R}$, prompt-specific exclusions $\mathcal{N}$, and refined prompt $\mathbf{q}^1$ in a single call, as in Eq. 3. The original prompt remains unchanged.

The semantic constraints $\mathcal{R}$ list the explicit content to preserve, including subjects, counts, attributes, relations, actions, scenes, temporal cues, camera cues, styles, and the requested degree of motion. The prompt-specific exclusions $\mathcal{N}$ identify omissions, contradictions, and unrequested additions to avoid. The planner is instructed to rewrite $\mathbf{q}^0$ as a clear description that preserves the content in $\mathcal{R}$ while avoiding the changes identified by $\mathcal{N}$. It may clarify wording and composition, but must not introduce new subjects, objects, actions, scene changes, on-screen text, viewpoints, or camera motion. Explicit requests for stillness must be preserved, and a prompt that needs no refinement may remain unchanged. The three outputs are produced together by the planner; the two sets guide the rewrite rather than being concatenated to form $\mathbf{q}^1$. In particular, $\mathcal{N}$ is not supplied to the generator as a separate negative prompt. Table 2 illustrates how a fragmented request can be rewritten without adding new content.

Output checks and fallback. A returned plan contains one to ten semantic constraints and at most six prompt-specific exclusions. The planner is asked to use at most 55 English words for the refined prompt; the acceptance checks require a nonempty prompt of at most 700 characters and reject prompts exceeding 70 words. A short scenario label and a brief explanation of how the refinement preserves the request are also recorded. These checks verify output structure and length, rather than establish semantic equivalence. If the planning call or these checks fail during plan preparation, we use $\mathcal{R} = \{\mathbf{q}^0\}$, $\mathcal{N} = \varnothing$, and $\mathbf{q}^1 = \mathbf{q}^0$. This fallback preserves the original request as one constraint rather than decomposing it into individual constraints. Both candidates are subsequently assessed against the original request by the vision–language critic.

D Theoretical Justification of Semantic KV Retention

This section proves Theorems 4.1–4.3 and explains their scope. The analysis follows the existing mean–maximum score and top-k policy; it introduces no additional query sampling, optimization stage, or error threshold.

Table 2: An illustrative example of conservative semantic planning. Matching colors trace the requested content from $\mathbf{q}^0$ through $\mathcal{R}$ to $\mathbf{q}^1$: subjects and attributes, actions and their order, and scene and camera. Prompt-specific exclusions identify changes to avoid.

Component and role	Example
Original prompt $\mathbf{q}^0$ Before refinement.	"park, one girl one boy, red coat on girl blue coat on boy, yellow ball, girl gives it to boy then boy lifts it over his head, fixed camera"
Semantic constraints $\mathcal{R}$ What to preserve.	r_1: One girl in a red coat. r_2: One boy in a blue coat. r_3: A yellow ball. r_4: The girl gives the ball to the boy. r_5: The boy then lifts the ball over his head. r_6: A park. r_7: A fixed camera.
Prompt-specific exclusions $\mathcal{N}$ What to avoid.	Swapped coat colors; reversed giver and receiver; reversed action order; camera movement; additional people, objects, or actions.
Refined prompt $\mathbf{q}^1$ After refinement.	"In a park, one girl in a red coat gives a yellow ball to one boy in a blue coat. The boy **then** lifts the **same ball** over his head. The camera remains fixed."

Setting and scope. Fix one candidate, one completed-chunk update t, and one layer/KV-head unit. The available cache $\mathcal{H}_t$, observed queries, native masks, and nonempty groups $\{\mathcal{P}_a\}_{a\in\mathcal{I}}$ are those used for scoring in Section 4.2, with $\mathcal{I} \neq \varnothing$. Queries and keys are held fixed when comparing attention before and after retention. A retained set $\mathcal{S} \subseteq \mathcal{H}_t$ contains the mandatory states $\mathcal{A}_t$. Whenever an output after retention is considered, $\mathcal{S}$ must contain at least one visible entry for every selected query; we call such a set *usable*. Native mandatory states can ensure this condition, but the score alone does not. The reference is the currently available cache, not a reconstructed full-history cache. The discarded-mass bounds remain defined for every subset, even when its retained attention output would be undefined.

D.1 From Discarded Attention to Constraint-Wise Output Error

Proof of Theorem 4.1. For query p, define the discarded attention mass $m_p(\mathcal{S}) = \sum_{j\in\mathcal{H}_t\setminus\mathcal{S}} \alpha_{pj}$. Masked entries have zero weight, and the visible weights sum to one, so $0 \le m_p(\mathcal{S}) < 1$ for a usable retained set. Because the retained keys and their logits are unchanged, the original and retained attention outputs are

$$\mathbf{o}_p = \sum_{j\in\mathcal{H}_t} \alpha_{pj}\mathbf{v}_j, \qquad \mathbf{o}_p^{\mathcal{S}} = \frac{\sum_{j\in\mathcal{S}} \alpha_{pj}\mathbf{v}_j}{1 - m_p(\mathcal{S})}. \tag{24}$$

If $m_p(\mathcal{S}) = 0$, the two outputs are identical. Otherwise, let $\mathbf{o}_p^{\mathrm{drop}} = m_p(\mathcal{S})^{-1} \sum_{j\in\mathcal{H}_t\setminus\mathcal{S}} \alpha_{pj}\mathbf{v}_j$ be the normalized average of the discarded values. Splitting the original output gives

$$\mathbf{o}_p = (1 - m_p(\mathcal{S}))\mathbf{o}_p^{\mathcal{S}} + m_p(\mathcal{S})\mathbf{o}_p^{\mathrm{drop}}, \qquad \|\mathbf{o}_p - \mathbf{o}_p^{\mathcal{S}}\|_2 \le D_{\mathrm{V}}\, m_p(\mathcal{S}). \tag{25}$$

To obtain the inequality, note that both normalized outputs are convex combinations of available values. Writing their difference as a convex combination of pairwise value differences and applying the triangle inequality bounds their distance by D_{V}. Averaging the discarded mass within each group yields

$$\frac{1}{|\mathcal{P}_a|}\sum_{p\in\mathcal{P}_a} m_p(\mathcal{S}) = \sum_{j\in\mathcal{H}_t\setminus\mathcal{S}} \frac{1}{|\mathcal{P}_a|}\sum_{p\in\mathcal{P}_a} \alpha_{pj} = \sum_{j\in\mathcal{H}_t\setminus\mathcal{S}} w_{aj} = \ell_a(\mathcal{S}). \tag{26}$$

Averaging Eq. 25 and then taking the largest groupwise error proves $\mathcal{E}(\mathcal{S};\mathbf{V}_t) \le D_{\mathrm{V}}\mathcal{L}(\mathcal{S})$. □

Sharpness and interpretation. For fixed queries, keys, masks, groups, and a usable $\mathcal{S}$, the bound is sharp over value tables whose diameter is at most a prescribed $D \ge 0$:

$$\sup_{\mathbf{V}_t:\,\mathrm{diam}(\mathbf{V}_t)\le D} \mathcal{E}(\mathcal{S};\mathbf{V}_t) = D\,\mathcal{L}(\mathcal{S}). \tag{27}$$

Here, $\mathrm{diam}(\mathbf{V}_t)$ is the largest pairwise distance between the value vectors. The theorem gives the upper bound. For equality, choose a unit vector $\mathbf{u}$, assign $\mathbf{v}_j = \mathbf{0}$ to every retained entry, and assign $\mathbf{v}_j = D\mathbf{u}$ to every discarded entry. Then $\mathbf{o}_p^{\mathcal{S}} = \mathbf{0}$ and $\mathbf{o}_p = D\, m_p(\mathcal{S})\mathbf{u}$ for every query. This single shared value table attains equality for all groups simultaneously. The construction is a mathematical worst case with queries and keys frozen, not a claim that arbitrary value tables occur in the generator. Actual errors can be smaller through cancellation. No small-deletion approximation is used, and these outputs are measured before the attention output projection.

D.2 Why the Maximum Score Leads to One Top-k Selection

Proof of Theorem 4.2. Recall the additive bound formed from the mean–maximum scores:

$$\mathcal{U}(\mathcal{S}) = \sum_{j \in \mathcal{H}_t \setminus \mathcal{S}} s_j = \sum_{j \in \mathcal{H}_t \setminus \mathcal{S}} \max_{a \in \mathcal{I}} w_{aj}. \tag{28}$$

Because the attention weights are nonnegative, $w_{aj} \le s_j \le \sum_{a' \in \mathcal{I}} w_{a'j}$ for every a and j. Summing over the discarded entries gives

$$\mathcal{L}(\mathcal{S}) = \max_a \ell_a(\mathcal{S}) \le \mathcal{U}(\mathcal{S}) \le \sum_{a \in \mathcal{I}} \ell_a(\mathcal{S}) \le |\mathcal{I}|\, \mathcal{L}(\mathcal{S}). \tag{29}$$

To establish minimality, consider fixed nonnegative coefficients $\{b_j\}$ satisfying

$$\sum_{j \in \mathcal{H}_t \setminus \mathcal{S}} b_j \ge \mathcal{L}(\mathcal{S}) \quad \text{for every } \mathcal{A}_t \subseteq \mathcal{S} \subseteq \mathcal{H}_t. \tag{30}$$

For any optional entry j, take $\mathcal{S} = \mathcal{H}_t \setminus \{j\}$. The bound then requires $b_j \ge \max_a w_{aj} = s_j$. Conversely, $b_j \ge s_j$ for every optional entry is sufficient by Eq. 29. Thus, these coefficient-wise conditions are necessary and sufficient, proving that $\mathcal{U}$ is the pointwise smallest zero-intercept additive bound valid for every deletion of optional entries. □

The universal requirement is important: it covers all deletion sets, not only the subsets satisfying one particular budget or mask-coverage condition. The singleton argument concerns discarded masses, which remain defined even if removing that entry leaves some query without a usable context. The theorem does not identify the tightest possible bound among nonadditive functions or bounds allowed to include an extra constant.

Proof of Theorem 4.3. The count-based feasible family in the main text is

$$\mathcal{F}_t = \{\mathcal{S} \subseteq \mathcal{H}_t : \mathcal{A}_t \subseteq \mathcal{S},\ |\mathcal{S} \setminus \mathcal{A}_t| \le k_t\}. \tag{31}$$

For any $\mathcal{S} \in \mathcal{F}_t$, mandatory entries are never discarded, so

$$\mathcal{U}(\mathcal{S}) = \sum_{j \in \mathcal{H}_t \setminus \mathcal{A}_t} s_j - \sum_{j \in \mathcal{S} \setminus \mathcal{A}_t} s_j. \tag{32}$$

The first sum is constant. Since the scores are nonnegative and optional entries have equal costs, retaining the k_t highest-scoring optional entries maximizes the second sum and therefore minimizes $\mathcal{U}$. If fewer optional entries are available, all are retained. Ties do not change the objective value. Let $\mathcal{S}^\star$ minimize $\mathcal{L}$ over the finite family $\mathcal{F}_t$. Theorem 4.2 and the established optimality imply

$$\mathcal{L}(\mathcal{S}_t) \le \mathcal{U}(\mathcal{S}_t) \le \mathcal{U}(\mathcal{S}^\star) \le |\mathcal{I}|\, \mathcal{L}(\mathcal{S}^\star) = |\mathcal{I}| \min_{\mathcal{S} \in \mathcal{F}_t} \mathcal{L}(\mathcal{S}). \tag{33}$$

This proves both statements. □

What is optimized, and what is relaxed. If the returned set is usable, Theorems 4.1 and 4.2 also imply

$$\mathcal{E}(\mathcal{S}_t; \mathbf{V}_t) \le D_{\mathrm{V}} \mathcal{L}(\mathcal{S}_t) \le D_{\mathrm{V}} \mathcal{U}(\mathcal{S}_t) = D_{\mathrm{V}} \min_{\mathcal{S} \in \mathcal{F}_t} \mathcal{U}(\mathcal{S}). \tag{34}$$

The exact optimization guarantee concerns $\mathcal{U}$, not actual output error or exact minimization of $\mathcal{L}$. For one group, $\mathcal{U} = \mathcal{L}$; with multiple groups, the relaxation can be conservative. Equal-cost, count-based feasibility is essential: arbitrary subset-dependent packing costs or additional mask-coverage constraints need separate treatment. If the selected top-k set is itself usable, its upper-bound optimality also holds against the usable sets in $\mathcal{F}_t$, since these form a subset of the comparison family. The guarantees apply within a layer/KV-head unit under its assigned budget; they do not optimize the allocation across units. The value diameter is used only for analysis and need not be computed during retention.

D.3 Why Keep Semantic Groups Separate?

Within-group averaging makes the score invariant to uniformly replicating all observations in any one group. Taking a maximum makes it invariant to adding a duplicate group with the same mean attention profile. Both statements follow directly from Eq. 14; arbitrary new observations need not leave the score unchanged. In contrast, pooling the observations counted with their group memberships produces

$$\begin{aligned} s_j^{\text{pool}} &= \sum_{a\in\mathcal{I}} \pi_a w_{aj}, \qquad \pi_a = \frac{|\mathcal{P}_a|}{\sum_{b\in\mathcal{I}} |\mathcal{P}_b|}, \\ \sum_{j\in\mathcal{H}_t\setminus\mathcal{S}} s_j^{\text{pool}} &= \sum_{a\in\mathcal{I}} \pi_a \ell_a(\mathcal{S}). \end{aligned} \tag{35}$$

Thus, pooling controls a query-count-weighted average rather than a bound applying separately to every group. An entry receiving weight w_{aj} only from group a receives pooled score $\pi_a w_{aj}$, whereas its mean–maximum score remains w_{aj}. Even with equal group sizes, this attenuation is a factor of $|\mathcal{I}|$. More generally, with $\pi_{\min} = \min_a \pi_a > 0$, the pooled discarded mass can upper-bound $\mathcal{L}$ only after a worst-case rescaling by $1/\pi_{\min}$; this factor is necessary when the discarded mass is concentrated in the least-weighted group. These observations explain the chosen aggregation, but do not imply that its retained set dominates pooled selection on every input.

Relation to prior work and practical meaning. Attention-output bounds and top-k bound minimization have precedents in cache eviction (Feng et al., 2025a). Mean–maximum aggregation across attention heads is also described in the OmniKV discussion of Hao et al. (2026). Here, the groups come from planner-extracted semantic constraints, and the analysis identifies the smallest universal additive bound for their individual discarded masses. Unlike the chunk-wise denoising analysis of Light Forcing (Lv et al., 2026), the result requires neither a diffusion convergence rate nor a model of how sparsity changes score-estimation error. It is a local guarantee for the observed groups and unchanged KV entries. Whether cross-attention selects semantically informative queries, whether those observations represent later queries, and whether the retained information improves video quality remain empirical questions. No error tolerance, extra optimization term, or output-reconstruction module is added to SemanticKV.

E Detailed Benchmark Results

This section collects detailed numerical comparisons. Table 3 retains all 16 VBench dimensions, throughput, speedup, and chunk discontinuity across backbones. The main-paper backbone table is a direct subset, with a common semantic diagnostic panel. For the compression-ratio comparison, Table 4 reports the six dimensions in Figure 3, together with throughput and speedup. Table 5 compares SemanticKV with Full KV for 30- and 60-second generation; Table 6 reports the 100-prompt preference evaluation.

E.1 VBench across Backbones

Table 3 expands the Self Forcing and Causal Forcing comparison in Table 1. Panel A reports efficiency and visual/temporal quality, and Panel B provides all nine semantic dimensions, including the six content diagnostics displayed in the main table.

E.2 VBench across Compression Ratios

Table 4 gives the numerical results underlying Figure 3 for compression ratios 0.5, 0.7, and 0.9 on LongLive model. It retains Subject Consistency, Background Consistency, Temporal Flickering, Human Action, Color, and Object Class, together with DiT-only throughput and speedup. Full KV is shared only within this run group. The labels denote compression ratios, i.e., target fractions of the cache removed, rather than measured per-update byte reductions or peak-memory reductions.

Table 3: Efficiency and generation quality on VBench for 5-second video generation at compression ratio 0.7. Full KV retains the native cache without additional eviction and serves as the reference. Speedup is measured relative to Full KV. VBench metrics retain their reported percentage-scaled values. ↑ indicates higher is better and ↓ indicates lower is better. **Bold** marks the best result among compressed methods for each backbone.

Panel A: Efficiency, Temporal Quality, and Overall Video Quality

Backbone	Length	Method	Efficiency		KV Cache		Temporal Quality				VBench Quality				
			FPS↑	Speedup↑	Memory (GB)↓	Reduction (×)↑	Chunk Disc.↓	Dynamic Degree↑	Temporal Flicker.↑	Motion Smooth.↑	Total Score↑	Imaging Quality↑	Subject Cons.↑	Background Cons.↑	Aesthetic Quality↑
Self Forcing	5 s	Full KV	6.95	1.00×	6.04	1.00×	2.25	91.67	98.54	97.77	82.96	67.50	93.84	95.24	61.07
		R-KV	7.36	1.06×	1.81	3.35×	**2.23**	92.78	98.45	97.77	82.90	67.52	**93.54**	95.01	**61.26**
		Dummy Forcing	**8.16**	**1.17**×	1.81	3.35×	**2.23**	91.39	98.42	**97.81**	82.87	67.49	93.49	**95.22**	61.09
		Forcing-KV	7.92	1.14×	1.81	3.35×	2.62	92.78	98.41	97.55	82.69	67.06	93.28	95.14	60.95
		SEMANTICKV	7.95	1.14×	1.81	3.35×	2.26	**93.33**	**98.50**	97.80	**83.03**	**67.56**	93.49	95.10	60.99
Causal Forcing	5 s	Full KV	7.24	1.00×	6.04	1.00×	2.39	83.61	98.44	97.87	83.99	71.12	96.62	97.07	64.03
		R-KV	8.14	1.13×	1.81	3.35×	2.54	86.39	98.32	97.72	84.00	71.17	96.32	96.81	64.24
		Dummy Forcing	**9.20**	**1.27**×	1.81	3.35×	2.53	86.39	**98.35**	**97.78**	84.05	**71.21**	**96.42**	96.87	**64.33**
		Forcing-KV	8.91	1.23×	1.81	3.35×	2.97	**90.00**	98.23	97.39	84.04	70.89	96.14	96.96	64.03
		SEMANTICKV	8.83	1.22×	1.81	3.35×	**2.52**	86.67	98.34	97.72	**84.10**	71.13	96.33	**96.97**	64.13

Panel B: Semantic Fidelity

Backbone	Length	Method	Object Class↑	Multiple Objects↑	Human Action↑	Color↑	Spatial Relation↑	Scene↑	Temporal Style↑	Appearance Style↑	Overall Cons.↑
Self Forcing	5 s	Full KV	89.83	75.55	92.00	90.54	71.67	48.05	24.43	19.85	26.31
		R-KV	88.29	74.15	**93.00**	**91.40**	71.66	47.53	24.35	19.85	26.30
		Dummy Forcing	89.26	**77.07**	92.00	91.12	71.01	47.43	24.38	19.85	**26.40**
		Forcing-KV	88.21	75.88	92.60	90.64	71.44	47.72	24.42	19.76	26.39
		SEMANTICKV	**89.43**	75.47	92.00	91.30	**72.46**	**48.65**	**24.45**	**19.92**	26.37
Causal Forcing	5 s	Full KV	91.11	75.88	90.00	88.72	77.57	46.25	24.51	20.05	25.54
		R-KV	**92.88**	76.77	91.40	87.89	75.60	45.20	24.48	19.98	25.55
		Dummy Forcing	91.72	76.34	**91.60**	88.12	77.57	43.88	24.52	**20.07**	25.30
		Forcing-KV	92.74	**79.19**	89.80	**89.27**	**78.18**	43.18	**24.53**	19.97	**25.60**
		SEMANTICKV	91.85	77.09	**91.60**	89.09	78.01	**45.35**	24.40	20.04	25.56

Table 4: VBench results across compression ratios for 5-second generation, reporting the six dimensions in Figure 3 and DiT-only efficiency. Full KV is the shared uncompressed reference within this run group. Speedup is measured relative to Full KV. VBench metrics retain their reported percentage-scaled values. ↑ indicates higher is better. **Bold** marks the best result among compressed methods at each compression ratio.

Compression Ratio	Length	Method	Efficiency		KV Cache		VBench Quality					
			FPS↑	Speedup↑	Memory (GB)↓	Reduction (×)↑	Temporal Flicker.↑	Subject Cons.↑	Background Cons.↑	Object Class↑	Human Action↑	Color↑
		Full KV	6.57	1.00×	3.45	1.00×	99.26	96.06	96.68	90.84	92.60	90.34
0.5	5 s	R-KV	7.53	1.15×	1.72	2.01×	**99.32**	**96.07**	96.61	91.01	93.00	**90.84**
		Dummy Forcing	7.32	1.11×	1.72	2.01×	98.81	95.61	95.95	90.09	93.60	88.95
		Forcing-KV	7.67	1.17×	1.72	2.01×	99.29	95.92	**96.63**	90.84	**94.00**	89.73
		SEMANTICKV	**8.00**	**1.22**×	1.72	2.01×	99.29	96.04	96.52	**91.93**	92.80	89.24
0.7	5 s	R-KV	8.47	1.29×	1.02	3.38×	99.24	95.88	96.37	91.01	92.80	88.10
		Dummy Forcing	7.82	1.19×	1.02	3.38×	99.01	95.35	94.50	**91.57**	92.40	87.88
		Forcing-KV	8.67	**1.32**×	1.02	3.38×	99.26	95.81	**96.41**	90.11	**93.60**	88.61
		SEMANTICKV	**8.70**	**1.32**×	1.02	3.38×	**99.27**	**95.97**	**96.41**	91.27	93.00	**89.29**
0.9	5 s	R-KV	**10.04**	**1.53**×	0.34	10.12×	99.02	94.03	93.70	89.73	93.40	**90.87**
		Dummy Forcing	8.84	1.34×	0.34	10.12×	**99.04**	**94.48**	**94.17**	88.78	92.60	90.15
		Forcing-KV	9.95	1.51×	0.34	10.12×	98.79	93.70	92.99	87.91	93.80	90.29
		SEMANTICKV	10.00	1.52×	0.34	10.12×	99.02	94.07	93.70	**91.60**	**94.60**	90.69

Table 5: Comparison of SEMANTICKV with Full KV on VBenchLong for 30- and 60-second video generation. Full KV retains the native cache without additional eviction and serves as the reference. Speedup is measured relative to Full KV. ↑ indicates higher is better and ↓ indicates lower is better. The comparison measures the quality–throughput trade-off relative to the uncompressed reference; no ranking among compression methods is reported.

Backbone	Length	Method	Efficiency		KV Cache		Temporal Quality		VBenchLong Quality					
			FPS↑	Speedup↑	Memory (GB)↓	Reduction (×)↑	Chunk Disc.↓	Dynamic Degree↑	Total Score↑	Imaging Quality↑	Subject Cons.↑	Background Cons.↑	Motion Smooth.↑	Aesthetic Quality↑
LongLive 2.0	30 s	Full KV	4.86	1.00×	20.76	1.00×	3.41	58.75	80.44	66.89	97.79	97.35	98.87	55.35
		SEMANTICKV	6.16	1.27×	6.23	3.33×	3.31	60.40	79.66	64.40	97.39	96.89	98.75	54.22
LongLive 2.0	60 s	Full KV	4.76	1.00×	20.76	1.00×	3.39	62.21	80.12	65.82	97.66	97.13	98.75	53.77
		SEMANTICKV	6.12	1.29×	6.23	3.33×	3.39	61.15	78.84	62.72	97.20	96.78	98.62	51.88

E.3 LONG-HORIZON EVALUATION AGAINST FULL KV

Table 5 compares SEMANTICKV with Full KV on LongLive 2.0 for 30- and 60-second VBench-Long generation. This reference comparison measures the quality–throughput trade-off relative to uncompressed generation, with the same displayed metrics at both durations.

Thirty-second generation. SEMANTICKV increases DiT-only throughput from 4.86 to 6.16 FPS (1.27×), while Total Score decreases from 80.44 to 79.66 (0.78 points). Subject and Background Consistency decrease by 0.40 and 0.46 points, respectively. Chunk discontinuity decreases from 3.41 to 3.31, and Dynamic Degree increases from 58.75 to 60.40. Imaging and Aesthetic Quality decrease from 66.89 and 55.35 to 64.40 and 54.22.

Sixty-second generation. SEMANTICKV increases throughput from 4.76 to 6.12 FPS (1.29×), with Total Score decreasing from 80.12 to 78.84 (1.28 points). Subject and Background Consistency decrease by 0.46 and 0.35 points, while chunk discontinuity remains 3.39. Imaging and Aesthetic Quality decrease by 3.10 and 1.89 points. The Total Score reduction relative to Full KV is larger at 60 seconds than at 30 seconds.

Scope of the evidence. These results characterize the quality–throughput trade-off relative to Full KV, not a ranking among compression methods. The local attention-output bounds concern observed query groups at a single cache update and do not guarantee unchanged video quality over long autoregressive rollouts.

E.4 VISIONREWARD RESULTS

Table 6 reports overall VisionReward and instruction-following measurements on LongLive at compression ratio 0.9. All methods are scored with the same VisionReward model and evaluator runtime on the separate 100-prompt set; these scores are not combined numerically with VBench or with online critic utilities. SEMANTICKV obtains 98.000 instruction following, matching R-KV and Forcing-KV and exceeding Full KV and Dummy Forcing by two points. Its 7.964 overall reward exceeds R-KV by 0.458 and Forcing-KV by 0.181, but trails Dummy Forcing by 0.497. Thus, the method improves instruction following relative to some baselines without uniformly maximizing the broader reward aggregate.

Table 6: VisionReward evaluation on LongLive at compression ratio 0.9. We use the 100 prompts from Causal Forcing and score all methods with the same evaluator runtime. Higher is better for both metrics.

Method	VisionReward↑	Instruction Following↑
Full KV	10.264	96.000
R-KV	7.506	98.000
Dummy Forcing	8.461	96.000
Forcing-KV	7.783	98.000
SEMANTICKV	7.964	98.000

F Detailed Ablation Results

The ablations use 5-second VBench generation with LongLive at compression ratio 0.9. Prompts, seeds, cache budget, protected native states, generation schedule, and evaluation protocol are unchanged. The planner and complete agentic workflow remain active in every variant.

F.1 Semantic Query Selection and Aggregation

With $n_q = 32$, *Uniform queries* replaces constraint-conditioned selection with the same number of uniformly spaced observations, preserving group sizes and maximum aggregation. *Pooled constraints* keeps the selected queries but replaces the groupwise maximum in Eq. 14 with pooled-mean attention. Each control changes only one component of cache scoring. We focus on whether these components improve semantic fidelity under the same cache budget. Table 7 first reports the official VBench Semantic Score and its nine constituent dimensions, followed by efficiency, Quality Score, and temporal diagnostics.

Table 7: Component ablations on LongLive and 5-second VBench at compression ratio 0.9 with $n_q = 32$. Panel A reports semantic fidelity; Panel B reports DiT-only efficiency and visual/temporal diagnostics. Semantic and Quality are the official VBench aggregate scores. VBench scores retain their reported percentage-scaled values. ↑ indicates higher is better and ↓ indicates lower is better. **Bold** marks the best displayed value among the three configurations.

Panel A: Semantic Score and Semantic Fidelity

Configuration	Semantic Score↑	Object Class↑	Multiple Objects↑	Human Action↑	Color↑	Spatial Relation↑	Scene↑	Temporal Style↑	Appearance Style↑	Overall Cons.↑
Uniform queries	76.94	90.44	**76.02**	**94.60**	90.26	72.12	44.36	25.05	20.61	**27.00**
Pooled constraints	76.78	91.00	75.90	93.80	89.40	73.05	43.71	25.04	20.52	26.98
SemanticKV	**77.18**	**91.60**	75.34	**94.60**	**90.69**	**73.27**	**44.46**	**25.09**	**20.66**	26.88

Panel B: Efficiency and Visual/Temporal Diagnostics

Configuration	Efficiency		KV Cache		Visual/Temporal Diagnostics				
	FPS↑	Speedup↑	Memory (GB)↓	Reduction (×)↑	Quality Score↑	Chunk Disc.↓	Dynamic Degree↑	Temporal Flicker.↑	Motion Smooth.↑
Uniform queries	9.98	**1.52**×	0.34	10.12×	**82.93**	**3.45**	**62.50**	**99.02**	98.55
Pooled constraints	9.94	1.51×	0.34	10.12×	82.64	3.52	59.44	99.01	98.56
SemanticKV	**10.00**	**1.52**×	0.34	10.12×	82.80	3.54	61.39	**99.02**	**98.58**

Semantic effects. Constraint-conditioned selection improves the Semantic Score over uniform queries by 0.24 points, with gains of 1.16 in Object Class and 1.15 in Spatial Relation; Human Action is tied at 94.60. Relative to pooled constraints, the full method improves the Semantic Score by 0.40 points, including gains of 0.60 in Object Class, 1.29 in Color, and 0.75 in Scene. These comparisons support the semantic contributions of query selection and groupwise maximum aggregation under the fixed cache budget.

Efficiency and scope. Throughput ranges from 9.94 to 10.00 FPS, so the measurements are similar across configurations rather than evidence of a substantial speed advantage. The full method does not dominate every metric: uniform queries achieve a higher Quality Score and lower chunk discontinuity, while both controls score higher on Multiple Objects and Overall Consistency. We interpret the results as a semantic-alignment benefit, not uniformly better visual or temporal quality, and do not attach statistical significance to the small score differences.

F.2 Semantic Query Limit

The query limit n_q bounds the number of selected self-attention queries per semantic constraint; it does not change the target cache budget. We compare $n_q \in \{16, 32, 64\}$ with all other settings fixed. Table 8 retains the complete results.

Table 8: Ablation of the semantic query limit n_q on VBench. The compression ratio is fixed to 0.9. VBench metrics retain their percentage-scaled values, except Chunk Discontinuity, which is the RAFT TopK-to-mean optical-flow ratio. ↑ indicates higher is better and ↓ indicates lower is better. The best displayed result in each column is highlighted in **bold**.

Panel A: Efficiency, Temporal Quality, and Overall Video Quality

n_q	Efficiency		KV Cache		Temporal Quality				VBench Quality				
	FPS↑	Speedup↑	Memory (GB)↓	Reduction (×)↑	Chunk Disc.↓	Dynamic Degree↑	Temporal Flicker.↑	Motion Smooth.↑	Total Score↑	Imaging Quality↑	Subject Cons.↑	Background Cons.↑	Aesthetic Quality↑
16	9.93	1.51×	0.34	10.12×	**3.54**	60.00	**99.02**	98.57	81.54	67.35	94.00	93.68	**62.45**
32	**10.00**	**1.52**×	0.34	10.12×	**3.54**	**61.39**	**99.02**	**98.58**	**81.68**	**67.36**	**94.07**	93.70	62.30
64	9.78	1.49×	0.34	10.12×	3.55	58.33	**99.02**	98.57	81.45	67.27	93.95	**93.81**	62.34

Panel B: Semantic Fidelity

n_q	Object Class↑	Multiple Objects↑	Human Action↑	Color↑	Spatial Relation↑	Scene↑	Temporal Style↑	Appearance Style↑	Overall Cons.↑
16	90.68	75.55	93.80	**90.80**	**73.28**	43.49	25.02	**20.73**	26.98
32	**91.60**	75.34	**94.60**	90.69	73.27	44.46	**25.09**	20.66	26.88
64	90.38	**76.25**	94.00	90.26	73.19	**44.49**	25.00	20.69	**27.01**

Results. The default $n_q = 32$ achieves the highest Total Score (81.68), improving over $n_q = 16$ and $n_q = 64$ by 0.14 and 0.23 points. It also gives the highest Object Class (91.60), Human Action (94.60), and Dynamic Degree (61.39) scores. Temporal Flickering is identical at 99.02, Motion Smoothness varies by only 0.01 points, and chunk discontinuity ranges from 3.54 to 3.55. Other dimensions favor different query limits: $n_q = 64$ improves Multiple Objects, Background Consistency, Scene, and Overall Consistency, whereas $n_q = 16$ improves Aesthetic Quality, Color, Spatial Relation, and Appearance Style.

Efficiency and default. The measured throughputs are 9.93, 10.00, and 9.78 FPS for $n_q = 16$, 32, and 64, respectively. Their small differences do not isolate the cost of query selection and may include system-level variation. We use $n_q = 32$ because it provides the highest measured aggregate quality without a throughput penalty in this comparison, not because it is optimal for every semantic dimension.

G Additional Semantic and Temporal Analyses

These analyses reuse existing VBench videos, evaluator outputs, and fixed planner decompositions. They assess the distribution and temporal behavior of the reported semantic differences without introducing additional generation experiments. Uncertainty is estimated over prompts; seeds from the same prompt are kept together when multiple seeds are available.

G.1 Paired Sample-Level Reliability

We pair each available prompt–seed tuple between SemanticKV and a comparator for six semantic dimensions. Each dimension–comparator pair contains 395–500 videos from 79–100 unique prompts. We average the five seed-level differences within each prompt and resample prompts 10,000 times for cluster-bootstrap 95% confidence intervals. Figure 4 reports both the mean differences and the tie-aware superiority rate $P(\Delta > 0) + \frac{1}{2}P(\Delta = 0)$, where Δ is the paired score difference. Table 9 lists the differences; its macro column averages the six dimensions without weighting and is not the official VBench Semantic Score. The reported intervals are marginal, not simultaneous multiple-comparison guarantees.

Paired semantic analysis. On matched prompt–seed outputs (Figure 4), the macro-average differences across six semantic dimensions favor SemanticKV over R-KV, Dummy Forcing, and Forcing-KV by 1.00, 0.86, and 1.25 points. Individual dimensions remain mixed, and most confidence intervals include zero. The following subsections examine semantic complexity and temporal drift; Appendix H provides qualitative examples.

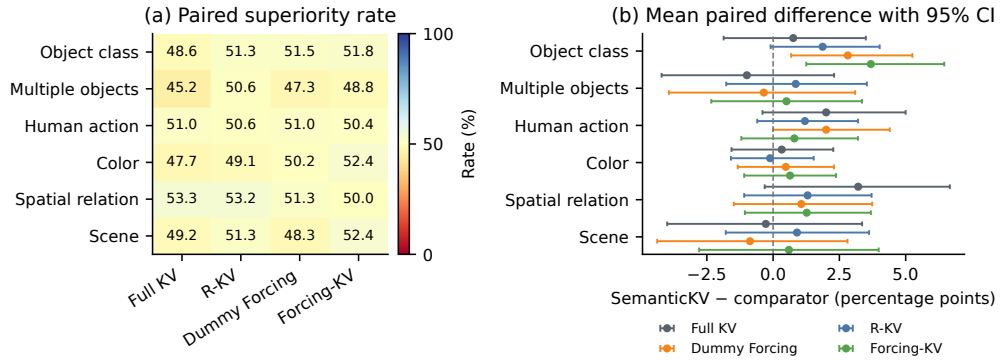


Figure 4: Paired semantic fidelity on matched prompt–seed tuples. (a) Tie-aware superiority rates. (b) Mean paired differences with prompt-clustered bootstrap 95% confidence intervals. Positive differences favor SEMANTICKV; Full KV is the uncompressed reference.

Table 9: Mean paired semantic-score difference (SEMANTICKV minus comparator), in percentage points. The macro column averages the six dimensions without reweighting them.

Comparator	Object	Multiple objects	Human action	Color	Spatial relation	Scene	Macro
Full KV	+0.76	−0.99	+2.00	+0.32	+3.21	−0.28	+0.84
R-KV	+1.87	+0.85	+1.20	−0.11	+1.30	+0.90	+1.00
Dummy Forcing	+2.82	−0.35	+2.00	+0.48	+1.06	−0.87	+0.86
Forcing-KV	+3.69	+0.50	+0.80	+0.64	+1.27	+0.60	+1.25

The macro differences are positive against all three compressed baselines. Of the 18 dimension–baseline comparisons, 15 favor SEMANTICKV, including all six comparisons with Forcing-KV. Object Class improves by 2.82 points over Dummy Forcing and 3.69 over Forcing-KV, with intervals excluding zero. The exceptions are Multiple Objects and Scene against Dummy Forcing (-0.35 and -0.87 points), and Color against R-KV (-0.11). Most remaining dimension-level intervals include zero. The evidence therefore supports a positive macro trend without establishing significance for every dimension.

G.2 PERFORMANCE VERSUS SEMANTIC COMPLEXITY

The 946 benchmark entries contain 944 distinct prompt strings, for which the planner outputs fixed decompositions. Among prompts evaluated on the six semantic dimensions, we group them by one, two, or at least three semantic constraints, yielding 126, 75, and 315 unique prompts. Grouping depends only on the planner output, not on evaluation scores. Within each group, we compute paired prompt-level differences per dimension and macro-average the six dimensions; uncertainty uses the same 10,000-sample prompt-clustered bootstrap protocol as above. The "atoms" shown in Figure 5 refer to these semantic constraints.

For one-constraint prompts, the macro differences against R-KV, Dummy Forcing, and Forcing-KV are $+0.68$, -0.02, and $+0.58$ points. They rise to $+4.62$, $+3.26$, and $+7.24$ for two constraints, and remain positive at $+1.89$, $+2.82$, and $+2.23$ for at least three constraints. All two-constraint intervals exclude zero. In the largest group, the interval excludes zero against Dummy Forcing but narrowly crosses zero against R-KV and Forcing-KV. Relative to Full KV, the corresponding differences are -0.75, $+0.42$, and $+3.83$, with the last interval $[+1.17, +6.98]$. The pattern suggests a benefit for compositional prompts, but is not monotonic in constraint count.

G.3 WITHIN-VIDEO TEMPORAL SEMANTIC DRIFT

This diagnostic reuses existing videos rather than generating new samples. We select the first 24 lexicographically sorted prompts from each of Object Class, Multiple Objects, Human Action, and Scene for which all five methods are available, using seed 42. The fixed 96-prompt subset contains 480 videos. Five timestamps—0%, 25%, 50%, 75%, and 100% of normalized duration—give 2,400 evaluated frames. CLIP ViT-B/32 (Radford et al., 2021) measures image–prompt alignment and

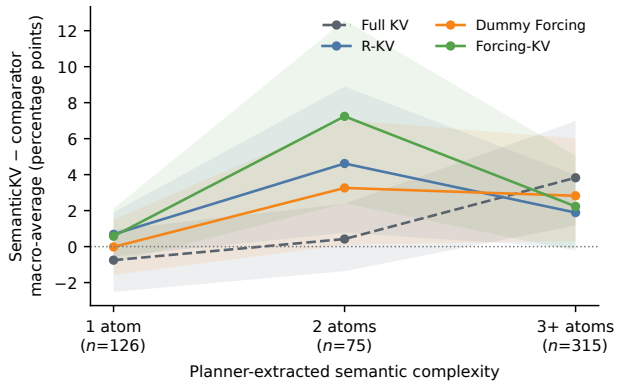


Figure 5: **Semantic fidelity as a function of planner-extracted prompt complexity.** Values are macro-averaged paired differences between SEMANTICKV and each comparator across six semantic VBench dimensions. Shaded regions show prompt-clustered bootstrap 95% confidence intervals. The grouping uses the planner decomposition only and is fixed before inspecting the video scores.

visual-feature similarity to the first frame. Alignment is reported as a change from the first frame to reduce prompt-specific offsets; confidence intervals are obtained by resampling prompts.

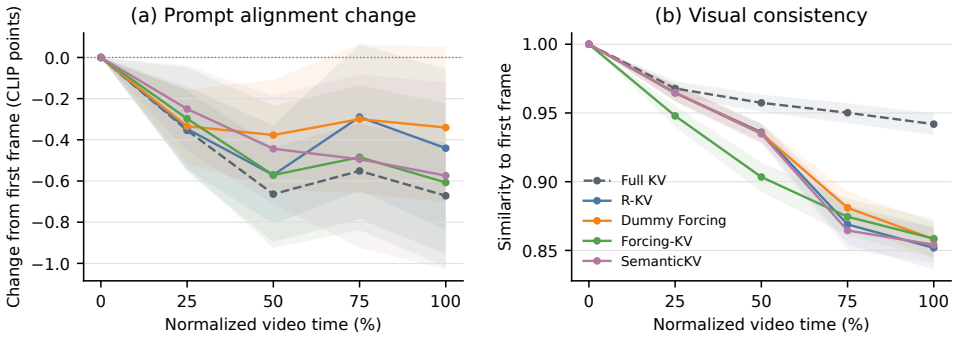


Figure 6: **Within-video drift on a fixed 96-prompt subset.** (a) Change in CLIP image–prompt alignment relative to the first frame. (b) CLIP visual-feature similarity to the first frame. Curves show means and shaded prompt-bootstrap 95% confidence intervals over 96 videos per method. Full KV is the uncompressed reference.

Table 10: End-of-video temporal analysis. Brackets give prompt-bootstrap 95% confidence intervals. A less negative alignment change and a higher visual similarity indicate less drift.

Method	Alignment change (CLIP points)	Visual similarity to first frame
Full KV	-0.67 $[-0.95, -0.41]$	0.942 $[0.934, 0.950]$
R-KV	-0.44 $[-0.84, -0.06]$	0.852 $[0.836, 0.867]$
Dummy Forcing	-0.34 $[-0.70, +0.05]$	0.858 $[0.845, 0.870]$
Forcing-KV	-0.61 $[-1.02, -0.22]$	0.859 $[0.844, 0.873]$
SEMANTICKV	-0.57 $[-1.03, -0.12]$	0.854 $[0.841, 0.867]$

All methods show a modest decline in prompt alignment. At the final frame, the paired alignment differences between SEMANTICKV and R-KV, Dummy Forcing, and Forcing-KV are -0.13, -0.23, and $+0.03$ CLIP points; all paired intervals include zero. Paired visual-similarity differences are at most 0.0044, again with intervals including zero. Full KV has higher visual similarity than every compressed method: SEMANTICKV is 0.088 lower, with an interval excluding zero. Because valid object or camera motion can also reduce feature similarity, this is a drift diagnostic rather than a standalone quality ranking. We find no detectable difference between the compressed methods on this subset, which should not be interpreted as proof of temporal equivalence.

H QUALITATIVE COMPARISONS

Figures 7–12 show diagnostic LongLive examples covering actions, object classes, multiple objects, and scenes. Methods are matched by original prompt and seed, and columns use the same uniformly spaced timestamps. The score beside each row is the official per-video result for the semantic dimension targeted by that prompt. Examples are selected where SEMANTICKV scores higher than every compressed baseline; they illustrate failure modes rather than estimate their frequency. The numerical comparisons in Table 4 and the paired analyses in Appendix G.1, including unfavorable dimensions, provide the quantitative context for these examples.

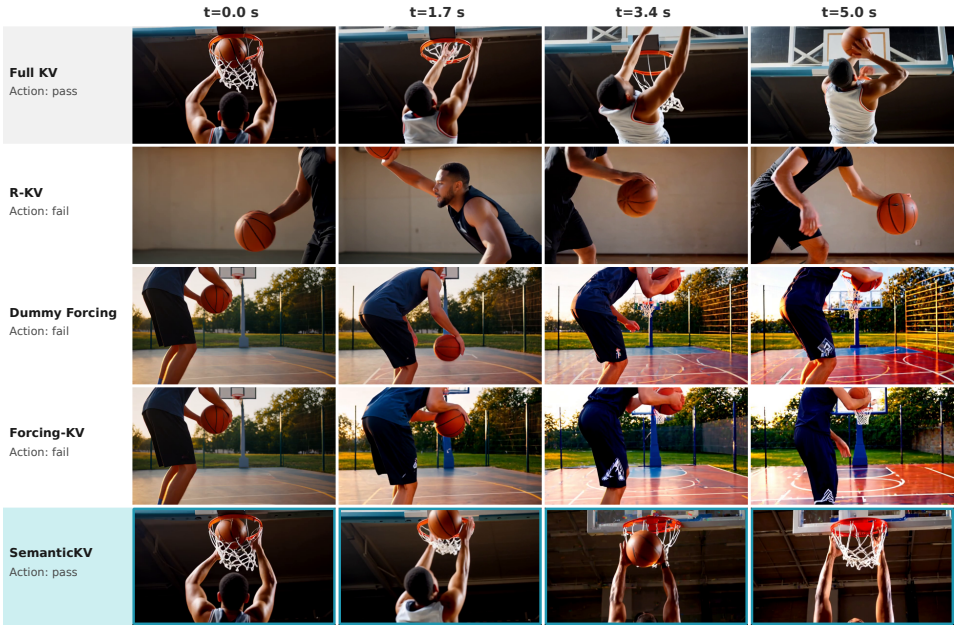


Figure 7: **Human-action fidelity on LongLive.** Prompt: "A person is dunking basketball" (seed 42). R-KV, Dummy Forcing, and Forcing-KV depict ball handling or dribbling and fail the VBench Human Action check. SEMANTICKV depicts the requested dunk and passes the check, matching the Full-KV reference. Rows compare methods and columns show uniformly sampled timestamps.

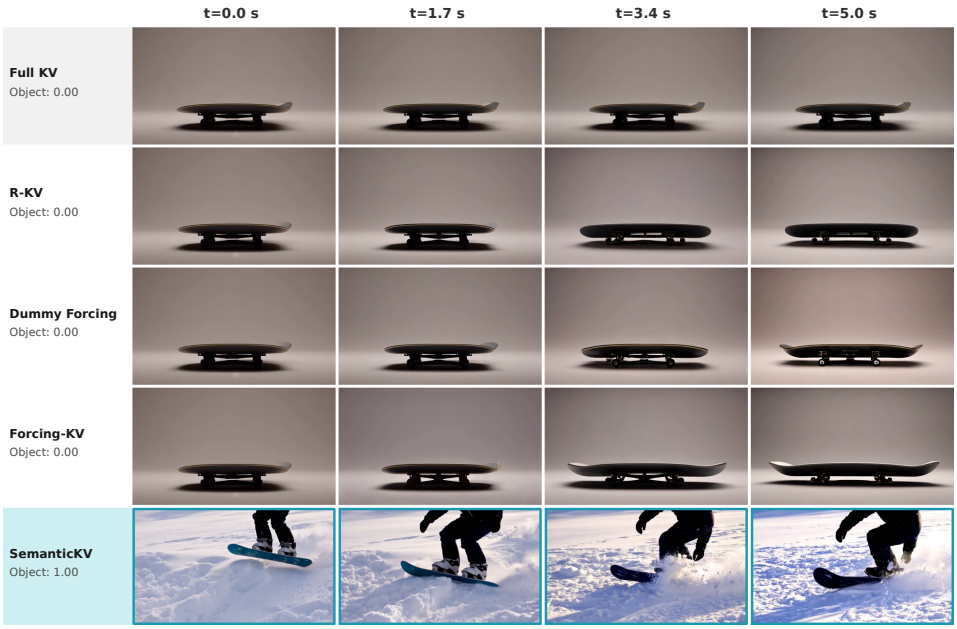


Figure 8: **Object-class fidelity on LongLive.** Prompt: "a snowboard" (seed 45). Full KV and all three compressed baselines generate a skateboard against a studio background and receive a per-video Object Class score of 0.00. SEMANTICKV generates a snowboard being ridden on snow and receives 1.00. The example illustrates that semantically guided retention can preserve the requested class even when a visually related concept is a strong failure mode.

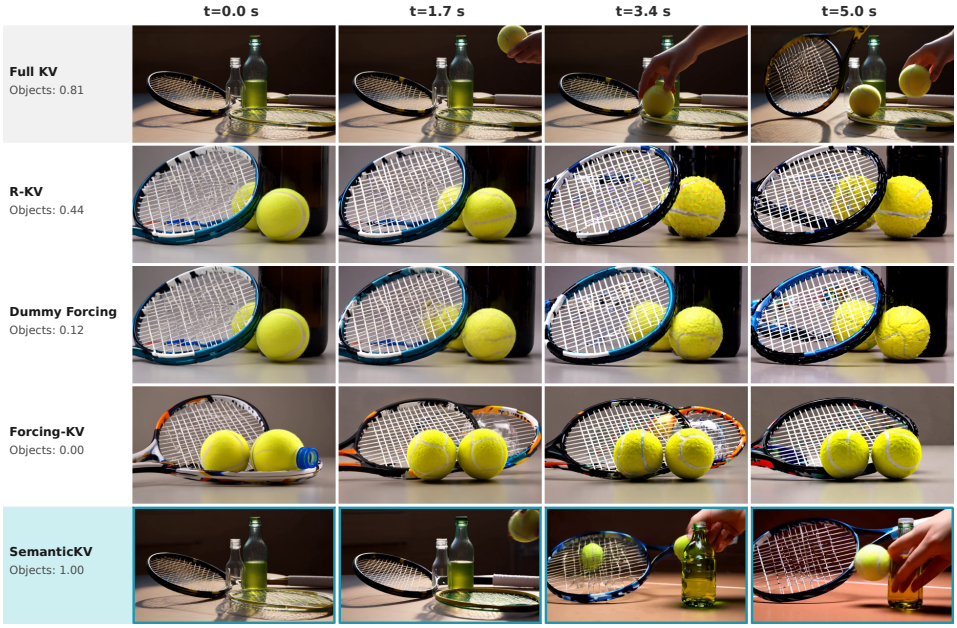


Figure 9: **Multi-object fidelity on LongLive.** Prompt: "a tennis racket and a bottle" (seed 43). The compressed baselines replace the requested bottle with tennis balls, producing per-video Multiple Objects scores of 0.44, 0.12, and 0.00 for R-KV, Dummy Forcing, and Forcing-KV, respectively. SEMANTICKV retains both requested objects and scores 1.00; Full KV scores 0.81.

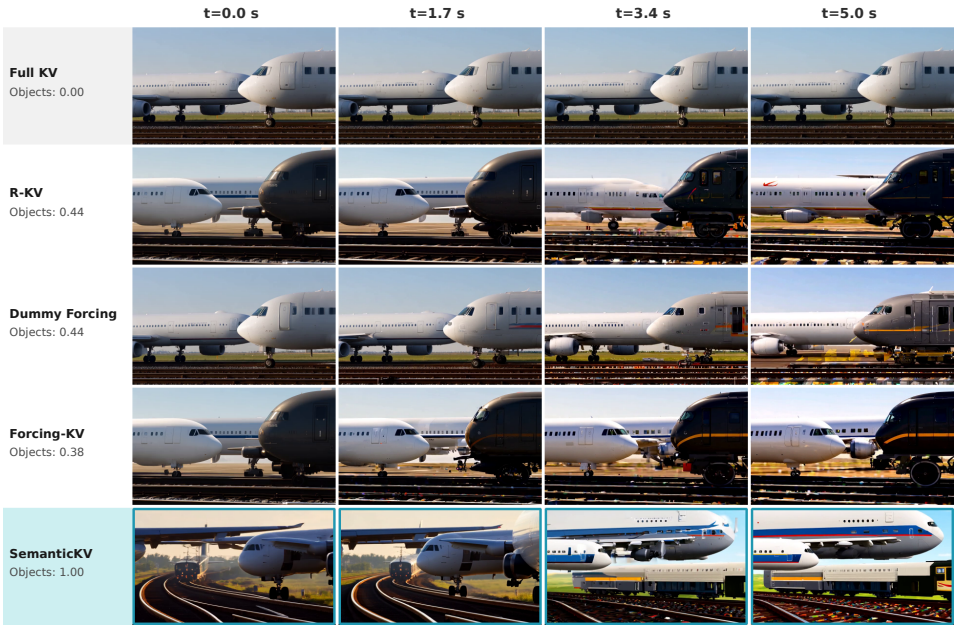


Figure 10: **Compositional object fidelity on LongLive.** Prompt: "an airplane and a train" (seed 44). The compressed baselines produce ambiguous airplane–train hybrids and obtain per-video Multiple Objects scores of 0.44, 0.44, and 0.38 for R-KV, Dummy Forcing, and Forcing-KV, respectively. SEMANTICKV keeps an airplane and a rail vehicle visually distinguishable and scores 1.00; Full KV scores 0.00.



Figure 11: **Scene fidelity on LongLive.** Prompt: "palace" (seed 45). Full KV and the three compressed baselines converge to a generic neoclassical institutional facade and receive per-video Scene scores of 0.00. SEMANTICKV produces palatial architecture across the sequence and scores 0.75.



Figure 12: **Activity-rich scene fidelity on LongLive.** Prompt: "construction site" (seed 43). SEMANTICKV preserves discriminative construction cues, including excavation, a worker, tools, and machinery, and obtains a per-video Scene score of 0.81. The corresponding scores are 0.00 for Full KV, 0.00 for R-KV, 0.06 for Dummy Forcing, and 0.00 for Forcing-KV.