EntityBench: Towards Entity-Consistent Long-Range Multi-Shot Video Generation

Ruozhen He[1,3] **Meng Wei**[1] **Ziyan Yang**[2] **Vicente Ordonez**[3]
[1]ByteDance [2]ByteDance Seed [3]Rice University
{catherine.he, vicenteor}@rice.edu
{weimeng.147, ziyan.yang}@bytedance.com

Abstract

Multi-shot video generation extends single-shot generation to coherent visual narratives, yet maintaining consistent characters, objects, and locations across shots remains a challenge over long sequences. Existing evaluations typically use independently generated prompt sets with limited entity coverage and simple consistency metrics, making standardized comparison across methods difficult. We introduce EntityBench, a benchmark consisting of 140 episodes (2,491 shots) derived from real narrative media, with explicit per-shot entity schedules tracking characters, objects, and locations simultaneously across easy, medium, and hard difficulty tiers of up to 50 shots, 13 cross-shot characters, 8 cross-shot locations, 22 cross-shot objects, and recurrence gaps spanning up to 48 shots. EntityBench pairs the dataset with a three-pillar evaluation framework that disentangles intra-shot visual quality, prompt-following alignment, and cross-shot entity consistency. Cross-shot consistency, the central pillar, evaluates each recurring entity through both embedding similarity and LLM per-criterion judgment across entity-type-specific dimensions, with a fidelity gate that admits accurate entity appearance. To establish baselines, we propose EntityMem, a memory-augmented generation system that plans and stores verified per-entity visual references in a persistent memory bank before generation begins, enabling the video backbone to retrieve the appearance of each entity across shots. Experiments on EntityBench show that cross-shot entity consistency degrades sharply with recurrence distance in existing methods, and that explicit per-entity memory yields the highest character fidelity (Cohen's d = +2.33) and presence among methods evaluated. Code and data are released at `https://github.com/Catherine-R-He/EntityBench`.

1 Introduction

Recent advances in video generation have enabled high-fidelity single-shot synthesis, and a growing body of work now extends these capabilities to multi-shot video generation, where a sequence of shots forms a coherent visual narrative [Guo et al., 2025, Meng et al., 2025b, Luo et al., 2026]. This progression opens new possibilities for automated storytelling, previsualization, and long-form content creation [Wang et al., 2025b, Guo et al., 2025].

Single-shot models focus on generating visually appealing clips with coherent motion and prompt adherence within a single scene [Gao et al., 2025, Wan et al., 2025]. Multi-shot generation, however, introduces an additional requirement that entities must maintain their visual identity across shots, not just within a single shot. This *entity consistency* requires awareness of how the same entity was rendered in previous shots, as even small appearance variations may accumulate over long sequences. The complexity grows further as realistic narratives involve multiple entity types simultaneously. Common entities include characters, objects, and locations, each with different consistency challenges and reappearance patterns across shots. Current methods address it implicitly through shared atten-

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Table 1: Comparison with existing video generation benchmarks. Char/Obj/Loc: number of annotated characters, objects, and locations. Entity Sched: per-shot entity-level schedule annotations. Transition: explicit cut/continuation labels. Intra/Inter-Shot: number of evaluation metrics for within-shot and cross-shot assessment. EntityBench provides large-scale multi-shot episodes with simultaneous tracking of 3 entity types, per-shot entity schedules, and a comprehensive evaluation suite.

Benchmark	Media-Source	Episodes	Max shots	Total shots	Multi-Shot	Entity Sched.	Char	Obj	Loc	Transition	Intra-Shot	Inter-Shot
VBench [Huang et al., 2024b]	✗	946	-	946	✗	✗	-	79	86	✗	16	0
OpenS2V-Nexus [Yuan et al., 2025a]	✗	240	-	240	✗	✗	3	64	14	✗	6	0
LongVGenBench [Gao et al., 2025]	✗	100	-	100	✗	✗	-	-	-	✗	7	0
MovieBench [Wu et al., 2025a]	✓	6	656	2,875	✓	✗	94	-	846	✗	0	1
VideoMemory [Zhou et al., 2026]	✓	54	12	648	✓	✗	54	54	54	✗	0	3
MSVBench [Shi et al., 2026]	✓	20	14	280	✓	✗	72	-	104	✗	10	10
NarrLV [Feng et al., 2025]	✗	14	6	280	✓	✓	-	570	347	✗	0	3
ST-Bench [Zhang et al., 2025]	✓	30	12	300	✓	✓	30	-	95	✓	3	3
EntityBench (ours)	✓	**140**	**50**	**2,491**	✓	✓	**987**	**2,077**	**654**	✓	**30**	**21**

tion [Meng et al., 2025b], reference conditioning [Zhang et al., 2025], or autoregressive context [Guo et al., 2025], but how well they actually preserve entity identity over long sequences remains difficult to assess without a standardized evaluation framework.

As shown in Table 1, existing benchmarks focus on single-shot quality, or provide limited multi-shot coverage with few episodes, short shot sequences, restricted entity types, no transition annotations, and narrow evaluation dimensions for intra- and inter-shot quality. This makes it difficult to systematically diagnose where and why entity consistency breaks down over long sequences.

We introduce EntityBench, a benchmark consisting of 140 episodes (2,491 shots) derived from real narrative media and enriched through LLM-based refinement. It spans easy, medium, and hard difficulty tiers with up to 50 shots organized into 1,146 scenes with explicit cut and continuation transitions, and recurrence gaps of up to 48 shots per episode. Each shot is annotated with an explicit entity schedule specifying which characters, objects, and locations should appear. For comprehensive analysis, we propose a three-pillar evaluation framework comprising 51 metrics: 6 intra-shot quality metrics, 24 prompt-following metrics, and 21 cross-shot consistency metrics combining embedding similarity with LLM judgments. A fidelity gate is used to ensure cross-shot consistency and is measured only on correctly rendered entities.

Using EntityBench, we explore entity-level memory management as a path to improving entity fidelity and cross-shot consistency. We propose EntityMem, a memory-augmented generation system that maintains a persistent per-entity memory bank, populated by VLM-based agents that generate, select, and verify entity visual and textual references before video generation begins. EntityMem enables the video backbone to retrieve entity information across shots while reducing error accumulation that arises from extracting references from generated outputs.

Our contributions are summarized as follows.

- We propose EntityBench, a multi-shot video generation benchmark with explicit per-shot entity schedules, simultaneous multi-entity tracking across characters, objects, and locations.
- We design a three-pillar evaluation framework that measures intra-shot quality, prompt-following alignment, and cross-shot entity consistency comprehensively.
- Through EntityMem, we show that entity memory management with quality-gated verification can help cross-shot fidelity and consistency.

2 Related Work

Benchmarks for Video Generation. Single-shot video generation quality has been extensively benchmarked. VBench [Huang et al., 2024b] established a standard evaluation across 16 dimensions, later extended to I2V and trustworthiness in VBench++ [Huang et al., 2025c] and to intrinsic faithfulness in VBench-2.0 [Zheng et al., 2025]. Other proposed single-shot benchmarks and metrics evaluate human-aligned multi-aspect quality [Liu et al., 2024b, Han et al., 2025, Koo et al., 2026], fine-grained text-video alignment [Liu et al., 2023], compositional generation [Sun et al., 2025], and video dynamics [Liao et al., 2024]. For identity and subject consistency, the IPVG Challenge

Table 2: Data curation funnel from raw clips to the final benchmark. Each stage shows the input count, output count, and retention rate.

Stage	Before	After	Retained
Quality filtering (clips)	100,000	45,589	46%
Content filtering (episodes)	831	606	73%
Window selection (shots)	55,142	2,491	5%

Table 3: Benchmark statistics across difficulty tiers. Cross-shot counts report entities appearing in 2+ shots. Recurrence gap measures the number of intervening shots between consecutive appearances of the same entity. Memory-test shots contain recurring entities without first-appearance descriptions.

	Easy	Medium	Hard	All
Episodes	80	40	20	140
Shots	873	618	1,000	2,491
Cross-shot characters	$5.1_{\pm 1.4}$	$6.0_{\pm 1.8}$	$8.9_{\pm 2.1}$	$5.9_{\pm 2.1}$
Cross-shot locations	$2.4_{\pm 0.8}$	$2.5_{\pm 0.7}$	$4.9_{\pm 1.7}$	$2.8_{\pm 1.3}$
Cross-shot objects	$3.9_{\pm 2.1}$	$6.1_{\pm 2.5}$	$13.3_{\pm 4.5}$	$5.9_{\pm 4.2}$
Mean recurrence gap	$2.1_{\pm 1.8}$	$2.2_{\pm 2.0}$	$3.4_{\pm 4.8}$	$2.7_{\pm 3.6}$
Max recurrence gap	$8.0_{\pm 2.0}$	$9.9_{\pm 2.2}$	$33.5_{\pm 7.8}$	$12.2_{\pm 9.4}$

released VIP-200K with 200K unique identities [Wang et al., 2025c] and OpenS2V-Nexus [Yuan et al., 2025a] provided a million-scale subject-to-video benchmark, but both evaluate only single-subject preservation within individual shots. LongVGenBench [Gao et al., 2025] evaluates controllability and consistency for minute-long single-scene videos but does not address multi-shot narratives.

Multi-shot Video Generation Evaluation. For multi-shot evaluation, MovieBench [Wu et al., 2025a] provides a hierarchical movie-level dataset with character banks, shot-level annotations, and evaluation tasks including character ID consistency measured via face recognition. VideoMemory [Zhou et al., 2026] introduces a 54-case benchmark structured as 3 entity subclasses $\times$ 3 shot lengths $\times$ 6 samples, evaluated at $K \in \{4, 8, 12\}$ shots with only 6 samples per condition. Each case isolates a single persistent entity type (character, property, or background) while deliberately varying the other two, which is an ablation protocol rather than a realistic narrative setting where multiple entity types must remain consistent simultaneously. Other multi-shot works [Luo et al., 2026, Wang et al., 2025b, Meng et al., 2025b, Wu et al., 2025b] each construct $\sim$100 ad-hoc LLM-generated prompts for their own comparisons. These efforts lack explicit per-shot entity schedules specifying which characters, objects, and locations should appear in each shot, and do not evaluate simultaneous multi-entity consistency at scale over 12 shots per episode. Our benchmark fills this gap with 140 curated episodes of up to 50 shots spanning easy/medium/hard tiers, explicit entity schedules tracking all entity types per shot, and a dual evaluation framework combining automated metrics with VLM-based holistic judgment for intra-shot quality and inter-shot consistency.

Multi-Shot Video Generation. While text-to-video models [Wan et al., 2025, Yang et al., 2024, Zheng et al., 2024] now produce high-fidelity single-shot clips, real-world narratives demand *multi-shot* sequences with consistent characters and scenes across shot boundaries [Guo et al., 2025, Meng et al., 2025b, Luo et al., 2026]. A comprehensive survey of these methods is beyond the scope of this work but there is already a rich body of work where existing approaches roughly fall into three broad categories: (1) Two-stage keyframe-then-animate methods that first generate consistent keyframes and then animate each with an image-to-video (I2V) model [Zhou et al., 2024, Huang et al., 2024a, Meng et al., 2025b, Xiao et al., 2025, Yang et al., 2026, Zhang et al., 2025, Zhou et al., 2026], (2) Holistic multi-shot methods that jointly process all shots in a single denoising pass, learning cross-shot consistency directly from data [Guo et al., 2025, Meng et al., 2025b, Wu et al., 2025b, Wang et al., 2025b, Qi et al., 2025, Wang et al., 2025a, Kara et al., 2025, Cai et al., 2025, Jia et al., 2025], and (3) Autoregressive multi-shot methods that reformulate the task as sequential next-shot prediction [Luo et al., 2026, Yin et al., 2025, Huang et al., 2025b, Liu et al., 2025a, Yang et al., 2025, Yesiltepe et al., 2025]. Across all three paradigms, entity consistency emerges implicitly from architectural design rather than being an explicit objective. Our work addresses this gap with both a benchmark that directly measures per-entity consistency across shots and a multi-agent system with explicit per-entity visual memory management.

3 EntityBench: Cross-Shot Entity Consistency Benchmark

As multi-shot video generation methods advance, there is a need for standardized evaluation of entity consistency across shots. Existing works typically evaluate on ad-hoc sets of LLM-generated prompts [Wu et al., 2025b, Wang et al., 2025b, Meng et al., 2025b], or on small controlled benchmarks that isolate individual entity types [Zhou et al., 2026] rather than evaluating simultaneous multi-entity consistency. EntityBench provides a curated benchmark of 140 episodes totaling approximately 2,491 shots across easy, medium, and hard difficulty tiers, with explicit entity schedules that specify which characters, objects, and locations should appear in each shot, and a standardized evaluation framework for intra-shot quality and inter-shot entity consistency.

3.1 Data Construction

Source data. Constructing multi-shot video scripts with natural entity dynamics is difficult through LLM prompting alone: scheduling entities across shots with reasonable occurrence patterns, diverse interactions, and coherent scene structures remains an open challenge. EntityBench instead derives its scripts from existing narrative media, filtered by visual clarity, aesthetics, and motion quality. This provides a foundation of natural character interactions, scene transitions, and entity reappearance patterns that reflect how characters, objects, and locations actually co-occur in real media. Yet, the source material serves only as a seed. The final scripts are generated through LLM-based enrichment, allowing for story adaptation and tolerance for deviation from the original narrative.

Entity extraction and linking. From the source material, we extract shots and identify recurring entities through a multi-stage annotation pipeline. Characters are first detected per frame using an object detector [Wang et al., 2024] and a face detector with embedding extraction [Deng et al., 2019], then tracked into per-shot tracklets via IoU-based assignment [Zhang et al., 2022]. To establish cross-shot character identities, tracklet embeddings (face and body features [Oquab et al., 2023]) are clustered within each episode using hierarchical agglomerative clustering, with a co-occurrence constraint that rejects merges between tracklets overlapping temporally within the same shot. However, embedding-based clustering alone produces fragmented identities with limited recurrence, especially when appearances are far apart in the source material. We address this with an LLM-based [Comanici et al., 2025] deduplication stage that consolidates character clusters across distant shots, while preserving the constraint that characters co-occurring in the same shot or sharing adjacent tracklet IDs remain distinct. Objects and locations are harder to cluster from visual features alone due to the entanglement of foreground and background regions. We instead use an LLM [Comanici et al., 2025] to first propose local registries within temporal chunks, then merge them into episode-level identities, and finally verify their appearance against the episode script and videos for potential contradictions.

Script refinement and enrichment. With entity identities established, the raw annotations undergo a multi-stage enrichment pipeline to produce generation-ready video prompts. Character descriptions are polished to focus on detailed facial features and demographics while removing actions, camera directions, and transient states. Object descriptions are refined to distinguish visual properties from functional context, and location descriptions are expanded with spatial detail. Per-shot action text is then enriched to avoid static actions and encourage interactions between characters, guided by the resolved entity schedules, the global story context, and a temporal window of neighboring shots.

Verification. We verify that every entity in a shot's entity schedule is actually mentioned in the action text, repairing mismatches through targeted LLM calls that decide whether it is logical to add the missing entity or remove it from the schedule. We adopt multi-pass refinement and validation in this stage. A final validation stage detects and repairs contradictions between action text and entity descriptions, as well as physically impossible actions. EntityBench provides a structured story script per episode containing scene boundaries, shot descriptions with entity descriptions, and an explicit entity schedule mapping each shot to its scheduled characters, objects, and locations.

Benchmark. Table 2 summarizes the curation funnel. Starting from 100K production clips, quality filtering retains 46% based on visual clarity and aesthetics. After entity annotation, content filtering removes 27% of episodes with high subtitle density or documentary-style content. Finally, we select the best contiguous shot windows using a sliding-window approach that scores windows by cross-shot entity recurrence, interaction density, and scene transition frequency, retaining 5% of

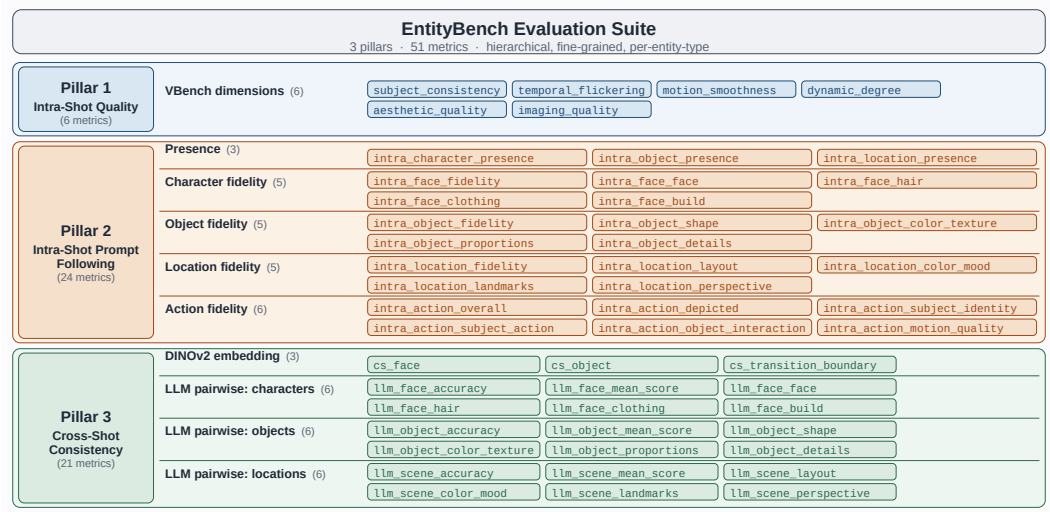


Figure 1: Overview of the EntityBench evaluation suite. Three pillars progressively assess whether each shot is well-formed (Pillar 1), whether it follows its prompt (Pillar 2), and whether entities remain consistent across shots (Pillar 3). Pillar 2's per-entity fidelity scores gate admission into Pillar 3's cross-shot pool. 51 metrics total across 3 pillars.

all shots. The final benchmark contains 140 episodes totaling 2,491 shots across 1,146 scenes, all passing comprehensive verification. As shown in Table 3, easy episodes contain 8–12 shots, already matching or exceeding existing benchmarks in scale, while hard episodes average 8.9 cross-shot characters with a maximum recurrence gap of 33.5 shots. 62% of hard-tier shots are recurrence-only (no first-appearance description), serving as direct tests of entity memory.

3.2 Evaluation Framework

EntityBench evaluates generated multi-shot videos through three pillars that ask questions progressively : (i) is each shot well-formed in isolation, (ii) does each shot match its prompt, and (iii) do shots agree with one another. Pillars build on each other. For example, Pillar 2's per-shot fidelity scores filter the cross-shot pool used in Pillar 3, and the same canonical entity crops are shared across pillars, so the audit chain is cohesive.

Pillar 1: Intra-shot quality. Inspired by Huang et al. [2024b], we adopt six intra-shot quality dimensions, including subject consistency, temporal flickering, motion smoothness, dynamic degree, aesthetic quality, and imaging quality. The first pillar measures each shot's quality independently.

Pillar 2: Intra-shot prompt-following alignment. For each shot, we evaluate three aspects of prompt-following through a unified grounding pass. GroundingDINO [Liu et al., 2024a] localizes each scheduled entity using the entity registry description as the query, yielding per-entity crops with a tri-valued status (present/weak/absent) gated on a CLIP [Radford et al., 2021] text-image similarity threshold. We then measure: (i) **presence**: the fraction of scheduled entities that achieve status *present* in the shot, computed separately for characters, objects, and locations; (ii) **per-entity fidelity**: a multimodal LLM [Comanici et al., 2025] scores each canonical crop against its registry description on type-specific criteria. It considers face, hair, clothing, build for characters; shape, color/texture, proportions, details for objects; layout, color mood, landmarks, perspective for locations; (iii) **action fidelity**: a labeled multi-frame grid is constructed by tiling six bounding-box-annotated frames into an image, and the LLM judges whether the prompted action is depicted correctly across six sub-criteria.

Pillar 3: Cross-shot consistency. For each entity that recurs across multiple shots, we measure whether its visual appearance remains stable. The pillar uses two signals computed on the canonical crops from Pillar 2: (i) **Embedding similarity** [Oquab et al., 2023] to a per-entity centroid, computing cross-shot consistency for characters and objects. A cross-shot transitioning boundary metric measures continuity at the scene-internal cuts. (ii) **LLM pairwise judging**: each non-anchor appearance is compared to a centroid-representative anchor on the same type-specific criteria as Pillar 2,

for accuracy and per-criterion similarity scores. Locations use full frames with a camera-invariant prompt that explicitly handles different angles and partial views of the same place. Centroid-anchored similarity is adopted rather than first-anchor, because the centroid is invariant to shot ordering, and more robust to outliers.

Cross-shot fidelity gate. A naive cross-shot metric may mistakenly reward methods that produce nearly static yet incorrect renderings. They are similar to one another, so their consistency scores are high despite a lack of entity fidelity. We prevent this by gating the cross-shot pool on Pillar 2's per-shot fidelity. Only (shot, entity) pairs with intra-shot fidelity above a threshold are admitted into Pillar 3 cross-shot computation. This ensures cross-shot consistency is measured on appearances where the entity was rendered correctly in the first place. Following this principle, we report all per-entity metrics as fidelity-gate-corrected means: an instance-weighted mean over all eligible (shot, entity) instances, with gate-skipped instances counting as zero contributions. This convention jointly captures rendering fidelity (the gate pass-rate) and consistency (the score on passed instances), preventing methods from inflating their scores by failing the gate on harder cases.

4 EntityMem: Entity-Aware Context Management

Multi-shot video narratives require characters, objects, and locations to maintain consistent visual identities across shots. Existing multi-shot approaches have shown conditioning each shot on whole-frame keyframes from earlier generations can improve overall consistency [Zhang et al., 2025, Zhou et al., 2026]. EntityMem explores whether it can further help the entity consistency by maintaining a persistent *entity memory bank* that stores isolated, per-entity visual and textual references rather than whole-frame keyframes. As a starting point, references are generated and verified before any video generation begins, so that each entity's visual identity is established once and reused consistently throughout the sequence. At generation time, the video backbone retrieves each entity's appearance independently of the scene in which it previously appeared, disentangling entity identity from scene context. *The full design is provided in Appendix C.*

The pipeline operates in three stages, each managed by specialized LLM agents [Comanici et al., 2025] that make planning, selection, and verification decisions while delegating deterministic execution to tools, such as text-to-image generator [Labs, 2024] and segmentation model [Ravi et al., 2024].

Stage 1: Entity reference generation. *A Classification Agent* first determines which entities require standalone visual references: characters always receive portraits, locations receive panoramic backgrounds, and objects are evaluated individually. For each entity that requires a reference, a *Portrait Agent* gathers the entity's description and first-appearance context, infers the visual style from the story overview, and writes a generation prompt. A text-to-image model produces N candidates on a chroma-key background, a segmentation model extracts the foreground of each, and the Portrait Agent selects the best result from a composite grid. A *Verification Agent* then inspects the selected portrait for incorrect characteristics or segmentation failures. If verification fails, the pipeline retries with an alternative background color to improve segmentation contrast. For locations, a panoramic image is generated and cropped into angle variants (left, center, right) for camera-aware keyframe composition. The bank also stores a textual description of each entity at its first appearance for prompt injection in later shots.

Stage 2: Keyframe composition. A *Layout Agent* translates each shot's narrative action into one or more keyframe layouts. Given the action text, entity schedule, and (for continuation shots) the previous shot's layout, it determines character positions, camera angle, and the number of keyframes needed. When the action changes the spatial arrangement mid-shot, the agent produces multiple keyframes capturing the progression. For continuation shots, the agent reasons about camera panning direction, shifting retained characters accordingly and selecting the matching location angle variant. A compositor then places height-normalized portraits at planned positions alongside scheduled objects.

Stage 3: Memory-augmented generation. The memory bank for each shot is assembled as an ordered sequence: per-character labeled portraits, followed by keyframe composites. The video backbone receives this alongside a text prompt that includes entity descriptions and shot actions. For recurring entities, stored descriptions are injected into the prompt automatically. For continuation shots, the last frame of the previous shot serves as a first-frame input for temporal continuity but is excluded from the memory bank to prevent it from overriding curated entity references.

Table 4: Main results on EntityBench. Reported as fidelity-gate-corrected means (§3.2). **Bold** marks the best score per row; † marks wins by EntityMem with Cohen's $d > 0.5$ vs the next-best baseline (Table 5). Pillars 1 and 2 evaluate within-shot quality and prompt alignment; Pillar 3 evaluates cross-shot consistency. For VBench (Pillar 1), `imaging_quality` is on $[0, 100]$; all other metrics are on $[0, 1]$. Full 51-metric results in Appendix F.1.

Pillars 1 & 2: Intra-shot					**Pillar 3: Cross-shot**				
Metric	Ours	StoryMem	HoloCine	CineTrans	Metric	Ours	StoryMem	HoloCine	CineTrans
P1: Quality					*P3: DINOv2 similarity*				
imaging_quality	66.00	56.41	49.97	**68.57**	cs_face	0.737	**0.792**	0.751	0.772
aesthetic_quality	0.593	0.475	0.518	**0.596**	cs_object	0.798	**0.839**	0.803	0.794
motion_smoothness	0.988	0.849	0.964	**0.990**	cs_transition_boundary	**0.738**	0.663	0.498	0.508
P2: Presence					*P3: LLM characters*				
char_presence	**0.967**†	0.849	0.882	0.796	llm_face_accuracy	**0.406**†	0.226	0.228	0.091
obj_presence	0.888	**0.893**	0.723	0.776	llm_face_mean_score	**0.426**†	0.234	0.242	0.145
loc_presence	**0.687**	0.681	0.624	0.651	llm_face_face	**0.381**†	0.216	0.223	0.145
P2: Fidelity (overall)					*P3: LLM objects*				
face_fidelity	**0.740**†	0.452	0.349	0.327	llm_object_accuracy	0.164	**0.203**	0.088	0.092
object_fidelity	0.601	**0.618**	0.267	0.384	llm_object_mean_score	0.202	**0.222**	0.094	0.145
location_fidelity	**0.555**	0.504	0.306	0.428	llm_object_shape	0.232	**0.239**	0.104	0.180
P2: Action					*P3: LLM scenes*				
action_overall	**0.618**†	0.547	0.569	0.273	llm_scene_accuracy	0.309	**0.398**	0.304	0.119
action_subject	**0.706**†	0.595	0.606	0.478	llm_scene_mean_score	0.659	**0.671**	0.616	0.432
action_interaction	**0.781**†	0.712	0.616	0.346	llm_scene_layout	**0.697**	0.684	0.641	0.449

5 Experiments

5.1 Experimental Setup

We evaluate three representative open-sourced SotA methods on EntityBench. For the holistic paradigm, we evaluate HoloCine [Meng et al., 2025b], which jointly processes all shots in a single denoising pass with window cross-attention and sparse inter-shot self-attention, and CineTrans [Wu et al., 2025b], which uses mask-based transition control for cinematic shot boundaries. For the two-stage keyframe-then-animate paradigm, we evaluate StoryMem [Zhang et al., 2025], which introduces a persistent memory module for cross-shot keyframe retrieval. We additionally evaluate EntityMem, which extends StoryMem with per-entity memory management without additional training. To ensure fair comparison, we convert EntityBench's structured story scripts into each method's native prompt format (e.g., character names or abstract entity IDs), resolving entity schedule annotations into the input representation each method expects.

All experiments are conducted on two nodes with 8 NVIDIA L20 GPUs. Given the scale of EntityBench (2,491 shots), each full benchmark run requires substantial compute. We report all 51 metrics across the three pillars of the EntityBench evaluation framework (§3.2), with the fidelity gate (§3.2) and corresponding fidelity-gate-corrected aggregation applied throughout.

5.2 EntityBench Evaluation

Table 4 reports fidelity-gate-corrected means across representative metrics from each pillar. The full 51-metric breakdown appears in Appendix F.1, and Cohen's d effect sizes for the head-to-head against the strongest baseline are reported in Table 5.

EntityMem dominates entity-centric prompt-following (Pillar 2). Across all five Pillar 2 sub-categories, EntityMem produces the most prompt-aligned characters and scenes. Character-related metrics show the largest gains: `face_fidelity` reaches 0.740 vs. 0.452 for the next-best baseline (StoryMem), with all four sub-criteria (face, hair, clothing, build) won by margins of 0.18–0.30 (Appendix F.1). EntityMem also achieves the highest character presence (0.967, vs. 0.882 for HoloCine), demonstrating that scheduled characters consistently appear in their intended shots. On action correctness, EntityMem's overall score (0.618) leads the next baseline by 0.05, with the largest gaps on `subject_identity` (+0.11) and `object_interaction` (+0.07). The per-entity memory bank not only renders characters correctly, but also keeps them recognizable while they execute the prompted action. Location fidelity follows the same pattern, with EntityMem winning all five

Table 5: Paired effect sizes of EntityMem vs. StoryMem on EntityBench, by metric category. Cohen's d is reported with pooled variance (positive favors EntityMem). n_{paired} is the number of episodes where both methods produced an evaluable score (averaged across metrics in the category). Per-metric values are in Appendix F.3.

Category	**# metrics**	**Avg.** d	n_{paired}
Where EntityMem helps most: character-centric metrics			
Character fidelity (intra-shot)	5	$+\mathbf{1.71}$	139
Character presence	1	$+\mathbf{1.23}$	139
Action overall & sub-criteria	6	$+\mathbf{0.25}$	138
Location fidelity (intra-shot)	5	$+\mathbf{0.17}$	140
LLM character (cross-shot)	6	-0.07^{*}	129
Where EntityMem trails: object-centric and embedding-similarity metrics			
Object presence	1	-0.24	138
Object fidelity (intra-shot)	5	-0.33	138
DINOv2 cross-shot (face / object / boundary)	3	$-0.50^{\dagger}$	124
LLM object (cross-shot)	6	-0.60	121
LLM scene (cross-shot)	6	-0.14	140
Single-Shot Metrics			
VBench intra-shot quality (Pillar 1)	6	$+0.13$	140

sub-criteria. The single Pillar 2 sub-category where EntityMem does not lead is object fidelity, where StoryMem holds a small margin (0.618 vs. 0.601); we discuss this trade-off in §F.4.

Cross-shot consistency: identity vs. embedding similarity (Pillar 3). Pillar 3 reveals a structural disagreement between embedding-based metrics and LLM identity judgment. On DINOv2 cosine similarity, StoryMem leads on `cs_face` (0.792 vs. 0.737) and `cs_object` (0.839 vs. 0.798). However, on the LLM-judged identity metrics that ask whether the same character is recognizably the same character across shots, EntityMem dominates. `llm_face_accuracy` reaches 0.406 vs. 0.226 for StoryMem (a 1.8× improvement), and EntityMem wins all six LLM character cross-shot metrics. This disagreement reflects a different concentration embedding-similarity metrics on consistency, where high embedding similarity may not relate to correct identities preserving similar details. EntityMem also wins `cs_transition_boundary` (0.738 vs. 0.663), capturing continuity at scene-internal cuts, and ties with StoryMem on the new camera-invariant scene metric (`llm_scene_layout` 0.697 vs. 0.684; `llm_scene_perspective` 0.727 vs. 0.696, both leading).

Visual quality vs. entity consistency are distinct. On Pillar 1 VBench dimensions, CineTrans wins three of three highlighted dimensions (`imaging_quality`, `aesthetic_quality`, `motion_smoothness`); HoloCine wins `dynamic_degree` and `temporal_flickering` on the full VBench (Appendix F.1). Both are holistic multi-shot methods that produce all shots in a single denoising pass, which favors per-frame polish but does not, by itself, specifically enforce entity-level consistency across shots. EntityMem is competitive on visual quality (second on imaging quality, second on aesthetic quality), but its contribution lies in a complementary direction that produces the most identifiable and prompt-aligned entities across long multi-shot sequences. The contrast is most visible on `character_presence`, where CineTrans drops to 0.796 despite winning the quality dimensions, and on `face_fidelity`, where CineTrans renders characters at less than half of EntityMem's quality (0.327 vs. 0.740).

5.3 Where EntityMem Helps Most

EntityMem builds and manages a per-entity memory bank which influences the rendering of recurring characters. Table 5 summarizes paired effect sizes across metric categories. The largest single effect is intra-shot character fidelity, with the broader character-fidelity category (face, hair, clothing, build) averaging $d = +1.71$. Character presence moves substantially as well ($d = +1.23$). EntityMem renders the scheduled character in 96.7% of shots vs. 84.9% for StoryMem, meaning roughly one of every eight scheduled character appearances is missing from StoryMem outputs. Both effects trace to the same architectural choice: each character is regenerated against its own dedicated memory bank rather than being averaged into a shared per-shot context. When a shot needs to depict a character,

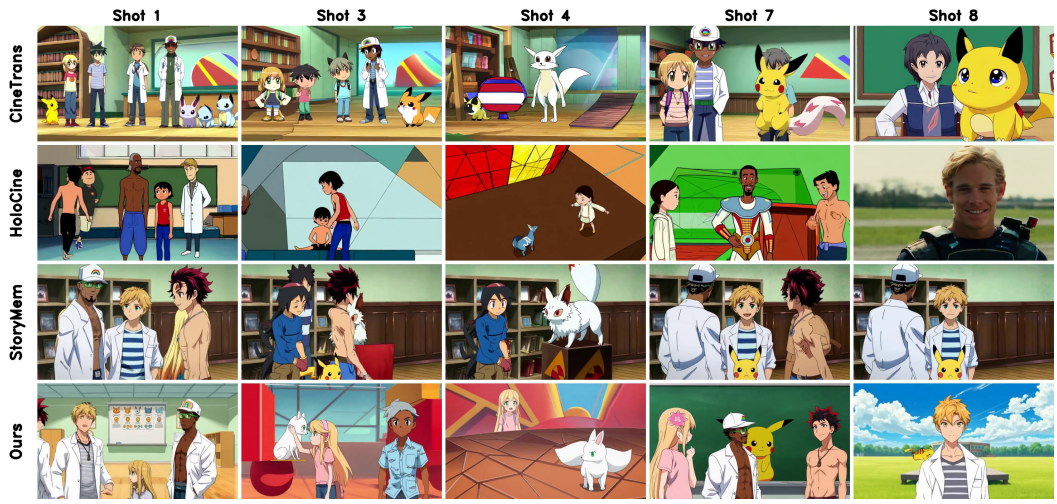


Figure 2: **Qualitative comparison on a representative episode.** Multiple characters recur in shots 1, 3, 4, 7, 8. EntityMem preserves character identity while changing locations according to the prompt.

the model conditions on a tight, per-entity description that survives across shots without being diluted by other entities or scene-level conditioning.

The picture inverts on objects ($d = -0.33$ intra-shot fidelity, $d = -0.60$ in pairwise cross-shot LLM scoring) and on DINOv2 cross-shot embeddings ($d = -0.50$). The DINOv2 deficit, however, is not a cross-shot consistency loss but an embedding-similarity limitation: on the LLM-judged cross-shot character metrics that evaluate identity rather than embedding distance, the comparison is essentially tied (Appendix F.2). The object regression is real: StoryMem's scene-level prompt expansion appears to retain object identity better when objects are scene-bound props rather than character-attached items. It may cause by the condition and entity incompatibility with the keyframe-finetuned storymem weight. The base model lacks knowledge of integrating objects with the video from independent object conditions.

5.4 Qualitative Comparison

Figure 2 provides qualitative comparisons. Across the four methods, identity stability and prompt alignment scales directly to per-entity context. The holistic generators (CineTrans, HoloCine) lose character identity gradually despite producing high-quality individual frames, while the persistent-memory baseline (StoryMem) preserves *some* characters but inserts entities not scheduled in the script and fails at generating corresponding locations. EntityMem's per-entity memory bank, preserves all four recurring characters and the recurring Pokémon across all eight shots while transitioning to different locations. *Qualitative video examples are provided in the supplementary materials.*

6 Conclusion

We introduced EntityBench, a comprehensive benchmark for evaluating entity consistency in multi-shot video generation, comprising 140 episodes (2,491 shots) derived from real narrative media with explicit per-shot entity schedules across three difficulty tiers. The accompanying three-pillar evaluation suite provides 51 metrics spanning intra-shot quality, prompt-following alignment, and cross-shot entity consistency, enabling fine-grained diagnosis of where and why current methods fail to maintain entity identity over long sequences. Using EntityBench, we showed that cross-shot consistency degrades with recurrence distance. Through EntityMem, per-entity visual and textual memory management system, we show that entity condition for shot generation improves the quality and consistency on 29 dimensions.

Acknowledgments: V.O. was partially supported by the Ken Kennedy Institute and NSF Awards No. 2201710 and No 2601419.

References

Shengqu Cai, Ceyuan Yang, Lvmin Zhang, Yuwei Guo, Junfei Xiao, Ziyan Yang, Yinghao Xu, Zhenheng Yang, Alan Yuille, Leonidas Guibas, et al. Mixture of contexts for long video generation. *arXiv preprint arXiv:2508.21058*, 2025.

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.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019.

Xiaokun Feng, Haiming Yu, Meiqi Wu, Shiyu Hu, Jintao Chen, Chen Zhu, Jiahong Wu, Xiangxiang Chu, and Kaiqi Huang. Narrlv: Towards a comprehensive narrative-centric evaluation for long video generation. *arXiv preprint arXiv:2507.11245*, 2025.

Jianxiong Gao, Zhaoxi Chen, Xian Liu, Jianfeng Feng, Chenyang Si, Yanwei Fu, Yu Qiao, and Ziwei Liu. Longvie: Multimodal-guided controllable ultra-long video generation. *arXiv preprint arXiv:2508.03694*, 2025.

Yuwei Guo, Ceyuan Yang, Ziyan Yang, Zhibei Ma, Zhijie Lin, Zhenheng Yang, Dahua Lin, and Lu Jiang. Long context tuning for video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17281–17291, 2025.

Hui Han, Siyuan Li, Jiaqi Chen, Yiwen Yuan, Yuling Wu, Yufan Deng, Chak Tou Leong, Hanwen Du, Junchen Fu, Youhua Li, et al. Video-bench: Human-aligned video generation benchmark. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18858–18868, 2025.

Xuanhua He, Quande Liu, Shengju Qian, Xin Wang, Tao Hu, Ke Cao, Keyu Yan, and Jie Zhang. Id-animator: Zero-shot identity-preserving human video generation. *arXiv preprint arXiv:2404.15275*, 2024.

Kaiyi Huang, Yukun Huang, Xintao Wang, Zinan Lin, Xuefei Ning, Pengfei Wan, Di Zhang, Yu Wang, and Xihui Liu. Filmaster: Bridging cinematic principles and generative ai for automated film generation. *arXiv preprint arXiv:2506.18899*, 2025a.

Lianghua Huang, Wei Wang, Zhi-Fan Wu, Yupeng Shi, Huanzhang Dou, Chen Liang, Yutong Feng, Yu Liu, and Jingren Zhou. In-context lora for diffusion transformers. *arXiv preprint arXiv:2410.23775*, 2024a.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009*, 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21807–21818, 2024b.

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.

Weinan Jia, Yuning Lu, Mengqi Huang, Hualiang Wang, Binyuan Huang, Nan Chen, Mu Liu, Jidong Jiang, and Zhendong Mao. Moga: Mixture-of-groups attention for end-to-end long video generation. *arXiv preprint arXiv:2510.18692*, 2025.

Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17191–17202, 2025.

Ozgur Kara, Krishna Kumar Singh, Feng Liu, Duygu Ceylan, James M Rehg, and Tobias Hinz. Shotadapter: Text-to-multi-shot video generation with diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 28405–28415, 2025.

Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5148–5157, 2021.

Jaywon Koo, Jefferson Hernandez, Moayed Haji-Ali, Ziyan Yang, and Vicente Ordonez. Evaluating Text-to-Image and Text-to-Video Synthesis with a Conditional Fréchet Distance. In *2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2052–2062, 2026. doi: 10.1109/WACV61042.2026.00204.

Black Forest Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024.

Mingxiang Liao, Hannan Lu, Xinyu Zhang, Fang Wan, Tianyu Wang, Yuzhong Zhao, Wangmeng Zuo, Qixiang Ye, and Jingdong Wang. Evaluation of text-to-video generation models: A dynamics perspective. *Advances in Neural Information Processing Systems*, 37:109790–109816, 2024.

Han Lin, Abhay Zala, Jaemin Cho, and Mohit Bansal. Videodirectorgpt: Consistent multi-scene video generation via llm-guided planning. *arXiv preprint arXiv:2309.15091*, 2023.

Xinran Ling, Chen Zhu, Meiqi Wu, Hangyu Li, Xiaokun Feng, Cundian Yang, Aiming Hao, Jiashu Zhu, Jiahong Wu, and Xiangxiang Chu. Vmbench: A benchmark for perception-aligned video motion generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13087–13098, 2025.

Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. Rolling forcing: Autoregressive long video diffusion in real time. *arXiv preprint arXiv:2509.25161*, 2025a.

Lijie Liu, Tianxiang Ma, Bingchuan Li, Zhuowei Chen, Jiawei Liu, Gen Li, Siyu Zhou, Qian He, and Xinglong Wu. Phantom: Subject-consistent video generation via cross-modal alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14951–14961, 2025b.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pages 38–55. Springer, 2024a.

Yaofang Liu, Xiaodong Cun, Xuebo Liu, Xintao Wang, Yong Zhang, Haoxin Chen, Yang Liu, Tieyong Zeng, Raymond Chan, and Ying Shan. Evalcrafter: Benchmarking and evaluating large video generation models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22139–22149, 2024b.

Yuanxin Liu, Lei Li, Shuhuai Ren, Rundong Gao, Shicheng Li, Sishuo Chen, Xu Sun, and Lu Hou. Fetv: A benchmark for fine-grained evaluation of open-domain text-to-video generation. *Advances in Neural Information Processing Systems*, 36:62352–62387, 2023.

Yawen Luo, Xiaoyu Shi, Junhao Zhuang, Yutian Chen, Quande Liu, Xintao Wang, Pengfei Wan, and Tianfan Xue. Shotstream: Streaming multi-shot video generation for interactive storytelling. *arXiv preprint arXiv:2603.25746*, 2026.

Xiangyu Meng, Zixian Zhang, Zhenghao Zhang, Junchao Liao, Long Qin, and Weizhi Wang. Identity-grpo: Optimizing multi-human identity-preserving video generation via reinforcement learning. *arXiv preprint arXiv:2510.14256*, 2025a.

Yihao Meng, Hao Ouyang, Yue Yu, Qiuyu Wang, Wen Wang, Ka Leong Cheng, Hanlin Wang, Yixuan Li, Cheng Chen, Yanhong Zeng, et al. Holocine: Holistic generation of cinematic multi-shot long video narratives. *arXiv preprint arXiv:2510.20822*, 2025b.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Tianhao Qi, Jianlong Yuan, Wanquan Feng, Shancheng Fang, Jiawei Liu, SiYu Zhou, Qian He, Hongtao Xie, and Yongdong Zhang. Maskˆ 2dit: Dual mask-based diffusion transformer for multi-scene long video generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18837–18846, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

Haoyuan Shi, Yunxin Li, Nanhao Deng, Zhenran Xu, Xinyu Chen, Longyue Wang, Baotian Hu, and Min Zhang. Msvbench: Towards human-level evaluation of multi-shot video generation. *arXiv preprint arXiv:2602.23969*, 2026.

Jaskirat Singh, Junshen Kevin Chen, Jonas Kohler, and Michael Cohen. Storybooth: Training-free multi-subject consistency for improved visual storytelling. *arXiv preprint arXiv:2504.05800*, 2025.

Kaiyue Sun, Kaiyi Huang, Xian Liu, Yue Wu, Zihan Xu, Zhenguo Li, and Xihui Liu. T2v-compbench: A comprehensive benchmark for compositional text-to-video generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8406–8416, 2025.

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 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.

Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, and Guiguang Ding. Yolov10: Real-time end-to-end object detection. *Advances in neural information processing systems*, 37: 107984–108011, 2024.

Jiahao Wang, Hualian Sheng, Sijia Cai, Weizhan Zhang, Caixia Yan, Yachuang Feng, Bing Deng, and Jieping Ye. Echoshot: Multi-shot portrait video generation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a.

Qinghe Wang, Xiaoyu Shi, Baolu Li, Weikang Bian, Quande Liu, Huchuan Lu, Xintao Wang, Pengfei Wan, Kun Gai, and Xu Jia. Multishotmaster: A controllable multi-shot video generation framework. *arXiv preprint arXiv:2512.03041*, 2025b.

Yuji Wang, Moran Li, Xiaobin Hu, Ran Yi, Jiangning Zhang, Han Feng, Weijian Cao, Yabiao Wang, Chengjie Wang, and Lizhuang Ma. Identity-preserving text-to-video generation guided by simple yet effective spatial-temporal decoupled representations. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 13743–13750, 2025c.

Zun Wang, Jialu Li, Han Lin, Jaehong Yoon, and Mohit Bansal. Dreamrunner: Fine-grained compositional story-to-video generation with retrieval-augmented motion adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 10503–10511, 2026.

Weijia Wu, Mingyu Liu, Zeyu Zhu, Xi Xia, Haoen Feng, Wen Wang, Kevin Qinghong Lin, Chunhua Shen, and Mike Zheng Shou. Moviebench: A hierarchical movie level dataset for long video generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 28984–28994, 2025a.

Xiaoxue Wu, Bingjie Gao, Yu Qiao, Yaohui Wang, and Xinyuan Chen. Cinetrans: Learning to generate videos with cinematic transitions via masked diffusion models. *arXiv preprint arXiv:2508.11484*, 2025b.

Junfei Xiao, Ceyuan Yang, Lvmin Zhang, Shengqu Cai, Yang Zhao, Yuwei Guo, Gordon Wetzstein, Maneesh Agrawala, Alan Yuille, and Lu Jiang. Captain cinema: Towards short movie generation. In *The Fourteenth International Conference on Learning Representations*, 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.

Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Yingcong Chen, Yao Lu, et al. Longlive: Real-time interactive long video generation. *arXiv preprint arXiv:2509.22622*, 2025.

Songlin Yang, Zhe Wang, Xuyi Yang, Songchun Zhang, Xianghao Kong, Taiyi Wu, Xiaotong Zhao, Ran Zhang, Alan Zhao, and Anyi Rao. Shotverse: Advancing cinematic camera control for text-driven multi-shot video creation. *arXiv preprint arXiv:2603.11421*, 2026.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

Hidir Yesiltepe, Tuna Han Salih Meral, Adil Kaan Akan, Kaan Oktay, and Pinar Yanardag. Infinity-rope: Action-controllable infinite video generation emerges from autoregressive self-rollout. *arXiv preprint arXiv:2511.20649*, 2025.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22963–22974, 2025.

Shenghai Yuan, Xianyi He, Yufan Deng, Yang Ye, Jinfa Huang, Bin Lin, Jiebo Luo, and Li Yuan. Opens2v-nexus: A detailed benchmark and million-scale dataset for subject-to-video generation. *arXiv preprint arXiv:2505.20292*, 2025a.

Shenghai Yuan, Jinfa Huang, Xianyi He, Yunyang Ge, Yujun Shi, Liuhan Chen, Jiebo Luo, and Li Yuan. Identity-preserving text-to-video generation by frequency decomposition. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12978–12988, 2025b.

Kaiwen Zhang, Liming Jiang, Angtian Wang, Jacob Zhiyuan Fang, Tiancheng Zhi, Qing Yan, Hao Kang, Xin Lu, and Xingang Pan. Storymem: Multi-shot long video storytelling with memory. *arXiv preprint arXiv:2512.19539*, 2025.

Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. In *European conference on computer vision*, pages 1–21. Springer, 2022.

Dian Zheng, Ziqi Huang, Hongbo Liu, Kai Zou, Yinan He, Fan Zhang, Lulu Gu, Yuanhan Zhang, Jingwen He, Wei-Shi Zheng, et al. Vbench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness. *arXiv preprint arXiv:2503.21755*, 2025.

Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*, 2024.

Yong Zhong, Zhuoyi Yang, Jiayan Teng, Xiaotao Gu, and Chongxuan Li. Concat-id: Towards universal identity-preserving video synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1906–1915, 2025.

Jinsong Zhou, Yihua Du, Xinli Xu, Luozhou Wang, Zijie Zhuang, Yehang Zhang, Shuaibo Li, Xiaojun Hu, Bolan Su, and Ying-cong Chen. Videomemory: Toward consistent video generation via memory integration. *arXiv preprint arXiv:2601.03655*, 2026.

Table 6: Top-level scale and entity statistics for EntityBench. *Registry* counts the unique entities declared once per episode in the entity-description block. *Total appearances* is the cumulative number of entity-slots across all per-shot schedules; one entity scheduled into one shot counts as one appearance.

Quantity	Total	Mean / episode
Episodes	140	—
Scenes	1,136	$8.1_{\pm 5.3}$
Shots	2,491	$17.8_{\pm 13.3}$
Entity registry (unique entities)		
Characters	987	7.05
Locations	654	4.67
Objects	2,077	14.84
Total registry	**3,718**	**26.56**
Total scheduled appearances (sum over per-shot schedules)		
Characters	4,989	35.64
Locations	2,436	17.40
Objects	4,020	28.71
Total appearances	**11,445**	**81.75**

Yupeng Zhou, Daquan Zhou, Ming-Ming Cheng, Jiashi Feng, and Qibin Hou. Storydiffusion: Consistent self-attention for long-range image and video generation. *Advances in Neural Information Processing Systems*, 37:110315–110340, 2024.

In this appendix, we provide benchmark statistics in Section A, details on evaluation metrics in Section B, EntityMem pipeline details in Section C, data examples in Section D, prompts used for EntityMem in Section E, supplementary experimental results in Section F, additional related work in Section G, and broader impact in Section H.

A Benchmark Statistics

This appendix provides comprehensive descriptive statistics for EntityBench, organized around the four properties that distinguish it from prior multi-shot benchmarks. We report (i) episode-level scale and taxonomy (Section A.1); (ii) per-shot multi-entity composition that probes simultaneous tracking of characters, objects, and locations (Section A.2); (iii) long-range structural properties that constitute the cross-shot memory test signal (Section A.3); and (iv) the prompt-level linguistic profile (Section A.4). Section A.5 closes with distributions that extends Table 3 of the main paper.

A.1 Scale and Taxonomy

EntityBench comprises 140 episodes spanning 1,136 scenes and 2,491 shots. Across the benchmark, episode registries collectively declare $3{,}718$ unique entities, each described once in the registry block of its episode. Per-shot schedules then reference registry entries by name, yielding $11{,}445$ entity-slot appearances aggregated over all 2,491 shots (each appearance is one entity scheduled into one shot). Table 6 reports both quantities with the per-type breakdown.

Characters and locations are scheduled most densely. Each character is referenced by 5.05 shots on average and each location by 3.72 shots, while objects skew toward the long tail of single-shot props (1.94 references per object). This per-type density gap motivates type-specific evaluation criteria (Sections B.3.3 and B.4.3): each character contributes roughly $2.6\times$ more to the cross-shot evaluation pool than each object.

Episode size. Easy and medium episodes range from 10 to 22 shots (median 12, mean 12.4 across these two tiers), drawn from real screenplay structure. The hard tier fixes episode length at 50 shots to provide a controlled stress test of long-range consistency without confounding episode length with content variation. EntityBench covers in-distribution and slightly challenging lengths for existing multi-shot video generation models [Meng et al., 2025b, Wu et al., 2025b, Zhang et al., 2025] plus a

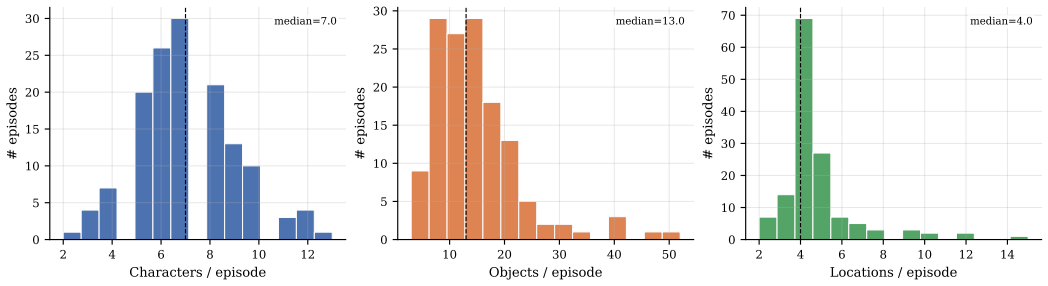


Figure 3: Per-episode entity counts (declared in the registry), broken down by entity type.

Table 7: Per-shot composition of EntityBench. **Left:** mean entity load by type. **Right:** fraction of shots satisfying each compositional condition; "2c+1o" denotes "$\geq$ 2 characters and $\geq$ 1 object," and "tri-type" denotes simultaneous presence of $\geq$ 1 character, $\geq$ 1 object, and $\geq$ 1 location. Single-entity-type evaluation protocols can only audit a small subset of these compositions.

Per-shot entity load	**Mean**
Characters / shot	2.00
Objects / shot	1.61
Locations / shot	0.98
Total entities / shot	**4.59**

Shot-composition fractions	**% shots**
0 characters	0.6%
Exactly 1 character	34.1%
Exactly 2 characters	40.6%
$\geq$ 3 characters	24.7%
2c+1o (multi-character + object)	54.3%
Tri-type (character + object + location)	**79.1%**

fixed-length stress test, measuring both typical-case behavior (easy/medium) and worst-case scaling (hard) within a tractable compute budget.

Scene structure. Each episode contains a median of 6 scenes; easy and medium episodes span 2 to 13 scenes, while hard-tier episodes extend up to 38 scenes per episode under the 50-shot constraint. The median shots-per-scene ratio is 2.1, reflecting short-form storytelling pacing and ensuring every episode contains multiple scene transitions, which we use to stratify cross-shot evaluation by cut type (Section B.4.2).

Entity counts per episode. An episode declares on average 7 characters, 5 locations, and 15 objects, with the largest episodes declaring up to 13 characters and 52 objects. Figure 3 shows the per-type histograms. The object distribution has a long right tail: the top 10% of episodes declare more than 25 distinct objects, driven by hard-tier episodes that span multiple sub-environments (kitchen, study, garden, etc.) each contributing their own object inventories.

A.2 Per-Shot Multi-Entity Composition

A central property of EntityBench is that each shot is annotated with a multi-type entity schedule, enabling joint evaluation of character, object, and location consistency rather than evaluation in isolation. Table 7 shows the resulting per-shot composition: the left sub-table reports the mean entity load by type, and the right sub-table reports the fraction of shots satisfying each compositional condition. The mean shot contains 2.0 characters, 1.6 objects, and effectively 1 location, for a mean total entity load of 4.6 scheduled entities. Beyond raw counts, the compositional breakdown highlights the simultaneous multi-entity test signal that distinguishes EntityBench from prior benchmarks: 79.1% of shots schedule at least one entity of *each* of the three types (character, object, location), and 54.3% schedule at least two characters together with at least one object. Figure 4 displays the per-type underlying distributions.

A.3 Long-Range Entity Structure

The cross-shot memory signal in EntityBench is determined by how entities recur across shot boundaries. We summarize four complementary structural quantities: (i) recurrence rates, (ii)

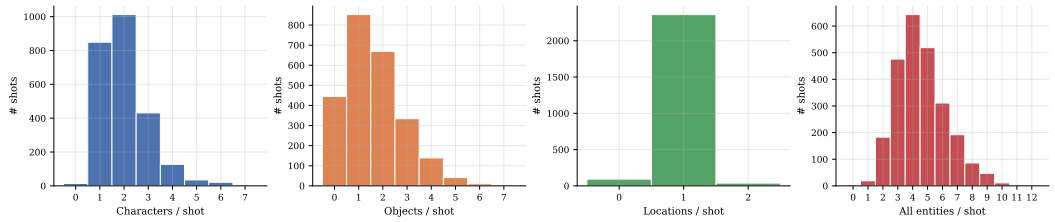


Figure 4: Per-shot entity-load distributions, broken down by type. Location counts cluster tightly at 1 (almost every shot has a single scheduled location), while character and object counts spread across a wide range, with characters concentrated at 1–3 and objects exhibiting a heavier right tail.

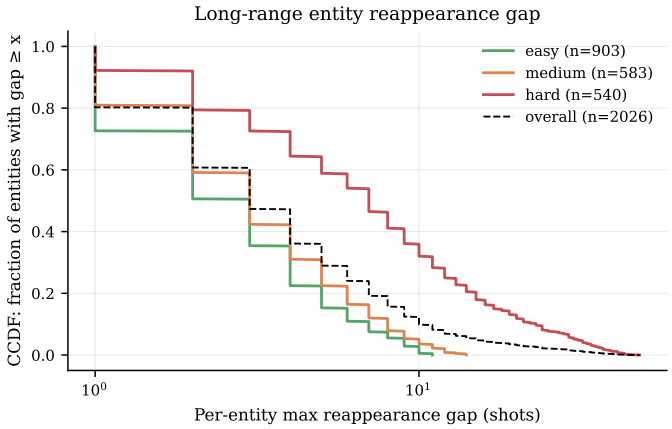


Figure 5: Complementary CDF of per-entity maximum reappearance gap, stratified by tier. The hard-tier curve dominates the easy and medium curves at every gap threshold and carries a heavy tail well past 30 intervening shots, providing a long-range stress test that is absent from prior benchmarks. Counts in the legend (n) are the numbers of recurring entities in each tier; entities that appear in only one shot are excluded by construction.

reappearance gap distributions, (iii) the cut/continuation pattern, and (iv) the registry-vs-memory test signal.

Recurrence and cross-scene reappearance. Of the 3,593 entities scheduled into at least one shot, 2,026 (56.4%) recur in two or more shots, and 1,445 (40.2%) recur across two or more scenes. This places EntityBench firmly in the cross-shot regime: the majority of registry entries cannot be evaluated within a single isolated shot, but only by tracking identity across shots.

Reappearance gap. For each recurring entity we compute the *maximum reappearance gap*: across all consecutive pairs of shots in which the entity appears, the largest number of *intervening* shots that the entity is absent. A gap of 0 means the entity reappeared in immediately consecutive shots; a gap of g means g intervening shots separate the two closest re-appearances. Figure 5 plots the complementary cumulative distribution (CCDF) of this quantity, stratified by tier. The benchmark contains 36.1% of recurring entities with max gap ≥ 5, 12.4% with max gap ≥ 10, and 3.5% with max gap ≥ 20; the global maximum is 48 intervening shots, observed in the hard tier.

Cut and continuation structure. Each shot is annotated with a binary cut flag, yielding a global cut rate of 45.6% (1,136 cuts across 2,491 shots). Equivalently, the benchmark partitions into 1,136 *continuation chains*, which represent maximal runs of consecutive shots not separated by a hard cut, with mean length 2.19 and a maximum chain of 36 consecutive non-cut shots. The distribution is heavily right-skewed (Figure 6): roughly 62% of chains are length-1 isolated shots, while the remaining 38% form multi-shot continuation runs that tests the ability of transitioning from and continuing the previous content. A chain of length $k > 1$ requires $k - 1$ smooth cross-shot transitions

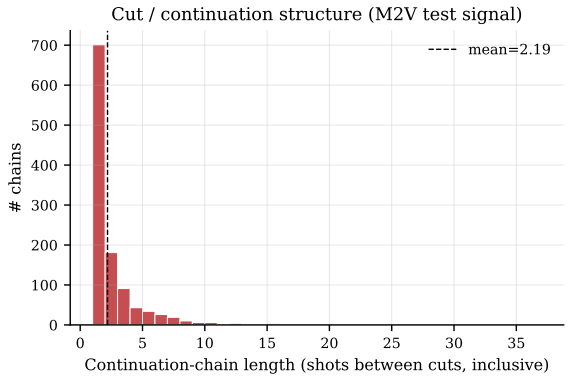


Figure 6: Continuation-chain length distribution (number of consecutive shots between two cuts). The bulk of mass at length 1 corresponds to isolated single-shot scenes; the right tail of multi-shot chains, extending to 36 shots, examine transitioning ability at scale.

Table 8: Memory test signal of EntityBench, measured at the entity-slot level. Each row counts (shot, entity) pairs across the entire benchmark. *First-appearance* pairs ship with a registry description block in the prompt; *re-appearance* pairs reference the entity by name only and must be rendered from episode-level memory of prior appearances.

	Characters	**Locations**	**Objects**	**All entities**
Total entity-slot appearances	4,989	2,436	4,020	11,445
First appearances (registry block in prompt)	984	648	1,892	3,593
Re-appearances (memory test)	4,005	1,788	2,128	7,852
Re-appearance rate	**80.3%**	**73.4%**	**52.9%**	**68.6%**

in addition to per-shot quality, so the right tail of this distribution (**e.g.**, chains of 5 shots and beyond) is the regime that most directly probes transition fidelity at scale.

Cross-cut entity carry-over. Of the 996 within-episode cuts, 555 (55.7%) preserve at least one entity across the boundary, where a character or object that was present in the last shot before the cut reappears in the first shot after it. Carry-over cuts are particularly difficult: the model must maintain identity across an explicit visual discontinuity, with no continuation context. Pure scene-change cuts (no carry-over, 44.3%) could be relatively easier in the consistency sense but force the model to handle an entirely new entity configuration without warm-up.

The memory test signal: re-appearance rate at the entity level. The cross-shot identity test in EntityBench is measured at the level of *entity-slot appearances*. Each (shot, entity) pair is either a *first appearance* in which case the entity's description block is supplied in the shot's prompt header, or a *re-appearance*in which case the entity is referenced by name only and must be rendered from prior context. By construction, the global re-appearance count equals total scheduled appearances minus the number of unique scheduled entities. Across the benchmark, 7,852 of 11,445 entity-slot appearances (**68.6**%) are re-appearances and constitute the memory test signal. A shot-level view, where a shot is counted as "memory-only" iff *all* of its scheduled entities are re-appearances, understates this, because a shot scheduling one new entity alongside two recurring entities still exercises memory on the two recurring entities even though the shot ships with a registry block. Table 8 reports the breakdown.

Characters are tested most aggressively. 80.3% of every character slot in the benchmark must be rendered from memory rather than from a prompt-level description. Locations follow at 73.4%, while objects, dominated by single-shot props, exhibit the lowest rate at 52.9%. The hard tier is even more demanding: 80.7% of all entity-slot appearances in hard episodes are re-appearances (versus 57.7% easy and 64.5% medium), with the per-tier character rate climbing further still.[1]

[1] Per-tier per-type unique-entity counts are summarized in Section A.5.

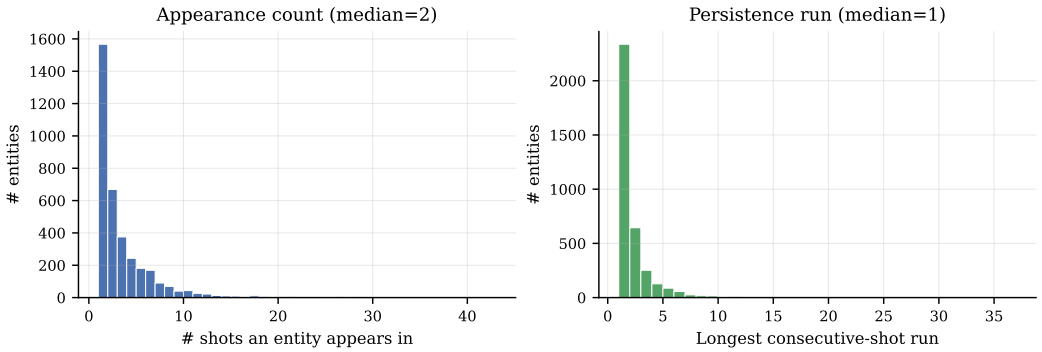


Figure 7: Per-entity persistence statistics. **Left**: number of shots an entity appears in (median 2; right tail extends past 25 appearances). **Right**: longest consecutive-shot run an entity sustains (median 1; the right tail corresponds to anchor entities across multi-shot continuation segments).

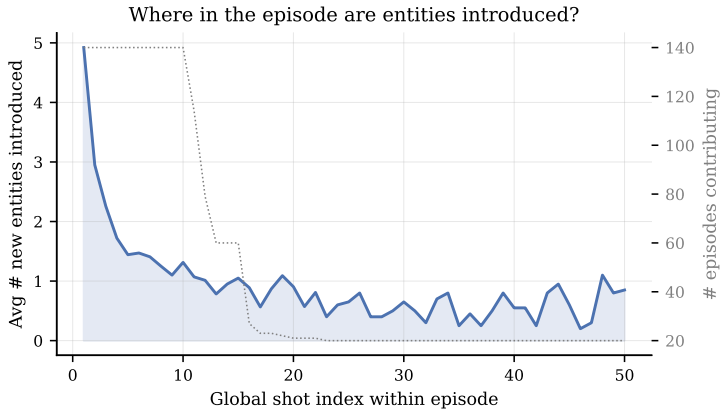


Figure 8: Average number of new entities introduced at each shot index (left axis, blue), with the number of episodes contributing at each index (right axis, gray dashed). New entity introductions are heavily front-loaded but never fully stop: the tail beyond shot ~ 22 reflects the 20 hard-tier episodes, which continue to introduce entities at a steady ~ 0.5–1.0 rate throughout their 50-shot length.

Persistence and appearance counts. Beyond gaps, we also examine entity *persistence*: the longest run of consecutive shots in which an entity appears. The median entity appears in 2 shots (left panel of Figure 7), and roughly two-thirds of entities have a persistence run of 1 – they appear, disappear, and possibly recur later, never anchoring a multi-shot continuation. The right tail of the persistence distribution corresponds to anchor entities that drive the narrative across consecutive shots, with persistence runs extending up to 9 shots.

Where in an episode are entities introduced? Figure 8 plots the average number of new entities introduced at each shot index, averaged across all episodes that contain at least that many shots. The first shot of an episode introduces, on average, ~ 5 new entities (the opening establishes the cast and setting), and roughly 70% of an episode's entity inventory is introduced within the first 10 shots. The curve then plateaus at ~ 0.5–1.0 new entities per shot for the remainder of the episode—hard-tier episodes (the only ones that contribute to shot indices beyond ~ 22) continue to introduce entities at a steady drip well past the midpoint. This shape implies that EntityBench does not partition cleanly into an "introduction phase" followed by a "recall phase"; instead, models must handle both regimes simultaneously throughout long episodes, with the recall burden growing monotonically while introductions never fully cease.

Table 9: Top-18 most frequent action-verb lemmas across all 2,491 action descriptions, with raw counts. Inflections are merged under their lemma; state descriptors (*wear/wearing*, *light/lit/illuminated*, *glow/glowing*) and ambiguous noun-dominant forms (*face, head, hand, hands*) are excluded. Verb extraction uses a curated ∼460-form English verb list rather than a POS tagger, so rarer or domain-specific verbs may be undercounted, but the relative ordering is informative.

Verb	**#**	**Verb**	**#**	**Verb**	**#**
stand	1,023	listen	292	gaze	128
look	711	smile	237	nod	80
sit	539	hold	234	stare	67
speak	530	turn	231	observe	54
watch	369	talk	194	glance	47
walk	315	lean	143	show	44

A.4 Linguistic Profile and Scene-Design Tags

EntityBench prompts are derived from natural narrative scripts rather than synthesized from a fixed template, which is reflected in their linguistic statistics. Across all 2,491 action descriptions the vocabulary contains 5,230 distinct word forms over 93,123 total tokens, yielding a type/token ratio of 0.056. Action descriptions average 37.4 words (median 37); full prompts including the registry header average longer due to the prepended entity descriptions. From a curated set of approximately 460 inflected English action-verb forms we identify 326 distinct verbs in use; Table 9 lists the top-18 action-verb lemmas after merging inflections (stand/stands/standing count as one) and excluding state descriptors (wear, light, glow) and ambiguous noun forms (face, head, hand). The resulting inventory mixes *posture* (stand, sit, lean; 33% of the top-18 mass), *perception and gaze* (look, watch, gaze, stare, observe, glance; 26%), *dialogue* (speak, talk, listen; 19%), and *motion* (walk, turn, hold; 15%). The relative weight of posture/perception/dialogue (78%) over motion (15%) is a property worth noting because most prior video-generation benchmarks favor high-motion prompts [Huang et al., 2024b, 2025c] with a dedicated Dynamic Degree dimension that explicitly penalizes static videos, and dedicated motion benchmarks [Ling et al., 2025, Liu et al., 2024b] structure their entire prompt suite around motion patterns. EntityBench is complementary: with motion held subtle, the visual evaluation budget shifts to entity-level identity preservation, which is the consistency property our benchmark targets.

Scene-design tag rates. Lexical-tag coverage on action descriptions is uneven: shot type is named in 52.5% of descriptions (close-ups and extreme close-ups together account for 61% of those), indoor/outdoor in 24.5%, time of day in 19.2% (2.7× more night than day), and explicit visual-style tags in only 2.2%. We do not use these tags as inputs to any evaluation metric, as our concentration is on entity consistency. They are reported here as a profile of the prompt corpus.

A.5 Tier-Stratified Comparison

Table 10 extends Table 3 of the main paper with fine-grained per-tier statistics. It shows that EntityBench's difficulty axis isolates long-range memory burden specifically. Per-shot composition is essentially constant across tiers, such as, mean characters per shot, multi-character rate, and tri-type rate all vary by less than two percentage points from easy to hard. However, the long-range memory load scales sharply. From easy to hard, the mean per-entity max gap triples ($3.2 \rightarrow 9.7$ shots), the global maximum gap quadruples ($11 \rightarrow 48$), and the entity-slot re-appearance rate climbs from $\sim 58\%$ to $\sim 81\%$. This separation is by design: it evaluate methods that target long-range identity preservation against tier-level scaling without confounding from increased intra-shot complexity, which would be a separate and orthogonal failure mode.

B Evaluation Metrics

This section specifies every metric in our evaluation suite formally. We begin with notation in Section B.1, then detail each pillar in turn: intra-shot quality (Section B.2), cross-shot consistency (Section B.3), and intra-shot prompt-following alignment (Section B.4). Section B.5 describes the strict-mode reproducibility contract that governs all aggregations.

Table 10: Tier-stratified statistics. Rows that also appear in the main paper's Table 3 (scale, shots/episode, recurrence gap) are not duplicated here; this table reports the additional dimensions made available by the released annotations. "≥ 3 chars" is the fraction of shots scheduling at least three characters; "2c+1o" is the fraction with ≥ 2 characters and ≥ 1 object. "Re-appearance rate" is the fraction of entity-slot appearances that are re-appearances (Section A.3); "memory-only rate" is the stricter shot-level analog (a shot is memory-only iff *all* of its scheduled entities are re-appearances).

	Easy	**Medium**	**Hard**	**All**
Episodes	80	40	20	140
Shots	873	618	1,000	2,491
Scheduled entities (unique)	1,694	1,021	878	3,593
Per-entity reappearance (recurring entities only)				
Recurring-entity rate	53.3%	57.1%	61.5%	56.4%
Mean per-entity max gap	3.24	3.85	9.72	5.14
Median per-entity max gap	3.0	3.0	7.0	3.0
Global max gap	11	14	**48**	48
Per-shot composition				
Mean characters / shot	1.99	2.06	1.98	2.00
Max characters / shot	6	7	6	7
Frac. shots ≥ 3 chars	23.5%	27.0%	24.4%	24.7%
Frac. shots 2c+1o	55.4%	55.3%	52.7%	54.3%
Memory test signal				
+Entity-slot re-appearance rate	$\sim$57.7%	$\sim$64.5%	$\sim$**80.7%**	68.6%
Cut rate	54.3%	39.5%	41.8%	45.6%
Registry-shot rate	69.5%	62.0%	**37.3%**	54.7%
Memory-only rate (shot-level)	30.5%	38.0%	**62.7%**	45.3%

B.1 Notation

Episodes, scenes, and shots. An episode $E = (S_1, S_2, \ldots, S_K)$ is a sequence of K shots, each a video clip S_k of F_k frames at fixed resolution. Each shot belongs to a scene $\mathcal{S}$, and a scene cut at shot k is indicated by an attribute $\mathrm{cut}(k) \in \{\mathsf{True}, \mathsf{False}\}$, with $\mathrm{cut}(k) = \mathsf{True}$ when shot k begins a new scene and $\mathrm{cut}(k) = \mathsf{False}$ when shot k continues the previous shot. Frames of shot k are denoted $f_{k,1}, \ldots, f_{k,F_k}$.

Entity registry and schedule. Each episode is equipped with an entity registry $\mathcal{E} = \mathcal{E}^{\mathrm{char}} \cup \mathcal{E}^{\mathrm{obj}} \cup \mathcal{E}^{\mathrm{loc}}$ partitioned into characters, objects, and locations. Each entity $e \in \mathcal{E}$ has a textual description $\mathrm{desc}(e)$. The script associates each shot k with a scheduled entity set $\mathcal{E}_k \subseteq \mathcal{E}$ (the entities expected to appear in shot k) and an action description a_k (free-form text describing what happens in the shot).

Visual encoders. We use three frozen pretrained encoders throughout. Let $\phi_{\mathrm{DINO}} : \mathbb{R}^{H \times W \times 3} \to \mathbb{S}^{767}$ denote DINOv2-base [Oquab et al., 2023] CLS embeddings (unit-normalized, 768-dim sphere); $\phi^{\mathrm{img}}_{\mathrm{CLIP}}$ and $\phi^{\mathrm{txt}}_{\mathrm{CLIP}}$ denote CLIP ViT-B/32 [Radford et al., 2021] image and text embeddings respectively (jointly trained, 512-dim, unit-normalized). For an image x and text t, the CLIP text-image similarity is

$$\mathrm{CLIPsim}(x, t) = \phi^{\mathrm{img}}_{\mathrm{CLIP}}(x)^\top \phi^{\mathrm{txt}}_{\mathrm{CLIP}}(t) \in [-1, 1]. \tag{1}$$

Grounding. Let G denote the GroundingDINO[Liu et al., 2024a] detector with text encoder `bert-base-uncased`. For frame f and query q, $G(f, q)$ returns a (possibly empty) set of detections $\{(b_i, p_i)\}_i$ where $b_i \subset f$ is a bounding box (xyxy pixel coordinates) and $p_i \in [0, 1]$ is the model's confidence. We threshold detections at $\tau_{\mathrm{box}} = 0.25$ and the per-token text alignment at $\tau_{\mathrm{text}} = 0.20$. The crop operator $\mathrm{Crop}(f, b)$ extracts the pixel region inside b with a 10% padding margin, then resizes to 224×224 for embedding.

LLM judgement. Let M_{LLM} denote the multimodal LLM `gemini-2.5-pro` [Comanici et al., 2025], which we treat as an oracle returning structured JSON conditioned on a list of images and a

textual prompt: $M_{\text{LLM}}(\{x_1, \ldots, x_n\}, t) \to J$ where J is a parsed dictionary. Per-criterion scores returned on a 1–10 scale are normalized to $[0, 1]$ via $s \mapsto s/10$.

Aggregation conventions. For a list of values $V = (v_1, \ldots, v_n)$, we write $\text{mean}(V)$ for the sample mean, $\text{median}(V)$ for the median, and $\|V\|$ for the cardinality n. A value of None is excluded from any aggregation; if all values are None, the aggregate is also None (never substituted with 0). See Section B.5 for the formal contract.

Human validation of LLM judgement. Because every per-entity fidelity score (Pillar 2) and cross-shot identity score (Pillar 3) ultimately depends on LLM, we conducted a human-agreement study to verify that the LLM judge produces decisions consistent with human raters. We sampled (shot, entity) pairs uniformly across the four evaluated methods and across all three difficulty tiers, stratified to include equal numbers of character, object, and location instances, and balanced between gate-passing and gate-failing cases. For each sampled instance, 3 independent human annotators who are research scientists in generative AI, were shown the same canonical crop and registry description used by LLM, and asked to provide both the binary present/absent verdict and the per-criterion fidelity scores on the same 1–10 scale. For cross-shot identity, annotators received the same anchor-vs-each pairwise format described in Section B.4, with the binary same/different verdict as the primary outcome. We report agreement using Cohen's κ for the binary verdicts and Pearson's r for the continuous scores, computed both between LLM and the human majority vote and between individual human raters as an upper bound on achievable agreement. Across the 200 samples, LLM achieved κ = 0.93 on intra-shot presence, κ = 0.94 on cross-shot identity verdicts, falling within the inter-human range of κ = [0.8, 0.96]. Disagreement cases were concentrated in (i) half face distortion and (ii) blurry features with dim lighting. We treat these as inherent to the task rather than judge-specific failures. The agreement levels support the use of LLM as the operational judge throughout the benchmark, with the caveat that all reported metrics inherit a residual uncertainty bounded by the LLM–human gap.

B.2 Pillar 1: Intra-Shot Quality

Inspired by Huang et al. [2024b], we adopt six standard intra-shot quality metrics that capture whether each shot is technically well-formed in isolation. We drop the `background_consistency` metric from the original VBench suite because it measures within-shot CLIP cosine on consecutive frames, which is confounded by intentional camera motion: a pan or zoom of a stable background is incorrectly penalized as inconsistency. For our long-range multi-shot benchmark where camera motion is common, this metric may be inaccurate. The remaining six are computed per shot and averaged across the episode.

For shot S_k with frames $f_{k,1}, \ldots, f_{k,F_k}$:

Subject consistency (range $[0, 1]$):

$$\text{SC}(S_k) = \frac{1}{F_k - 1} \sum_{i=1}^{F_k - 1} \phi_{\text{DINO}}(f_{k,i})^\top \phi_{\text{DINO}}(f_{k,i+1}). \tag{2}$$

Measures stability of the dominant subject within the shot.

Temporal flickering (range $[0, 1]$):

$$\text{TF}(S_k) = 1 - \frac{1}{F_k - 1} \sum_{i=1}^{F_k - 1} \text{MAE}(f_{k,i}, f_{k,i+1}), \tag{3}$$

where MAE is the mean absolute pixel difference normalized to $[0, 1]$. Penalizes high-frequency flicker.

Motion smoothness (range $[0, 1]$): RAFT [Teed and Deng, 2020] optical flow is used to interpolate intermediate frames; $\text{MS}(S_k)$ is the mean reconstruction quality of the interpolation, with higher values indicating smoother apparent motion. Implementation follows Huang et al. [2024b].

Dynamic degree (range $[0, 1]$): the fraction of inter-frame pairs whose RAFT optical flow magnitude exceeds a threshold; penalizes static slideshow-like outputs.

Aesthetic quality (range $[0, 1]$):

$$\mathrm{AQ}(S_k) = \frac{1}{F_k} \sum_{i=1}^{F_k} \mathrm{MLP}_{\mathrm{LAION}}\Big(\phi^{\mathrm{img}}_{\mathrm{CLIP}}(f_{k,i})\Big), \tag{4}$$

where $\mathrm{MLP}_{\mathrm{LAION}}$ is the LAION aesthetic predictor head trained on human aesthetic ratings.

Imaging quality (range $[0, 100]$):

$$\mathrm{IQ}(S_k) = \frac{1}{F_k} \sum_{i=1}^{F_k} \mathrm{MUSIQ}(f_{k,i}), \tag{5}$$

where MUSIQ is the no-reference image quality predictor of Ke et al. [2021]. We report this on its canonical $[0, 100]$ scale rather than normalizing to $[0, 1]$, to maintain comparability with the broader literature.

Episode-level aggregation. For each Pillar 1 metric m, the episode-level value is the mean over admissible shots:

$$m(E) = \frac{1}{K} \sum_{k=1}^{K} m(S_k). \tag{6}$$

B.3 Pillar 2: Intra-Shot Prompt-Following Alignment

Pillar 2 measures, for each shot in isolation, three aspects of prompt-following. (i) entity presence: do the scheduled entities actually appear? (ii) per-entity fidelity: when an entity does appear, does it match its registry description? (iii) action fidelity: does the shot depict the action described in the script? All three sub-evaluations are built on a unified grounding pass, described next, that is also reused by Pillar 3 (Section B.4). The same canonical crop saved per (shot, entity) pair is the exact image used for fidelity judging *and* for cross-shot comparison. This ensures the audit chain from headline metric to underlying pixels is consistent, representing the review process of drilling into a cross-shot score for a specific entity sees exactly the crops that produce that score.

B.3.1 Unified grounding pass

For each shot S_k and each scheduled entity $e \in \mathcal{E}_k$, we compute a canonical crop $c^*(k, e)$ as follows. We sample $N_{\mathrm{frame}} = 5$ frames evenly across the shot. For each frame $f_{k,i}$ we run grounding $G(f_{k,i}, \mathrm{desc}(e))$ to obtain candidate detections, and for each detection we compute three quality components:

$$\alpha_{\mathrm{clip}}(f_{k,i}, b) = \mathrm{CLIPsim}\big(\mathrm{Crop}(f_{k,i}, b), \mathrm{desc}(e)\big), \tag{7}$$

$$\alpha_{\mathrm{sharp}}(f_{k,i}, b) = \sigma\left(\frac{\mathrm{LapVar}(\mathrm{Crop}(f_{k,i}, b)) - 100}{200}\right), \tag{8}$$

$$\alpha_{\mathrm{area}}(f_{k,i}, b) = \sigma\left(\frac{\mathrm{AreaPct}(b, f_{k,i}) - 2}{5}\right), \tag{9}$$

where LapVar is the variance of the Laplacian of luminance as a standard sharpness proxy, $\mathrm{AreaPct}$ is the bounding-box area as a percentage of frame area, and $\sigma(z) = (1 + e^{-z})^{-1}$ is the logistic. The composite selection score is the product

$$\alpha(f_{k,i}, b) = \alpha_{\mathrm{clip}}(f_{k,i}, b) \cdot \alpha_{\mathrm{sharp}}(f_{k,i}, b) \cdot \alpha_{\mathrm{area}}(f_{k,i}, b). \tag{10}$$

Among all (frame, detection) pairs for entity e in shot k, the canonical crop is the argmax:

$$c^*(k, e) = \mathrm{Crop}(f_{k,i^*}, b^*), \qquad (i^*, b^*) = \operatorname*{arg\,max}_{(i,b) \in G_k(e)} \alpha(f_{k,i}, b), \tag{11}$$

where $G_k(e) = \bigcup_{i=1}^{N_{\mathrm{frame}}} G(f_{k,i}, \mathrm{desc}(e))$ is the union of all detections for entity e across the sampled frames.

The selection score balances three quality aspects. Crops with high CLIP score but motion blur lose on sharpness; sharp and large but wrong-entity crops lose on CLIP; right-entity sharp but tiny crops lose on area. All three components must be high for the score to be high; a crop with any one near zero is rejected.

Presence status. Each canonical crop is assigned a tri-valued status:

$$\text{status}(k,e) = \begin{cases} \texttt{absent} & G_k(e) = \emptyset, \\ \texttt{weak} & G_k(e) \neq \emptyset \text{ and } \alpha_{\text{clip}}(c^*(k,e), \text{desc}(e)) < \tau_{\text{CLIP}}, \\ \texttt{present} & G_k(e) \neq \emptyset \text{ and } \alpha_{\text{clip}}(c^*(k,e), \text{desc}(e)) \geq \tau_{\text{CLIP}}, \end{cases} \tag{12}$$

with $\tau_{\text{CLIP}} = 0.20$. Under any model that fails to render the right entity, GroundingDINO either returns nothing (`absent`) or returns a hallucinated box rejected by CLIP (`weak`); only `present` appearances are confidently the scheduled entity.

B.3.2 Presence

For each entity type $\mathcal{T}$, the per-shot presence rate is the fraction of scheduled entities of that type that achieved status `present` in the shot:

$$\rho^{\mathcal{T}}(S_k) = \frac{|\{e \in \mathcal{E}_k \cap \mathcal{E}^{\mathcal{T}} : \text{status}(k,e) = \texttt{present}\}|}{|\mathcal{E}_k \cap \mathcal{E}^{\mathcal{T}}|}, \tag{13}$$

with the convention that $\rho^{\mathcal{T}}(S_k) = \text{None}$ when the denominator is zero (*i.e.*, the shot has no scheduled entities of this type). The episode-level metric is the mean over shots that scheduled at least one entity of the type:

$$\text{intra_character_presence}(E) = \text{mean}\big(\{\rho^{\text{char}}(S_k) : k \in [K],\ \rho^{\text{char}}(S_k) \neq \text{None}\}\big), \tag{14}$$

and analogously for intra_object_presence and intra_location_presence. Note that absent entities pull down the rate (*e.g.*, a shot scheduling 2 characters with only 1 detected contributes 0.5 to the mean), while shots scheduling no entities of a type are skipped rather than contributing 1.0, which would inflate the metric.

B.3.3 Per-entity fidelity

For each (shot, entity) pair with $\text{status}(k,e) \in \{\texttt{present}, \texttt{weak}\}$, we send the canonical crop $c^*(k,e)$ to M_{LLM} along with the entity's textual description:

$$J_{k,e} = M_{\text{LLM}}\big(\{c^*(k,e)\}, \pi_{\text{fid}}(\text{desc}(e), \mathcal{T}_e, \text{status}(k,e))\big). \tag{15}$$

Appearances with status `weak` are still scored, but the resulting fidelity values are flagged as low-confidence in the audit JSON because the underlying crop did not pass the CLIP threshold. The LLM returns an overall fidelity score $\phi(k,e) \in [0,1]$ and four per-criterion scores $\phi^j(k,e) \in [0,1]$ for $j \in \mathcal{J}^{\mathcal{T}_e}$, where the per-type criterion sets are:

$$\begin{aligned} \mathcal{J}^{\text{char}} &= \{\text{face},\ \text{hair},\ \text{clothing},\ \text{build}\}, \\ \mathcal{J}^{\text{obj}} &= \{\text{shape},\ \text{color_texture},\ \text{proportions},\ \text{details}\}, \\ \mathcal{J}^{\text{loc}} &= \{\text{layout},\ \text{color_mood},\ \text{landmarks},\ \text{perspective}\}. \end{aligned}$$

The same criterion sets are reused identically in Pillar 3 (Section B.4.3), enabling direct comparison between within-shot fidelity and cross-shot consistency on the same axes.

Episode-level aggregation. For each entity type $\mathcal{T}$ and shot S_k, the per-shot mean fidelity across that type's entities is

$$\bar{\phi}^{\mathcal{T}}(S_k) = \text{mean}\big(\{\phi(k,e) : e \in \mathcal{E}_k \cap \mathcal{E}^{\mathcal{T}},\ \phi(k,e) \neq \text{None}\}\big), \tag{16}$$

and the episode-level fidelity metric is the mean over shots:

$$\text{intra_face_fidelity}(E) = \text{mean}\big(\{\bar{\phi}^{\text{char}}(S_k) : k \in [K],\ \bar{\phi}^{\text{char}}(S_k) \neq \text{None}\}\big), \tag{17}$$

and analogously for intra_object_fidelity and intra_location_fidelity. Per-criterion metrics are defined identically with ϕ^j in place of ϕ:

$$\text{intra_face_}j(E) = \text{mean}\big(\{\bar{\phi}^{\text{char},j}(S_k) : k \in [K],\ \bar{\phi}^{\text{char},j}(S_k) \neq \textsf{None}\}\big), \tag{18}$$

where $\bar{\phi}^{\text{char},j}(S_k)$ is the per-shot mean of $\phi^j(k,e)$ over scheduled characters. This yields 5 metrics per entity type (one overall, four per-criterion), for 15 fidelity metrics in total.

The fidelity scores $\phi(k,e)$ from this section are reused by Pillar 3's cross-shot fidelity gate (Eq. 22, Section B.4.1).

B.3.4 Action fidelity

To evaluate whether shot S_k depicts its action description a_k, we construct a labeled multi-frame grid that explicitly resolves the visual identity of each subject in the action.

Labeled action grid. We sample 6 frames evenly across the shot. For each frame $f_{k,i}$ and each scheduled entity $e \in \mathcal{E}_k$ (characters and objects only; locations are omitted from the grid), we draw the bounding box of the highest-confidence detection from $G_k(e)$ on $f_{k,i}$, with a unique color assigned to entity e across the entire grid. The text label is the entity name. The 6 annotated frames are tiled into a 2×3 grid image A_k. The colored labeled boxes help identify the characters so that the LLM can then assess the directional language unambiguously.

LLM judgment. The grid is sent to M_{LLM} with a prompt parameterized by the action description and the labeling legend:

$$J_k^{\text{action}} = M_{\text{LLM}}\big(\{A_k\}, \pi_{\text{action}}(a_k, \text{legend})\big). \tag{19}$$

The LLM returns six values per shot:

$\text{ovr}_k \in [0,1]$	overall action-fidelity score,
$\text{dep}_k \in \{0,1\}$	binary verdict on whether the action is depicted,
$\text{ai}_k \in [0,1]$	subject identity: are the labeled boxes the right characters?
$\text{aa}_k \in [0,1]$	subject action: does the named subject perform the verb?
$\text{ao}_k \in [0,1] \cup \{\textsf{None}\}$	object interaction; None if no object referenced in a_k,
$\text{am}_k \in [0,1]$	motion quality: is motion natural across frames?

Episode-level aggregation. Each of the six action metrics is the mean over shots for which the corresponding value is not None:

$$\text{intra_action_overall}(E) = \text{mean}\big(\{\text{ovr}_k : k \in [K],\ \text{ovr}_k \neq \textsf{None}\}\big), \tag{20}$$

$$\text{intra_action_depicted}(E) = \text{mean}\big(\{\text{dep}_k : k \in [K],\ \text{dep}_k \neq \textsf{None}\}\big), \tag{21}$$

and similarly for intra_action_subject_identity, intra_action_subject_action, intra_action_object_interaction, and intra_action_motion_quality. The object-interaction metric in particular has a smaller denominator: only shots whose action description explicitly references an object contribute, since asking "did the action use the object correctly" is meaningless for actions like "[character 1] walks toward the door" that do not name an object. This brings the action sub-evaluation to 6 metrics, and Pillar 2 to 24 (3+15+6) metrics total.

B.4 Pillar 3: Cross-Shot Consistency

Pillar 3 measures whether scheduled entities maintain identical across the shots in which they appear. It is the core of EntityBench's evaluation for long-range cross-shot entity consistency. The pillar reuses the canonical crops $c^*(k,e)$ produced by the unified grounding pass in Pillar 2 (Section B.3.1), and reuses Pillar 2's per-shot fidelity scores $\phi(k,e)$ to admit only well-rendered appearances into the cross-shot pool. The pillar comprises three stages: (i) an admissibility gate built from the Pillar 2 fidelity scores, (ii) DINOv2-based metrics (Section B.4.2) that score each appearance against the appearance centroid, and (iii) LLM-based metrics (Section B.4.3) that score appearances pairwise against a centroid-representative anchor.

B.4.1 Cross-shot fidelity gate

Even `present` appearances may render the entity poorly. Without further filtering, a method that produces nearly-static frames (*e.g.*, the same low-quality rendering repeated) would be rewarded with high consistency. We prevent this with a fidelity gate keyed on Pillar 2's per-shot fidelity scores.

For each (shot, entity) pair, recall that $\phi(k, e) \in [0, 1] \cup \{\mathsf{None}\}$ is the intra-shot fidelity score from Pillar 2 (Section B.3.3). The cross-shot pool for entity e is defined as

$$\mathcal{C}(e) = \big\{c^*(k, e) : \text{status}(k, e) = \texttt{present} \text{ and } (\phi(k, e) \geq \tau_{\text{fid}} \text{ or } \phi(k, e) = \mathsf{None})\big\}, \tag{22}$$

with $\tau_{\text{fid}} = 0.5$. The disjunction with None ensures that appearances for which Pillar 2 could not be computed (e.g., LLM call failure) are admitted by default rather than silently dropped, with the fact that they bypassed the gate logged for audit. The number of gated-out appearances is recorded per episode in the auxiliary metric `_meta_cross_shot_gate`.

Fidelity-gate-corrected aggregation. The gate filters which instances enter cross-shot computation, but a method that fails the gate on most of its outputs should not benefit from being scored only on the few it passes. We therefore aggregate per-entity metrics with an *instance-weighted, gate-corrected mean* that treats gate-skipped and gate-failed instances as zero contributions.

Let $m \in \mathcal{M}^{\text{ent}}$ denote a per-entity metric (any metric in Pillars 2 and 3 except presence and Pillar 1 VBench dimensions). For each episode E, let $v_E^m \in [0, 1] \cup \{\mathsf{None}\}$ be the episode-level value of m, and let $n_E^{\text{eval},m}$, $n_E^{\text{skip},m}$, $n_E^{\text{fail},m}$ count the underlying entity-instances (per-shot pairs for intra-shot metrics, per-comparison pairs for cross-shot metrics, locations for scene metrics) that respectively (i) passed the gate and were scored, (ii) were dropped by the fidelity gate, and (iii) failed at the LLM-call or grounding step. The aggregated metric for a method across the benchmark is

$$\overline{m} \;=\; \frac{\sum_{E\,:\,v_E^m \neq \mathsf{None}} v_E^m \cdot n_E^{\text{eval},m}}{\sum_E \left(n_E^{\text{eval},m} + n_E^{\text{skip},m} + n_E^{\text{fail},m}\right)}. \tag{23}$$

The numerator weights each episode's score by how many gate-passing instances it contributed, so episodes with more recurring entities (which carry more cross-shot evidence) are weighted accordingly. The denominator includes *all* eligible instances across all benchmark episodes, so a method failing the gate on a hard episode is correctly penalized.

Equivalently, $\overline{m}$ can be written as $\text{rawmean}(m) \times \text{coverage}(m)$, where

$$\text{rawmean}(m) = \frac{\sum_E v_E^m \cdot n_E^{\text{eval},m}}{\sum_E n_E^{\text{eval},m}}, \tag{24}$$

$$\text{coverage}(m) = \frac{\sum_E n_E^{\text{eval},m}}{\sum_E \left(n_E^{\text{eval},m} + n_E^{\text{skip},m} + n_E^{\text{fail},m}\right)}. \tag{25}$$

We report $\overline{m}$ in the main results and report $\text{rawmean}(m)$ alongside $\text{coverage}(m)$ in Appendix F.2 for transparency. Pillar 1 VBench metrics, which are computed on every shot of every episode without gating, have $\text{coverage}(m) = 1$ by construction and so $\overline{m} = \text{rawmean}(m)$.

B.4.2 DINOv2-based metrics

For each entity e with $|\mathcal{C}(e)| \geq 2$, we compute its appearance centroid in DINOv2 embedding space:

$$\mathbf{c}_e = \text{normalize}\left(\frac{1}{|\mathcal{C}(e)|} \sum_{c \in \mathcal{C}(e)} \phi_{\text{DINO}}(c)\right), \qquad \text{normalize}(\mathbf{v}) = \mathbf{v}/\|\mathbf{v}\|_2. \tag{26}$$

The per-appearance similarity to the centroid is

$$s(c, e) = \phi_{\text{DINO}}(c)^\top \mathbf{c}_e \in [-1, 1] \quad \text{for } c \in \mathcal{C}(e). \tag{27}$$

Discussion: Why centroid rather than anchor. An anchor-based metric, comparing each appearance to a designated reference appearance, suffers from two problems. First, it depends on which appearance is chosen as anchor: if the chosen reference is a poor rendering, the entire entity is unfairly

penalized as the bad anchor pulls all per-appearance similarities down. Second, in keyframe-then-animate methods, the first appearance is often generated by a different pipeline branch (*e.g.*, T2I) than later appearances (*e.g.*, I2V); pinning the anchor to the first appearance systematically biases the metric. The centroid is the unique reference point invariant to ordering, and an outlier crop only drags the centroid by a factor of $1/N$ rather than dominating the comparison.

Episode-level aggregation. For entity type $\mathcal{T} \in \{\text{char}, \text{obj}\}$, the episode-level metric pools all per-appearance similarities across all entities of that type:

$$\text{cs_face}(E) = \text{mean}\left(\bigcup_{e \in \mathcal{E}^{\text{char}}} \{ s(c,e) : c \in \mathcal{C}(e),\ |\mathcal{C}(e)| \geq 2 \} \right), \tag{28}$$

and similarly for cs_object. This pooling means an entity that appears in N shots contributes N samples to the mean. This is reasonable as a character that appears 8 times usually matters more than one that appears 2 times, so is in for episode-level consistency.

We additionally record per-entity diagnostics in the audit JSON: the mean, minimum (worst-deviation appearance), maximum (representative appearance), and pairwise median similarities; the shot keys of the worst and most-representative appearances; and the full per-shot breakdown for failure analysis.

Cross-Shot transition boundary. For each continuation pair (S_k, S_{k+1}) where $\text{cut}(k+1) = \text{False}$, we compute the boundary similarity

$$\text{btrans}(k) = \phi_{\text{DINO}}(f_{k,F_k})^\top \phi_{\text{DINO}}(f_{k+1,1}) \tag{29}$$

between the last frame of the previous shot and the first frame of the next. The episode-level metric is

$$\text{cs_transition_boundary}(E) = \text{mean}\big(\{\text{btrans}(k) : \text{cut}(k+1) = \text{False}\}\big). \tag{30}$$

This measures motion continuity at scene-internal boundaries. Hard scene cuts ($\text{cut}(k+1) = \text{True}$) are excluded since discontinuity at scene boundaries is intentional.

Discussion: Why no DINOv2 location metric. A location bounding box necessarily includes the entire visible scene, including any foreground characters. Two location appearances that share the same background but with different foreground characters present will produce different DINOv2 embeddings, and the metric would penalize this as inconsistency. We therefore evaluate location consistency using only the LLM-based metrics (Section B.4.4), which can be instructed to better ignore foreground.

B.4.3 LLM-based metrics: characters and objects

For each entity e with $|\mathcal{C}(e)| \geq 2$, we select an anchor crop $c_{\text{anchor}}(e) \in \mathcal{C}(e)$ and compare it pairwise against each remaining appearance using M_{LLM}. The anchor is the centroid-representative crop:

$$c_{\text{anchor}}(e) = \underset{c \in \mathcal{C}(e)}{\arg\max}\ s(c,e), \tag{31}$$

i.e., the appearance whose DINOv2 embedding is closest to the entity's centroid. This anchor choice is principled in the same way the centroid metric is: it does not depend on shot order, and it does not systematically bias toward T2V outputs.

For each pair $(c_{\text{anchor}}(e), c)$ with $c \in \mathcal{C}(e) \setminus \{c_{\text{anchor}}(e)\}$, we query the LLM with both crops and the entity's textual description:

$$J_{e,c} = M_{\text{LLM}}\big(\{c_{\text{anchor}}(e), c\}, \pi_{\text{pair}}(\text{desc}(e), \mathcal{T})\big), \tag{32}$$

where π_{pair} is the pairwise prompt template parameterized by entity type $\mathcal{T}$. The LLM returns a JSON dictionary $J_{e,c}$ with a binary same/different verdict $\text{same}_{e,c} \in \{0,1\}$, an overall similarity score $\text{sim}_{e,c} \in [0,1]$, and four type-specific per-criterion scores $\text{crit}^j_{e,c} \in [0,1]$ for $j \in \mathcal{J}^{\mathcal{T}}$. The per-type criterion sets $\mathcal{J}^{\mathcal{T}}$ are identical to those used in Pillar 2 (Section B.3.3), enabling direct comparison.

Discussion: Why pairwise rather than set-based. An alternative is to send all $|\mathcal{C}(e)|$ appearances in a single LLM call and ask the model to identify outliers. We empirically found that when $|\mathcal{C}(e)|$ is large, set-based judging may produce unreliable counts. The model sometimes returns out-of-range or inaccurate indices. Pairwise judging reduces each LLM call to a clean binary decision ("are these two the same?") which the model handles more consistently.

Episode-level aggregation. Let $\mathcal{P}^{\mathcal{T}}(E)$ denote the multiset of all (anchor, comparison) pairs in episode E for entity type $\mathcal{T}$:

$$\mathcal{P}^{\mathcal{T}}(E) = \big\{(e, c) : e \in \mathcal{E}^{\mathcal{T}},\ |\mathcal{C}(e)| \geq 2,\ c \in \mathcal{C}(e) \setminus \{c_{\text{anchor}}(e)\}\big\}. \tag{33}$$

The Pillar 3 LLM metrics for characters are

$$\text{llm_face_accuracy}(E) = \text{mean}\big(\{\text{same}_{e,c} : (e, c) \in \mathcal{P}^{\text{char}}(E)\}\big), \tag{34}$$

$$\text{llm_face_mean_score}(E) = \text{mean}\big(\{\text{sim}_{e,c} : (e, c) \in \mathcal{P}^{\text{char}}(E)\}\big), \tag{35}$$

$$\text{llm_face_}j(E) = \text{mean}\big(\{\text{crit}^{j}_{e,c} : (e, c) \in \mathcal{P}^{\text{char}}(E),\ \text{crit}^{j}_{e,c} \neq \mathsf{None}\}\big), \tag{36}$$

for each $j \in \{\text{face}, \text{hair}, \text{clothing}, \text{build}\}$. The objects suite (`llm_object_*`) is defined identically with $\mathcal{P}^{\text{obj}}$ and $\mathcal{J}^{\text{obj}}$. This yields 6 metrics per entity type, including overall accuracy, overall mean score, four per-criterion scores, for a total of 12 metrics across characters and objects.

B.4.4 LLM-based metrics: locations

Locations are evaluated differently from characters and objects in two respects. First, location judging uses full frames rather than crops, with a prompt explicitly instructing the LLM to ignore foreground characters and focus on the depicted place. Second, different camera angles, distances, framings, partial views, and zoom levels of the same physical location may look completely different, and uses a chain-of-thought structure that forces the LLM to commit to a per-frame description of the location before making a similarity judgment. This per-frame identification step mitigates the failure mode where high cinematographic diversity (close-ups, wide shots, pans) is mistaken for location inconsistency under naive set-based judging.

Following the character pipeline, location judging is anchor-vs-each pairwise. For each location $\ell \in \mathcal{E}^{\text{loc}}$ with $|\mathcal{C}(\ell)| \geq 2$, we select an anchor shot $c^{\star}_{\ell} \in \mathcal{C}(\ell)$ (the centroid-representative appearance, see Section B.4.2). For each non-anchor shot $c \in \mathcal{C}(\ell) \setminus \{c^{\star}_{\ell}\}$, we sample at most $N_{\text{frames_per_set}} = 2$ sharpness-ranked full frames from each of the two shots, yielding ≤ 4 images per pairwise call. Let X^{c}_{ℓ} denote the resulting image set:

$$P^{c}_{\ell} = M_{\text{LLM}}\big(X^{c}_{\ell},\ \pi^{\text{loc}}_{\text{pair}}(\text{desc}(\ell))\big). \tag{37}$$

Each pairwise call returns a binary same-location verdict $\text{same}^{c}_{\ell} \in \{0, 1\}$, an overall similarity $\text{sim}^{c}_{\ell} \in [0, 1]$, and four per-criterion scores $\text{crit}^{c,j}_{\ell} \in [0, 1]$ for $j \in \mathcal{J}^{\text{loc}} = \{\text{layout, color_mood, landmarks, perspective}\}$.

Per-location aggregation. For each location, we aggregate across the $|\mathcal{C}(\ell)| - 1$ pairwise comparisons:

$$\text{allcons}_{\ell} = \prod_{c \neq c^{\star}_{\ell}} \text{same}^{c}_{\ell}, \tag{38}$$

$$\text{cons}_{\ell} = \text{mean}\big(\{\text{sim}^{c}_{\ell} : c \neq c^{\star}_{\ell}\}\big), \tag{39}$$

$$\text{crit}^{j}_{\ell} = \text{mean}\big(\{\text{crit}^{c,j}_{\ell} : c \neq c^{\star}_{\ell}\}\big). \tag{40}$$

Episode-level aggregation.

$$\text{llm_scene_accuracy}(E) = \text{mean}\big(\{\text{allcons}_{\ell} : \ell \in \mathcal{E}^{\text{loc}},\ |\mathcal{C}(\ell)| \geq 2\}\big), \tag{41}$$

$$\text{llm_scene_mean_score}(E) = \text{mean}\big(\{\text{cons}_{\ell} : \ell \in \mathcal{E}^{\text{loc}},\ |\mathcal{C}(\ell)| \geq 2\}\big), \tag{42}$$

$$\text{llm_scene_}j(E) = \text{mean}\big(\{\text{crit}^{j}_{\ell} : \ell \in \mathcal{E}^{\text{loc}},\ |\mathcal{C}(\ell)| \geq 2,\ \text{crit}^{j}_{\ell} \neq \mathsf{None}\}\big), \tag{43}$$

for each $j \in \mathcal{J}^{\text{loc}}$. This yields 6 location metrics, bringing the Pillar 3 LLM total to 18 and the Pillar 3 overall total to 21.

B.4.5 Gap-decay diagnostic

In addition to the 21 headline Pillar 3 metrics, we record a per-pair gap-decay dataset for diagnostic plotting. For each entity e with $|\mathcal{C}(e)| \geq 2$ and each ordered pair of distinct appearances $(c_i, c_j) \in \mathcal{C}(e) \times \mathcal{C}(e)$ with $i < j$ in shot order, we record the triple

$$\big(\text{gap} = |k_j - k_i|, \quad \text{sim} = \phi_{\text{DINO}}(c_i)^\top \phi_{\text{DINO}}(c_j), \quad \text{type} = \mathcal{T}_e\big), \tag{44}$$

where k_i, k_j are the shot indices of the two appearances. The dataset enables construction of the gap-vs-similarity curve for each method, which characterizes how identity drifts as recurrence distance increases. A flat curve indicates that consistency is maintained regardless of how far apart two appearances are; a falling curve indicates degradation with distance.

B.5 Implementation Details

Metric value type. Every metric value is a structured tuple

$$m = (v, n_{\text{eval}}, n_{\text{failed}}, n_{\text{skipped}}), \tag{45}$$

where $v \in [0, 1] \cup \{\mathsf{None}\}$ (or appropriate canonical range) is the headline value, n_{eval} counts the number of items that contributed to the aggregation, n_{failed} counts items where the underlying computation errored (e.g., LLM call failed), and n_{skipped} counts items legitimately excluded (e.g., entity appears in only one shot, so cross-shot pairing is undefined).

No silent-zero contract. A metric with $n_{\text{eval}} = 0$ is recorded as $v = \mathsf{None}$, never as $v = 0$. The contract distinguishes three distinct outcomes:

- All items succeeded: $v \in [0, 1]$ with $n_{\text{failed}} = n_{\text{skipped}} = 0$.
- Some items failed but the rest were valid: $v \in [0, 1]$ with $n_{\text{failed}} > 0$. The mean is taken only over successful items.
- No items contributed: $v = \mathsf{None}$ with $n_{\text{eval}} = 0$. The episode is excluded from the across-episode mean.

This avoids the common failure mode where a method that produces unevaluable outputs (e.g., crashed videos) artificially looks "perfect" or "terrible" because missing values are silently substituted with extreme defaults.

Run manifest. Each evaluation run produces a manifest JSON that records every model checkpoint with file fingerprints, every library version, every BENCHMARK_CONFIG value (thresholds, sampling counts, criterion sets, etc.), and the evaluator's git revision. Two runs whose manifests differ in any non-trivial field are flagged as not directly comparable, with the differences listed in machine-readable form. The fields excluded from the comparability check are limited to: `method_name`, `timestamp_utc`, `platform`, `n_llm_keys`.

Hyperparameters. The complete list of fixed hyperparameters in the canonical configuration is given in Table 11.

C EntityMem: Entity-Aware Context Management

C.1 Memory Bank Design

The memory bank stores visual and textual entity references that the video generation model retrieves at each shot. We explore a baseline for setting up an entity memory bank where we pre-generate all entity references before any video generation begins, so that each entity's visual identity is established once and reused consistently throughout the sequence. This avoids a failure mode common in autoregressive approaches, where references are extracted from previously generated outputs: distortions in early shots quietly enter the reference pool and compound in later shots.

Table 11: Hyperparameters used in the canonical evaluation. All values are recorded in the run manifest and locked across reported numbers.

Parameter	Value	Description
N_{frame}	5	frames sampled per shot for grounding
τ_{box}	0.25	GroundingDINO box confidence threshold
τ_{text}	0.20	GroundingDINO text alignment threshold
τ_{CLIP}	0.20	CLIP threshold for `present` status
τ_{fid}	0.50	cross-shot fidelity gate threshold
crop padding	10%	padding applied around bounding boxes
crop resolution	224×224	input size to encoders
action grid	2×3	frames per action evaluation
N_{shots} (loc. set)	8	max shots sampled for set-based location LLM judging
$N_{\text{frames_per_shot}}$ (loc. set)	2	frames per shot for location judging
DINOv2 model	`facebook/dinov2-base`	visual encoder for embeddings
CLIP model	`openai/clip-vit-base-patch32`	for text-image matching
Multimodal LLM	`gemini-2.5-pro`	for all judgment metrics

Per-entity references. The bank maintains both visual and textual references for each entity. On the visual side, each entity receives a reference tailored to its type. For characters, the reference is a segmented portrait showing a single character in isolation with the background removed, labeled with the character's name rendered as text at the bottom of the image. The labeling provides an explicit name-to-appearance mapping that helps the video backbone bind textual names to visual identities, particularly when multiple characters co-occur in a shot. For locations, a panoramic image is cropped into angle variants (left, center, right), giving the compositor a choice of camera-angle-aware backgrounds when assembling keyframes. For objects, a Classification Agent (§C.2) first determines whether the object requires a standalone visual reference at all: mobile props such as creatures or vehicles receive segmented portraits, while wearable items and scene fixtures are parts of character or location portraits. On the textual side, the bank extracts and stores a description of each entity at its first appearance, which can be retrieved when that entity recurs in a later shot.

Per-shot keyframes. For each shot, the bank also stores a keyframe composite showing the spatial arrangement of all scheduled entities against the location background. Unlike entity portraits, which are generated once and reused, keyframes are composed per shot from the pre-generated references. The Layout Agent (§C.2) plans each keyframe's composition: character positions on a discrete horizontal grid, camera angle selection, and (for continuation shots) reasoning about how camera panning shifts retained characters and where entering characters appear. When characters enter or exit mid-shot, the Layout Agent decomposes the shot into multiple keyframes. A compositor then height-normalizes the character portraits and places them at planned positions alongside any scheduled objects.

Consuming the memory bank. At generation time, the references for a given shot are assembled as an ordered sequence: per-character labeled portraits first, followed by the keyframe composites. The video backbone receives this sequence alongside a text prompt that describes the shot's camera direction, entity description, and actions. For recurring entities whose appearance descriptions do not appear in the current shot's script, the pipeline retrieves stored descriptions from the bank and injects them into the prompt, ensuring the video backbone has appearance guidance for every scheduled entity. For continuation shots, the last frame of the previous shot is provided to the video backbone as a separate first-frame input for temporal continuity, but is excluded from the memory bank to prevent it from overriding the curated entity references.

C.2 Agent-Based Context Management

The memory bank requires high-quality, complete content. Populating it requires a chain of context management decisions: determining what each entity needs as a reference, generating that reference, verifying its quality before it enters the bank, and arranging the bank's contents into per-shot keyframes for the video backbone. EntityMem delegates each of these decisions to a specialized agent, while deterministic operations such as image generation, segmentation, and compositing are handled by tools.

Classification Agent. Not every entity requires a pre-generated visual reference. The Classification Agent examines each entity in the story and determines its reference needs based on entity type and role: characters always receive portraits, locations receive panoramic backgrounds, and objects are evaluated individually. It distinguishes mobile props that need cross-shot visual consistency (creatures, vehicles, artifacts) from wearable items and scene fixtures that are parts of character or location references. This filtering step keeps the memory bank focused on entities that genuinely require visual anchoring.

Portrait Agent. For each entity that requires a visual reference, the Portrait Agent manages its generation. It gathers the entity's description, its first-appearance context from the story script, and the story overview to infer the visual style (e.g., anime, photorealistic). For characters and objects, it writes a generation prompt for a text-to-image tool, which produces N candidates on a chroma-key background. After a segmentation tool extracts the foreground of each candidate, the Portrait Agent evaluates the segmented results on a composite grid and selects the best one based on segmentation quality, composition, and body proportions. For locations, it generates a panoramic image and crops it into angle variants (left, center, right) to provide camera-aware backgrounds for keyframe composition.

Verification Agent. Before a portrait enters the memory bank, the Verification Agent inspects it for failure modes: incorrect or missing characteristic generation, or segmentation-related failures such as missing body regions, transparent clothing, or incompletely removed backgrounds. If verification fails, it triggers a retry with an alternative background color (e.g., magenta, blue) to improve segmentation contrast, addressing cases where character appearance blends with the original chroma-key color.

Layout Agent. Once the memory bank contains verified entity references, the Layout Agent translates each shot's narrative action into one or more keyframe layouts. Given the shot's action text, the entity schedule, and (for continuation shots) the full state of the previous shot, it determines how many keyframes the shot requires, which entities appear in each, their positions, and the camera angle. For static shots, a single keyframe captures the scene. When the action changes the spatial arrangement mid-shot, the agent produces multiple keyframes that capture the progression: for example, the first keyframe may show two characters in conversation, while the second introduces a third character arriving at a new position. For continuation shots, the agent simulates physical camera behavior: it reasons about which direction the camera should pan to accommodate the action, shifts retained characters' positions accordingly (e.g., a character previously on the right moves to the left as the camera pans right), and selects the matching angle variant from the location's panoramic crops. A compositor then realizes each layout by placing height-normalized portraits at the planned positions alongside any scheduled objects.

Tools. The agents above rely on three tools for execution: a text-to-image generator [Labs, 2024] that produces candidate portraits from agent-written prompts, a segmentation model [Ravi et al., 2024] that extracts foreground masks using entity-type-specific point prompt strategies, and a compositor that arranges segmented portraits onto location backgrounds at agent-specified positions.

D ENTITYBENCH: Data Examples

This section showcases two stories from ENTITYBENCH. For each example we present three classes of figure: an *overview* showing the story summary and the entity registry; an *entity-persistence strip* that visualizes which entities recur in which shots; and a *shot timeline* containing the verbatim per-shot `action_descriptions` text and the per-shot `entity_schedule` chips. The two representative examples are chosen. Section D.1 is a compact, single-location piece with a small cast and a clear within-scene continuation chain, while Section D.2 is a multi-location story whose principal character carries the same wardrobe and props across four distinct locations.

D.1 Example 1: single-location, ten-shot continuation

The first example is a short three-scene story with three characters, two locations (*The Scholar's Study* and *The Quiet Room*), and five recurring objects. It is built from a single hard-cut opening shot followed by a six-shot continuation chain in *The Scholar's Study*, a one-shot interlude in *The*

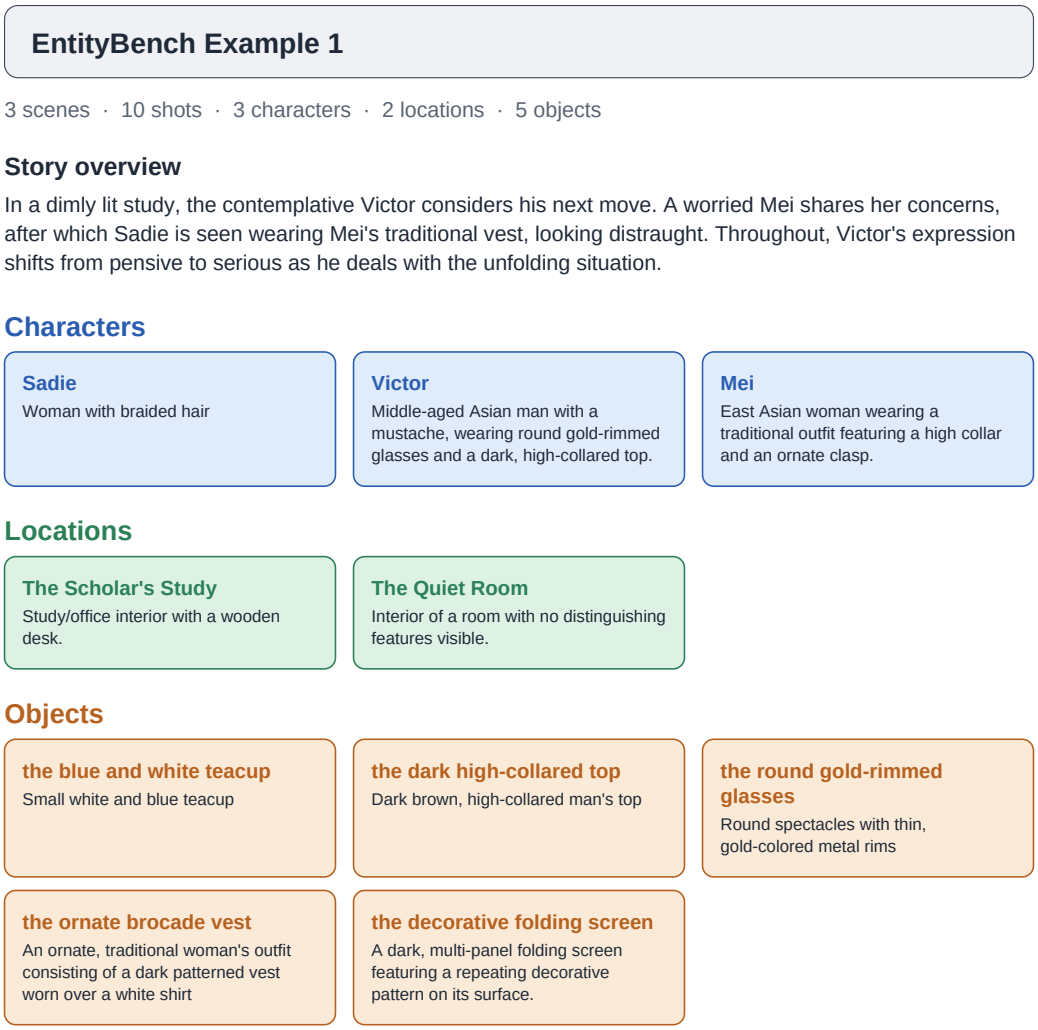


Figure 9: ENTITYBENCH Example 1: story overview and entity registry. The header reports the structural counts (scenes, shots, characters, locations, objects). The registry below, with chip color indicating entity type, is at the bottom.

Quiet Room, and a final three-shot continuation back in the study. Fig. 9 demonstrates the entity descriptions. Fig. 10 visualizes the persistence pattern across all ten shots. Fig. 11 shows each shot's action descriptions and entity schedule chips.

D.2 Example 2: multi-location, cross-scene entity persistence

The second example demonstrates the longer-range entity-persistence properties from ENTITYBENCH. The story spans six scenes and four locations (*The City Bus*, *The Old Stone Chapel*, *The Normandy Campaign Map*, and *The Interview Room*), and follows a single principal character across them. Two wearable objects (*the blue denim jacket* and *the white t-shirt*) recur in nearly every shot the principal appears in, providing a near-continuous wardrobe signal across all four locations. A location-bound prop, *the wooden church pew*, is reused only within *The Old Stone Chapel* (scenes 2 and 4), *the antique French letter*, is referenced only in two non-consecutive shots; the persistence strip in Fig. 13 makes both the dominant wardrobe-and-prop thread and the sparser narrative props visible at a glance. The two shot-timeline figures (Fig. 14–15) show fifteen shots in story order.

EntityBench Example 1 — entity persistence

Rows are entities in registry order; columns are shots in story order. A filled cell means the entity appears in that shot's entity_schedule. Solid vertical rules separate scenes; dashed rules mark within-scene hard cuts.

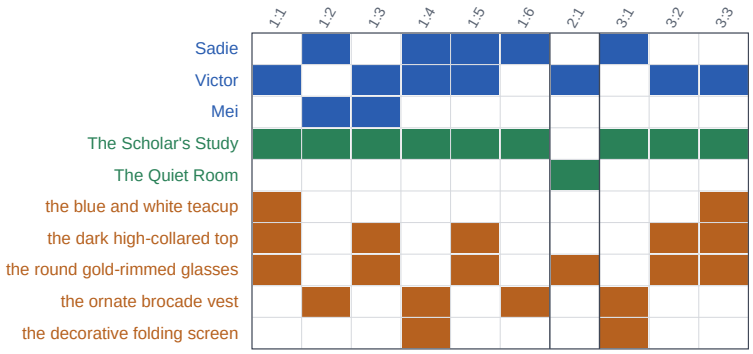


Figure 10: ENTITYBENCH Example 1: entity-persistence strip. Rows are entities in registry order and columns are shots in story order. A filled cell means the entity is scheduled in that shot. Solid vertical rules separate scenes; dashed rules mark within-scene hard cuts.

E Agent Prompts

This section provides the full text of every prompt used by the four EntityMem agents, including Classification, Portrait, Verification, and Layout agents. Variables filled in at runtime are typeset in italic blue (e.g. *{name}*). All other text is verbatim from our implementation.

E.1 Classification Agent

The Classification Agent decides whether each object entity warrants a pre-generated visual reference (a creature, vehicle, or recurring prop) or should be handled implicitly through the character portrait or location background (a garment, a piece of furniture, a fixture). Characters and locations are unconditionally classified for portrait and panoramic generation, so they bypass this prompt.

E.2 Portrait Agent

The Portrait Agent runs three distinct prompt-writing tasks, *i.e.*, one per entity type, followed by a single multi-image selection call that picks the best candidate after segmentation. Style inference (anime, photorealistic, 3D rendered, etc.) is deferred to the agent in every case.

Character portraits. For each character, the agent writes a tailored image generation prompt that preserves visual cues from the registry description while constraining the output to a single view on a chroma-key background suitable for SAM2 segmentation (Fig. 17).

Object portraits. Objects that pass the Classification Agent receive their own square-format prompt, again with style inferred from the story (Fig. 18).

Location panoramas. Locations are generated as a single ultra-wide (1536×512, 3:1) panorama and then deterministically cropped into left, center, and right variants by the compositor. The agent, therefore, writes one prompt specifying a wide establishing shot for panoramic view generation (Fig. 19). This helps address the consistency issues that plague three separately generated angle variants.

EntityBench Example 1 — shot timeline

Each row is one shot. Entity names in the action description are bolded in their type colour; the chip strip beneath each action lists the shot's entity_schedule. Hard cuts are flagged with a bold shot index and a tinted row background.

Shot	Action description (entity names tinted) and entity_schedule
1:1 *(cut)*	In **The Scholar's Study**, a close-up of **Victor** in the dimly lit room. He wears **the round gold-rimmed glasses** and **the dark high-collared top**. He holds **the blue and white teacup**, his expression serious and contemplative as he looks down in thought and then looks up slightly. Victor · The Scholar's Study · the round gold-rimmed glasses · the dark high-collared top · the blue and white teacup
1:2	In **The Scholar's Study**, a close-up of **Mei** with a worried expression as she speaks to **Sadie** in the dimly lit room. **Sadie** wears **the ornate brocade vest**. Sadie · Mei · The Scholar's Study · the ornate brocade vest
1:3	In **The Scholar's Study**, **Victor** wears **the round gold-rimmed glasses** and **the dark high-collared top** as he raises his head to speak to **Mei**. Victor · Mei · The Scholar's Study · the round gold-rimmed glasses · the dark high-collared top
1:4	In **The Scholar's Study**, **Sadie** adjusts the ornate collar piece of **the ornate brocade vest** she is wearing. In the background, **Victor** stands near **the decorative folding screen**, watching her. Sadie · Victor · The Scholar's Study · the ornate brocade vest · the decorative folding screen
1:5	In **The Scholar's Study**, a low-angle close-up of **Victor** and **Sadie**. **Victor**, wearing **the round gold-rimmed glasses** and **the dark high-collared top**, leans back, talking with a pensive expression as **Sadie** listens intently. Victor · Sadie · The Scholar's Study · the round gold-rimmed glasses · the dark high-collared top
1:6	A close-up of **Sadie** wearing **the ornate brocade vest**. She slowly looks down, her expression filled with sadness. Sadie · The Scholar's Study · the ornate brocade vest
2:1 *(cut)*	In **The Quiet Room**, a close-up of **Victor** wearing **the round gold-rimmed glasses**. He is talking, his facial expression transitioning from a slight smile to a serious, concerned look. Victor · The Quiet Room · the round gold-rimmed glasses
3:1 *(cut)*	In **The Scholar's Study**, a close-up of **Sadie** wearing **the ornate brocade vest**. She has a worried expression on her face as she speaks, looking slightly downwards, with **the decorative folding screen** in the background. Sadie · The Scholar's Study · the ornate brocade vest · the decorative folding screen
3:2	In **The Scholar's Study** at night, a frontal close-up of **Victor**. **Victor**, wearing **the dark high-collared top** and **the round gold-rimmed glasses**, has a pensive expression as he speaks. Victor · The Scholar's Study · the dark high-collared top · the round gold-rimmed glasses
3:3	**Victor** lowers his head to drink from **the blue and white teacup**. Victor · The Scholar's Study · the round gold-rimmed glasses · the dark high-collared top · the blue and white teacup

Figure 11: ENTITYBENCH Example 1: shot timeline. Each row is one shot. The verbatim `action_descriptions` with every entity that the shot's `entity_schedule` references, bolded and tinted in its type color. Hard cuts are flagged with bold shot indices and a tinted row background.

Best-candidate selection. After the image generator [Labs, 2024] produces $N=5$ candidates per entity and SAM2 [Ravi et al., 2024] segments each one, the Portrait Agent calls the LLM [Comanici et al., 2025] on a side-by-side grid of the segmented candidates rendered on a checkered background (Fig. 20). The checkered background makes mask artifacts visible, which are otherwise less conspicuous with a solid grey fill.

E.3 Verification Agent

After selection, the Verification Agent inspects the chosen segmented portrait for the failure modes that defeat downstream compositing: missing body regions, see-through clothing, etc. A failed

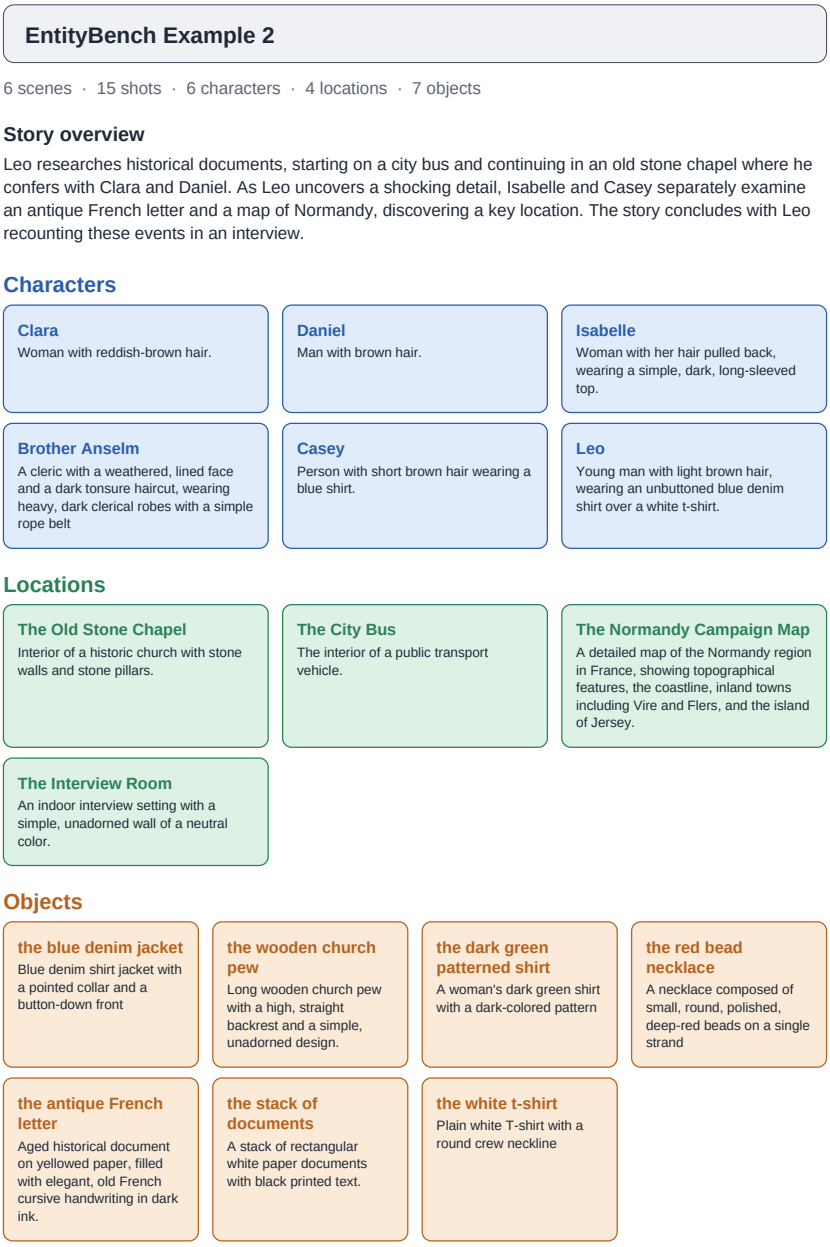


Figure 12: ENTITYBENCH Example 2: story overview and entity registry.

verification triggers a retry with an alternative chroma-key color, addressing the common case where a part of the foreground matches the original green key (Fig. 21).

E.4 Layout Agent

The Layout Agent is context-dependent. For each shot, it receives the action text, the entity schedule, and the previous shot's character positions and camera angle if the shot is a continuation. It returns a structured plan of one or more keyframes, each with the participating entities, their positions on a discrete 7-cell horizontal grid, and the camera angle (front/left/right) to use as background. The prompt explicitly walks the agent through camera-pan reasoning so that characters retained across a continuation translate the correct way as the camera moves. The prompts are illustrated across two

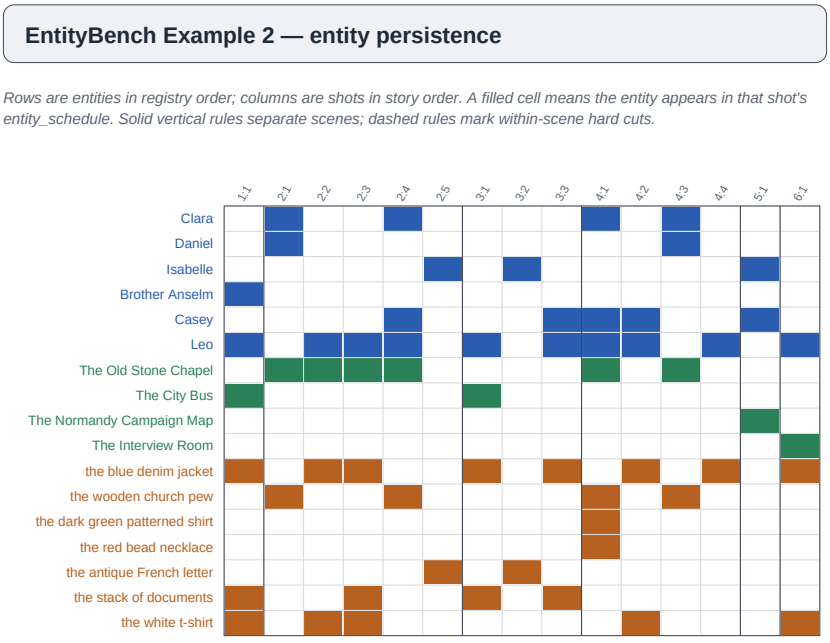


Figure 13: ENTITYBENCH Example 2: entity-persistence strip.

figures: Fig. 22 contains the inputs and the global task rules, and Fig. 23 contains the camera-pan reasoning, hard-cut handling, and output schema.

F Additional Experimental Results

F.1 EntityBench Evaluation: Full 51-metric Results

This appendix reports the complete EntityBench evaluation suite across all 51 metrics for the four methods compared in the main paper. Numbers are fidelity-gate-corrected means following the convention defined in §3.2 (formal definition in Appendix B.4.1). Tables 12, 13, and 14 report Pillars 1, 2, and 3 respectively. **Bold** marks the column winner per row. The 12-metric subset highlighted in main-paper Table 4 is identified by an asterisk (*).

Reading the tables. For Pillar 1 (VBench [Huang et al., 2024b]), `imaging_quality` is reported on its native MUSIQ scale of $[0, 100]$; all other Pillar 1 metrics and all metrics in Pillars 2 and 3 are bounded in $[0, 1]$. Pillar 2 organizes per-entity scores by entity type (characters, objects, locations) and includes action correctness as a separate sub-pillar. Pillar 3 organizes cross-shot consistency scores by signal source: DINOv2 embedding similarity, then LLM-judged identity for characters, objects, and scenes (the latter using the camera-invariant pairwise prompt described in Appendix B.4.4).

Per-method coverage. Methods produce evaluable outputs at different rates. Per-metric coverage fractions are reported in Appendix F.2. The fidelity-gate-corrected means in this appendix already incorporate coverage by treating gate-skipped instances as zero contributions; raw means without gate correction are also tabulated in Appendix F.2.

F.2 Per-method Coverage and Raw Means

The fidelity-gate-corrected means in the main paper (Table 4) and Appendix F.1 aggregate as $\overline{m} = \text{rawmean}(m) \times \text{coverage}(m)$, where coverage is the fraction of eligible (shot, entity) instances that pass the fidelity gate (Equation 22). This appendix decomposes the corrected means into their two components for transparency.

EntityBench Example 2 — shot timeline (part 1/2)

Each row is one shot. Entity names in the action description are bolded in their type colour; the chip strip beneath each action lists the shot's entity_schedule. Hard cuts are flagged with a bold shot index and a tinted row background.

Shot	Action description (entity names tinted) and entity_schedule
1:1 *(cut)*	Close-up of **Leo**, wearing **the blue denim jacket** and **the white t-shirt**, sitting in a seat next to **Brother Anselm** in **The City Bus**. He holds **the stack of documents**, his eyes scanning the top page as a content smile slowly forms on his face. Leo · Brother Anselm · The City Bus · the stack of documents · the blue denim jacket · the white t-shirt
2:1 *(cut)*	Over-the-shoulder medium shot inside **The Old Stone Chapel**. **Clara** sits on **the wooden church pew** with a pensive expression, speaking earnestly to **Daniel**, whose back is to the camera. Clara · Daniel · The Old Stone Chapel · the wooden church pew
2:2	Close-up of **Leo**, wearing **the blue denim jacket** and **the white t-shirt**, in **The Old Stone Chapel**. He looks down at something just out of frame, and a warm smile spreads across his face. Leo · The Old Stone Chapel · the blue denim jacket · the white t-shirt
2:3	Close-up of **Leo**'s face inside **The Old Stone Chapel**. Wearing **the blue denim jacket** and **the white t-shirt**, he smiles and then looks down at **the stack of documents**, held in his hands just out of frame. Leo · The Old Stone Chapel · the blue denim jacket · the white t-shirt · the stack of documents
2:4	Over-the-shoulder shot from behind **Casey** in **The Old Stone Chapel**. **Clara** sits in **the wooden church pew** next to **Leo**, listening intently as a slight, thoughtful smile forms on her face. Clara · Casey · Leo · The Old Stone Chapel · the wooden church pew
2:5	Extreme close-up of **Isabelle**'s finger tracing down a column of handwritten names on **the antique French letter**. The finger pauses on several entries that share the same surname, finally stopping on the name Casey. Isabelle · the antique French letter
3:1 *(cut)*	Close-up of **Leo**, wearing **the blue denim jacket**, sitting in **The City Bus**. He holds **the stack of documents**, looking down and speaking as he reads from them. Leo · The City Bus · the blue denim jacket · the stack of documents
3:2	Extreme close-up on **the antique French letter**. **Isabelle**'s finger enters the frame and points decisively at one of the names on the list. Isabelle · the antique French letter
3:3	Close-up on **Leo**, wearing **the blue denim jacket**, holding a single document from **the stack of documents**. He looks down at the paper, his eyes widening in shock as he reads. **Casey** stands beside him, watching his reaction. Leo · Casey · the blue denim jacket · the stack of documents

Figure 14: ENTITYBENCH Example 2: shot timeline, part 1 of 2. Each row is one shot; the verbatim `action_descriptions` text appears with every entity that the shot's `entity_schedule` references bolded and tinted in its type color. Hard cuts are flagged with bold shot indices and a tinted row background.

What coverage measures. For each per-entity metric, coverage answers a different question: For Pillar 2 fidelity (intra-shot) and Pillar 3 DINOv2, the fidelity gate is applied at the embedding-similarity level rather than as a hard rejection, so all eligible instances enter the pool with coverage $= 1$ for these metrics. However, for Pillar 3 cross-shot LLM, coverage is the fraction of (anchor, comparison) pairs where the gate admits both appearances and the LLM call completes successfully. Low coverage indicates a method whose entity renderings often fail intra-shot fidelity, leaving few admissible appearances for cross-shot comparison. A method's coverage on Pillar 3 LLM metrics thus principally reflects *intra-shot rendering fidelity*, because a method that fails the gate frequently has fewer pairs available for cross-shot judgment, and the corrected mean correctly penalizes this because gate failure is a method failure, not an evaluation artifact.

EntityBench Example 2 — shot timeline (part 2/2)

Each row is one shot. Entity names in the action description are bolded in their type colour; the chip strip beneath each action lists the shot's entity_schedule. Hard cuts are flagged with a bold shot index and a tinted row background.

Shot	Action description (entity names tinted) and entity_schedule
4:1 *(cut)*	Over-the-shoulder shot from behind **Casey**, showing **Leo** and **Clara** sitting beside them on **the wooden church pew** in **The Old Stone Chapel**. **Clara**, wearing **the dark green patterned shirt** and **the red bead necklace**, turns toward **Leo** and smiles as she talks. **Leo** and **Casey** watch her as she speaks. Clara · Leo · Casey · The Old Stone Chapel · the wooden church pew · the dark green patterned shirt · the red bead necklace
4:2	Close-up of **Leo**, wearing **the blue denim jacket** and **the white t-shirt**, smiling happily as he talks to **Casey** just off-screen. Leo · Casey · the blue denim jacket · the white t-shirt
4:3	Over-the-shoulder close-up from behind **Daniel**, looking at **Clara**. They are seated together on **the wooden church pew** inside **The Old Stone Chapel**. **Clara** has a sad expression while talking. Clara · Daniel · The Old Stone Chapel · the wooden church pew
4:4	Close-up of **Leo**, wearing **the blue denim jacket**. He listens with a thoughtful expression, then his gaze shifts downward. A slight smile appears on his face as he begins to speak. Leo · the blue denim jacket
5:1 *(cut)*	Close-up on **The Normandy Campaign Map**, where **Casey** and **Isabelle** are leaning over it. **Casey**'s finger traces a path starting near Vire and Flers, moves up along the coast, and finally points directly at the island of Jersey. Casey · Isabelle · The Normandy Campaign Map · The Normandy Campaign Map
6:1 *(cut)*	Close-up of **Leo**, wearing **the blue denim jacket** and **the white t-shirt**, in **The Interview Room**. He smiles warmly and talks directly to an off-camera interviewer. Leo · The Interview Room · the blue denim jacket · the white t-shirt

Figure 15: EntityBench Example 2: shot timeline, part 2 of 2 (continuation of Fig. 14).

Why EntityMem wins the corrected LLM metrics despite a slightly lower raw score. The most informative entries in Table 15 are the Pillar 3 LLM rows. On `llm_face_accuracy`, StoryMem's raw mean (0.718) is slightly higher than EntityMem's (0.678). It indicates when StoryMem manages to produce two gate-passing appearances of the same character, the LLM judges them roughly correctly. But StoryMem's coverage is only 0.31, vs. 0.60 for EntityMem: nearly half as many appearances pass the gate, leaving correspondingly fewer pairs to evaluate. The fidelity-gate-corrected means are therefore $0.678 \times 0.60 = 0.407$ for EntityMem vs. $0.718 \times 0.31 = 0.222$ for StoryMem—a $1.83\times$ advantage that derives entirely from EntityMem's better intra-shot rendering rate. The same pattern, slightly muted, applies to `llm_object_accuracy`: StoryMem's raw mean is higher (0.699 vs. 0.522), but neither method has high coverage on objects (0.29 vs. 0.31), so the corrected means are closer (0.203 vs. 0.162). This decomposition validates the fidelity-gate-corrected aggregation as the appropriate metric for evaluating cross-shot generators.

Coverage on cs_transition_boundary. The `cs_transition_boundary` metric measures continuity at scene-internal cuts and is computed only for shot pairs with detectable matched content. Coverage is near-1 for all methods except StoryMem (0.83), which fails to produce detectable

Classification Agent · Object Reference Necessity

```
Classify this object for a video generation pipeline.

Object name: {name}
Object description: {description}
Story context: {story_overview}

Should this object be PRE-GENERATED as a standalone reference image for
compositing into scenes?

Answer YES only if ALL of these are true:
1. It is a distinct, mobile entity (creature, pet, vehicle, handheld prop)
2. It needs to look consistent across shots (visual identity matters)
3. It is NOT clothing/wearable (shirts, hats, shoes, accessories worn on body)
4. It is NOT part of the environment (furniture, fixtures, structures that belong
to a specific location — bookshelves, chalkboards, perches, platforms, poles,
walls, decorations)

Examples:
- "the yellow electric pokémon" (Pikachu-like creature) → YES, distinct creature
- "the classroom bookshelves" → NO, part of classroom environment
- "the red and yellow perch" → NO, a structure/fixture in a location
- "the green chalkboard" → NO, classroom fixture
- "the striped t-shirt" → NO, clothing
- "the group of small pokémon" → YES, distinct creatures that need consistency
- "a magical sword" → YES, recurring handheld prop

Respond with ONLY JSON: {"generate": true/false, "reason": "brief"}
```

Figure 16: Prompt used by the Classification Agent to decide whether an object entity requires a pre-generated standalone reference. The agent receives the object name, description, and a story overview. A negative classification routes the object's appearance into either the owning character's portrait prompt or the location background.

Table 12: **Pillar 1: Intra-shot quality** (6 VBench dimensions). `imaging_quality` on $[0, 100]$; others on $[0, 1]$.

Metric	Ours	StoryMem	HoloCine	CineTrans
subject_consistency*	0.881	0.759	0.860	**0.968**
temporal_flickering	0.976	0.838	0.957	**0.979**
motion_smoothness*	0.988	0.849	0.964	**0.990**
dynamic_degree	0.657	0.562	**0.721**	0.688
aesthetic_quality*	0.593	0.475	0.518	**0.596**
imaging_quality*	66.00	56.41	49.97	**68.57**

continuity content on roughly 17% of in-scene boundaries; the corrected mean penalizes this gap, dropping StoryMem's raw 0.795 to a corrected 0.660 and reversing the ranking against EntityMem.

Pillar 1 and action metrics. Pillar 1 VBench dimensions and Pillar 2 action metrics are computed on every shot of every episode without an admission gate, so coverage is uniformly 1.00 and raw equals corrected. We omit those rows from this appendix; their values appear in Tables 12 and 13.

F.3 Per-metric Effect Sizes (EntityMem vs. StoryMem)

This appendix reports Cohen's d for each of the 51 metrics in the head-to-head between EntityMem and its backbone StoryMem. d is reported with pooled-variance (the more common Cohen's d, used for between-groups comparison) and as d_z (the paired-samples variant, $d_z = \text{mean}(\Delta)/\text{sd}(\Delta)$, more appropriate when the same episodes are evaluated under both methods). Both are reported because

Portrait Agent · Character Portrait Prompt

SYSTEM

```
You are an expert at writing image generation prompts for character portraits.

Given a character's description, the story context, and their first appearance
scene, write a prompt that will generate a clear character portrait suitable for
use as a visual reference in video generation.

Rules:
- INFER the visual style from the story context and character description. If the
description mentions "anime style", "animated", or the story involves anime-like
elements, write an anime-style prompt. If it describes realistic
clothing/settings, write a photorealistic prompt.
- Include ALL visual details from the description: clothing, hair, accessories,
skin tone, build, etc.
- The output must show EXACTLY ONE character, standing upright, facing the
camera. Show from head to at least the knees.
- Normal body proportions (no chibi, no super-deformed).
- Solid bright green background (#00FF00 chroma key green). This is critical —
the background MUST be bright green for segmentation.
- No text, no annotations, no extra views.
- Do NOT add details not mentioned in the description.
- Keep the prompt under 60 words. CLIP truncates at 77 tokens.
- ALWAYS include "full body" in the prompt.
- NEVER use "reference sheet", "turnaround", "model sheet", or "concept art" —
these trigger multi-view layouts.

Respond with ONLY the prompt text, no explanation, no quotes, no markdown.
```

USER

```
Story overview: {story_overview}

Character name: {name}
Character description: {description}
First appearance context: {first_appearance}

Write a portrait generation prompt for this character.
```

Figure 17: Prompt used by the Portrait Agent to write a character-specific prompt. The first-appearance context is the registry line from the shot in which the character is introduced. On a verification failure (Fig. 21), the chroma-key color in the agent's output is rewritten to magenta, blue, or orange before the next Flux invocation to improve segmentation contrast.

their values diverge slightly under our pairing structure. Per the convention in Table 5, we lead with pooled d in the main paper. Positive values are where EntityMem contributes. Δ is the raw mean difference (EntityMem minus StoryMem) computed on episodes where both methods produced an evaluable score. n_{paired} is the number of such episodes.

Largest effects. The largest single-metric advantage for EntityMem is `intra_face_fidelity` at $d = +2.33$ ($\Delta = +0.262$, $n = 139$). Five additional metrics exceed $d > +1.0$, all in character-related categories: `intra_face_clothing` ($d = +1.94$), `intra_face_hair` ($d = +1.81$), `intra_face_face` ($d = +1.66$), `intra_face_build` ($d = +1.51$), and `intra_character_presence` ($d = +1.23$). The largest deficit is `llm_object_accuracy` at $d = -0.68$, with five additional cross-shot object metrics in the range $d \in [-0.68, -0.53]$. The DINOv2 cross-shot metrics show $d \in [-0.66, -0.40]$, but as noted in Table 18's caption, this disagrees with LLM identity judgment on the same episodes; we discuss this disagreement in Appendix F.4.

Portrait Agent · Object Portrait Prompt

SYSTEM

```
You are an expert at writing image generation prompts for object reference
images.

Given an object description and story context, write a prompt that will generate
a clear, centered image of the object suitable for compositing into video frames.

Rules:
- INFER the visual style from the story context. Anime story → anime style
object. Realistic story → photorealistic object.
- The object should be centered and occupy most of the frame.
- Solid bright green background (#00FF00 chroma key green) for segmentation. No
text, no annotations, no people.
- Show the object from a clear front/side angle.
- Include ALL visual details from the description: colors, patterns, markings.
- Keep the prompt under 50 words.

Respond with ONLY the prompt text, no explanation, no quotes, no markdown.
```

USER

```
Story overview: {story_overview}

Object name: {name}
Object description: {description}

Write an image generation prompt for this object.
```

Figure 18: Prompt used by the Portrait Agent for objects that the Classification Agent flagged as needing a standalone reference.

Portrait Agent · Location Panoramic Prompt

```
Story overview: {story_overview}

Location name: {name}
Location description: {description}

Write a prompt for a WIDE PANORAMIC establishing shot of this location. INFER the
visual style from the story (anime, photorealistic, 3D, etc) and START the prompt
with the style. The image should be very wide, showing the full room/space from
wall to wall. Full environment, ultra-wide angle, no people. Under 50 words.
```

Figure 19: Prompt used by the Portrait Agent to write a panoramic-shot image generation prompt for each location. The agent's single prompt drives a 1536×512 generation that is then cropped into left/center/right thirds by the compositor.

F.4 Trade-offs and Limitations

Embedding similarity rewards uniformity, not identity. EntityMem's largest negative effects concentrate on DINOv2 cross-shot similarity ($d = -0.50$) and on LLM-pairwise object metrics ($d = -0.60$). These metrics measure different things, and their disagreement on character faces is diagnostic of *what each metric actually rewards*. For example, the episode script describes *long-haired blonde girl in a fur-collared coat* in Figure 24. EntityMem and StoryMem reach near-identical DINOv2 face similarity (`cs_face` $= 0.883$ vs. 0.875), yet the LLM-pairwise rater identifies zero of StoryMem's face pairs as the same character (`llm_face_accuracy` $= 0.000$) vs. 71% for EntityMem (0.714). DINOv2 cs_face rewards low-fidelity, generic facial renderings that cluster

Portrait Agent · Best-Candidate Selection

```
You are selecting a CHARACTER PORTRAIT for {name} — {description}. The image
shows {N} candidates labeled #0 to #{N-1} side by side. Each candidate has been
segmented (background removed) and is shown on a checkered pattern. The checkered
pattern shows where the background was removed — a CLEAN segmentation has smooth
edges with no checkered pattern bleeding into the character's body or hair.

Pick the candidate with the BEST COMBINATION of these criteria (segmentation
quality is the MOST important):
1. CLEAN SEGMENTATION — smooth, artifact-free edges around hair and clothing. No
jagged edges, no speckle noise, no holes in the body. REJECT candidates with
visible checkered pattern showing through hair or clothing.
2. Shows EXACTLY ONE view of the character (reject
multi-view/turnaround/reference sheet layouts)
3. Shows most of the character's body (more visible = better, but a clean
half-body is better than a glitchy full-body)
4. Normal body proportions (not chibi or super-deformed)
5. Character identity is clear (face, hair, clothing recognizable)

IGNORE any text/watermarks. Respond with ONLY JSON: {"best_index": N, "reason":
"brief"}
```

Figure 20: Vision-language prompt used by the Portrait Agent to select the best of N segmented candidates from a single side-by-side grid image. A single multi-image call replaces N independent quality calls and compares candidates directly. The same prompt is reused (with "CHARACTER PORTRAIT" replaced by the relevant entity type) for object selection.

Verification Agent · Segmentation Quality Check

```
This is a segmented character portrait of {name} ({description}) shown on a
checkered background. The checkered pattern shows where the background was
removed (transparent areas).

Check for these problems:
1. Is most of the character's body MISSING (only head visible, or large holes)?
2. Is the clothing TRANSPARENT when it shouldn't be (can you see checkers through
clothes)?
3. Is the background NOT removed (character has a solid rectangle around them)?

Respond with ONLY JSON: {"pass": true/false, "reason": "brief explanation"}
```

Figure 21: Prompt used by the Verification Agent to gate portraits before they enter the memory bank. A failure marks the candidate's chroma-key color as "contaminated" and triggers regeneration with the next backup color (magenta → blue → orange) up to two retries.

tightly in embedding space without depicting the named character. The LLM, asked an identity-verification question, sees through the mode collapse. Where the two metrics disagree, the LLM is the more honest signal of what a downstream user would notice. This disagreement is not asymptotic. Across all 4 methods, character cs_face decays as a function of recurrence gap (number of shots between appearances), but the rate of decay differs sharply.

Per-entity bank limitations on objects. Object presence is comparable across methods (`intra_object_presence` $= 0.884$ for EntityMem vs. 0.906 for StoryMem), both methods detect roughly the same number of objects, with StoryMem slightly ahead. The cross-shot object gap ($d = -0.60$) is therefore not a denominator artifact: the LLM identifies more genuine inconsistencies in EntityMem's object renderings. We attribute this to the "sticker look" produced by per-entity

Layout Agent · Per-Shot Keyframe Layout (1/2)

You are a scene composition director for anime/video keyframe generation.

We composite character portraits onto a location background. You plan the spatial layout for each keyframe in a video shot. We have 3 location background angles: front, left, right (panoramic crops of the same room). You decide which angle to use.

=== CONTEXT ===

Action text: *{action_text}*
Characters: *{char_list}*
Objects: *{obj_list}*
Location: *{location_desc}*
Visual style: *{visual_style}*
Shot type: *{shot_type}*
{prev_state_block}

=== TASK ===

Plan ALL keyframes for this shot. For each keyframe, specify which characters are visible and their positions.

Position options: far-left, left, center-left, center, center-right, right, far-right

Rules:
- 1 keyframe for static shots, 2-3 for character entries/exits
- Characters not listed in a keyframe's entities are NOT shown
- Order entities from back to front (furthest → nearest to camera)
- For 2nd+ keyframe, note what changed (removed, added)
- CRITICAL: Only include characters already on-screen in kf_0. Characters who "enter", "walk in", or "arrive" MUST appear for the first time in kf_1 or later — NEVER in kf_0.
- Every entity name MUST exactly match one of the character names listed above
- End each description with the visual style tag (e.g. "*{visual_style}*")

(continued in Part 2: continuation-shot rules, hard cuts, camera angle, and output schema)

Figure 22: Layout Agent prompt, part 1 of 2: input fields and the global task rules. *{prev_state_block}* is empty for hard-cut shots; for continuations, it contains the previous shot's characters, their positions, camera angle, and last keyframe description, along with explicit lists of retained, entering, and leaving characters. The position vocabulary (*far-left* → *far-right*) matches a 7-cell discrete grid that the compositor maps to pixel coordinates after height-normalizing the segmented portraits.

Layout Agent · Per-Shot Keyframe Layout (2/2)

```
(continued from Part 1)

=== CONTINUATION SHOT RULES ===
For continuation shots (cut=False), camera panning changes where characters
appear in frame. Think about it like a real camera:

When the camera PANS RIGHT:
  - A character who was on the RIGHT of the frame slides to the LEFT (camera
moved past them)
  - New characters entering from the right appear on the RIGHT
  - Use camera_angle="right"

When the camera PANS LEFT:
  - A character who was on the LEFT slides to the RIGHT
  - New characters entering from the left appear on the LEFT
  - Use camera_angle="left"

Decision process for continuation shots:
1. Look at where the retained character was positioned (e.g. "right")
2. New characters need to enter from the side with more space
3. The camera pans TOWARD the new characters
4. The retained character's position FLIPS (right → left, left → right)
5. Place new characters on the side the camera panned toward

Example: Previous shot had CharA at "right" with camera_angle="front". New shot
introduces CharB and CharC. The camera should pan right to make room →
camera_angle="right". CharA moves to "left" or "center-left". CharB and CharC
enter at "center-right" and "right".

If the retained character was at "center", keep them at "center" and use "front"
angle. Place new characters on either side.

=== HARD CUT SHOTS ===
For hard cut shots, default to camera_angle="front" unless the action text
suggests a specific direction.

=== CAMERA ANGLE ===
Choose which location background angle to use:
- "front": default center view of the location
- "left": camera looking to the left side of the room/space
- "right": camera looking to the right side of the room/space

=== OUTPUT ===
Respond with ONLY JSON:
{
  "reasoning": "Explain your camera angle choice and character placement...",
  "camera_angle": "front",
  "keyframes": [
    {
      "keyframe_index": 0,
      "description": "Scene description. {visual_style}.",
      "removed": [],
      "added": ["Name"],
      "location_changed": false,
      "entities": [{"entity": "Name", "position": "center-left"}]
    }
  ]
}
```

Figure 23: Layout Agent prompt, part 2 of 2: continuation-shot reasoning, hard-cut defaults, the camera-angle vocabulary, and the JSON output schema. The agent's `camera_angle` choice determines which of the location's three panoramic crops the compositor uses as the background; its `entities` list names the characters to render in each keyframe and their positions on the 7-cell horizontal grid.

Table 13: **Pillar 2: Intra-shot prompt-following alignment** (24 metrics). Per-entity scores aggregate over all (shot, entity) instances passing the fidelity gate.

Metric	Ours	StoryMem	HoloCine	CineTrans
Presence (3 metrics)				
intra_character_presence*	**0.967**	0.849	0.882	0.796
intra_object_presence*	0.888	**0.893**	0.723	0.776
intra_location_presence	**0.687**	0.681	0.624	0.651
Character fidelity (5 metrics)				
intra_face_fidelity*	**0.740**	0.452	0.349	0.327
intra_face_face	**0.607**	0.424	0.369	0.366
intra_face_hair	**0.684**	0.485	0.482	0.413
intra_face_clothing	**0.802**	0.504	0.339	0.378
intra_face_build	**0.726**	0.539	0.449	0.521
Object fidelity (5 metrics)				
intra_object_fidelity*	0.601	**0.618**	0.267	0.384
intra_object_shape	**0.712**	0.701	0.373	0.508
intra_object_color_texture	0.691	**0.709**	0.331	0.480
intra_object_proportions	**0.728**	0.715	0.383	0.539
intra_object_details	0.573	**0.598**	0.256	0.371
Location fidelity (5 metrics)				
intra_location_fidelity*	**0.555**	0.504	0.306	0.428
intra_location_layout	**0.603**	0.529	0.354	0.474
intra_location_color_mood	**0.706**	0.627	0.474	0.588
intra_location_landmarks	**0.562**	0.522	0.305	0.429
intra_location_perspective	**0.557**	0.520	0.346	0.488
Action correctness (6 metrics)				
intra_action_overall*	**0.618**	0.547	0.569	0.273
intra_action_depicted	**0.519**	0.446	0.458	0.124
intra_action_subject_identity	**0.706**	0.595	0.606	0.478
intra_action_subject_action	**0.697**	0.626	0.695	0.323
intra_action_object_interaction	**0.781**	0.712	0.616	0.346
intra_action_motion_quality	0.716	0.723	**0.772**	0.528



Figure 24: DINOv2 similarity measures consistency in a different way from LLM.

object compositing and potential less focus in the pretrain data. Even when objects are provided as conditions, the video generation models tend to fail include or generate it consistently, indicating the challenge in object consistency.

Visual quality is not entity consistency. CineTrans wins all four highlighted VBench dimensions (`subject_consistency` 0.97, `imaging_quality` 68.6) yet has the lowest character presence (0.80) and the worst LLM character accuracy (0.09) of all four methods. A method can render beautifully and still fail to depict the right characters. The two are orthogonal axes. Pixel-quality benchmarks

Table 14: **Pillar 3: Cross-shot consistency** (21 metrics). DINOv2 metrics use centroid-anchor cosine similarity; LLM metrics use anchor-vs-each pairwise judgment with type-specific criteria. Scenes use the camera-invariant pairwise prompt (Appendix B.4.4).

Metric	Ours	StoryMem	HoloCine	CineTrans
DINOv2 embedding similarity (3 metrics)				
cs_face*	0.737	**0.792**	0.751	0.772
cs_object*	0.798	**0.839**	0.803	0.794
cs_transition_boundary*	**0.738**	0.663	0.498	0.508
LLM characters (6 metrics)				
llm_face_accuracy*	**0.406**	0.226	0.228	0.091
llm_face_mean_score*	**0.426**	0.234	0.242	0.145
llm_face_face	**0.381**	0.216	0.223	0.145
llm_face_hair	**0.447**	0.248	0.282	0.175
llm_face_clothing	**0.464**	0.241	0.242	0.143
llm_face_build	**0.489**	0.260	0.285	0.217
LLM objects (6 metrics)				
llm_object_accuracy*	0.164	**0.203**	0.088	0.092
llm_object_mean_score*	0.202	**0.222**	0.094	0.145
llm_object_shape	0.232	**0.239**	0.104	0.180
llm_object_color_texture	0.235	**0.243**	0.104	0.190
llm_object_proportions	0.238	**0.244**	0.105	0.195
llm_object_details	0.184	**0.209**	0.087	0.124
LLM scenes — camera-invariant pairwise (6 metrics)				
llm_scene_accuracy	0.309	**0.398**	0.304	0.119
llm_scene_mean_score*	0.659	**0.671**	0.616	0.432
llm_scene_layout	**0.697**	0.684	0.641	0.449
llm_scene_color_mood	0.716	**0.724**	0.669	0.619
llm_scene_landmarks	0.603	**0.637**	0.563	0.346
llm_scene_perspective	**0.727**	0.696	0.713	0.467

Table 15: **Per-method raw means and coverage fractions** for representative per-entity metrics. Each cell shows raw mean / coverage.

Metric	Ours	StoryMem	HoloCine	CineTrans
Pillar 2: Intra-shot fidelity (coverage = 1.00 by design)				
intra_face_fidelity	**0.740 / 1.00**	0.452 / 1.00	0.349 / 1.00	0.327 / 1.00
intra_object_fidelity	0.601 / 1.00	**0.618 / 1.00**	0.267 / 1.00	0.384 / 1.00
intra_location_fidelity	**0.555 / 1.00**	0.504 / 1.00	0.306 / 1.00	0.428 / 1.00
Pillar 3: Cross-shot DINOv2 (coverage ≈ 1.00; transition is gated)				
cs_face	0.737 / 1.00	**0.792 / 1.00**	0.751 / 1.00	0.771 / 1.00
cs_object	0.798 / 1.00	**0.839 / 1.00**	0.802 / 1.00	0.794 / 1.00
cs_transition_boundary	**0.738 / 1.00**	0.795 / 0.83	0.509 / 0.98	0.508 / 1.00
Pillar 3: Cross-shot LLM (gate-corrected)				
llm_face_accuracy	**0.678 / 0.60**	0.718 / 0.31	0.536 / 0.43	0.266 / 0.34
llm_object_accuracy	0.522 / 0.31	**0.699 / 0.29**	0.592 / 0.15	0.296 / 0.31

do not predict entity consistency, and EntityBench's three-pillar structure is designed precisely to surface this distinction.

F.5 Long-range Identity Stability: Gap-Decay Analysis

EntityMem explores whether per-entity memory bank can maintain character identity over long-range recurrence. We test this by binning adjacent-appearance pairs of the same character by their *gap*. The number of intervening shots are between two appearances. We report per-bin mean similarity for each method.

Table 16: **Pillar 1: Intra-shot quality.** VBench dimensions are computed on every shot of every episode, so $n_{\text{paired}} = 140$ for all rows. `imaging_quality` Δ and d are computed on the native MUSIQ scale of $[0, 100]$.

Metric	Δ	d	d_z	n_{paired}
subject_consistency	$+0.122$	$+1.18$	$+1.13$	140
temporal_flickering	$+0.137$	$+1.27$	$+1.21$	140
motion_smoothness	$+0.139$	$+1.30$	$+1.24$	140
dynamic_degree	$+0.095$	$+0.43$	$+0.39$	140
aesthetic_quality	$+0.118$	$+1.04$	$+0.97$	140
imaging_quality	$+9.59$	$+1.21$	$+1.13$	140

Table 17: **Pillar 2: Intra-shot prompt-following alignment.** Effect sizes computed at the episode level. Per-entity sub-metrics inherit the fidelity-gate convention from Pillar 2's overall metric.

Metric	Δ	d	d_z	n_{paired}
Presence				
intra_character_presence	$+0.097$	$+1.23$	$+0.97$	139
intra_object_presence	-0.021	-0.24	-0.18	138
intra_location_presence	-0.012	-0.05	-0.04	139
Character fidelity				
intra_face_fidelity	$\mathbf{+0.262}$	$\mathbf{+2.33}$	$+2.06$	139
intra_face_face	$+0.183$	$+1.66$	$+1.49$	139
intra_face_hair	$+0.199$	$+1.81$	$+1.62$	139
intra_face_clothing	$+0.298$	$+1.94$	$+1.74$	139
intra_face_build	$+0.187$	$+1.51$	$+1.36$	139
Object fidelity				
intra_object_fidelity	-0.053	-0.41	-0.37	138
intra_object_shape	$+0.012$	$+0.10$	$+0.09$	138
intra_object_color_texture	-0.018	-0.13	-0.12	138
intra_object_proportions	$+0.013$	$+0.09$	$+0.08$	138
intra_object_details	-0.025	-0.20	-0.18	138
Location fidelity				
intra_location_fidelity	$+0.017$	$+0.11$	$+0.10$	140
intra_location_layout	$+0.074$	$+0.49$	$+0.45$	140
intra_location_color_mood	$+0.078$	$+0.65$	$+0.59$	140
intra_location_landmarks	$+0.039$	$+0.27$	$+0.24$	140
intra_location_perspective	$+0.037$	$+0.31$	$+0.28$	140
Action correctness				
intra_action_overall	$+0.043$	$+0.33$	$+0.30$	139
intra_action_depicted	$+0.074$	$+0.42$	$+0.38$	139
intra_action_subject_identity	$+0.110$	$+0.71$	$+0.64$	139
intra_action_subject_action	$+0.040$	$+0.33$	$+0.30$	139
intra_action_object_interaction	$+0.069$	$+0.50$	$+0.45$	138
intra_action_motion_quality	-0.008	-0.07	-0.06	139

We compute gap-decay for two cross-shot signals: DINOv2 cosine similarity (face embeddings, the same signal as `cs_face`) and LLM identity similarity (per-pair scores from the anchor-vs-each pairwise judge, the same signal as `llm_face_mean_score`).

We compare EntityMem against the two holistic baselines (HoloCine, CineTrans). StoryMem is excluded from this analysis: its strict fidelity gate admits only $\sim$1.3 LLM pairs per episode (180 pairs total across the benchmark, vs. EntityMem's 502 and HoloCine's 155), and these surviving pairs are systematically the easy cases where character identity is unambiguous. Per-bin estimates from such a heavily-gated, selection-biased subset are unstable. The aggregate EntityMem-vs-StoryMem comparison, which folds together coverage and per-pair quality, is reported in Table 4 (corrected) and Table 15 (decomposed). Per-bin numbers for StoryMem are tabulated in Table 21 for completeness but should not be interpreted as a directional comparison.

Table 18: **Pillar 3: Cross-shot consistency.** Effect sizes computed at the episode level on the paired subset.

Metric	Δ	d	d_z	n_{paired}
DINOv2 embedding similarity				
cs_face	-0.043	-0.66	-0.62	129
cs_object	-0.031	-0.43	-0.39	121
cs_transition_boundary	-0.052	-0.40	-0.36	130
LLM characters				
llm_face_accuracy	-0.026	-0.11	-0.10	129
llm_face_mean_score	-0.017	-0.10	-0.09	129
llm_face_face	-0.037	-0.19	-0.17	129
llm_face_hair	-0.014	-0.08	-0.07	129
llm_face_clothing	$+0.006$	$+0.03$	$+0.03$	129
llm_face_build	-0.001	-0.00	-0.00	129
LLM objects				
llm_object_accuracy	-0.208	-0.68	-0.62	121
llm_object_mean_score	-0.142	-0.64	-0.58	121
llm_object_shape	-0.112	-0.53	-0.48	121
llm_object_color_texture	-0.100	-0.53	-0.48	121
llm_object_proportions	-0.112	-0.57	-0.51	121
llm_object_details	-0.152	-0.63	-0.56	121
LLM scenes (camera-invariant pairwise)				
llm_scene_accuracy	-0.097	-0.29	-0.26	140
llm_scene_mean_score	-0.032	-0.16	-0.13	140
llm_scene_layout	$+0.006$	$+0.03$	$+0.02$	140
llm_scene_color_mood	-0.031	-0.17	-0.13	140
llm_scene_landmarks	-0.052	-0.24	-0.20	140
llm_scene_perspective	$+0.013$	$+0.07$	$+0.05$	140

Table 19: **LLM face identity similarity by gap distance**, comparing EntityMem against the two holistic baselines.

Method	Gap 1–2	Gap 3–5	Gap 6–10	Gap 11–20	Gap 21–50
EntityMem	**0.744** (250)	**0.698** (126)	**0.646** (76)	**0.669** (36)	**0.657** (14)
HoloCine	0.765 (82)	0.517 (36)	0.614 (22)	0.420 (15)	—
CineTrans	0.371 (51)	0.408 (25)	0.333 (6)	0.600 (10)	0.457 (7)

On the LLM identity metric (Table 19), EntityMem's score declines by only 0.075 from gap 1-2 to gap 11-20, and remains essentially flat (0.66-0.67) thereafter. HoloCine declines by 0.345 over the same range and falls below EntityMem at every gap distance beyond 1-2 shots, with the gap widening to $+0.249$ at gap 11-20. The DINOv2 measurement (Table 20) shows little gap effect for any method, as DINOv2 cosine similarity reflects identity differently, discussed in Appendix F.4.

For completeness we include StoryMem's per-bin numbers in Table 21, with the caveat that these are computed on a heavily-gated subset (StoryMem admits only 22% of the comparison pairs that EntityMem admits) and are not directly comparable to EntityMem's broader pool. The EntityMem-vs-StoryMem comparison should be read at the aggregate level, where the corrected metric in Table 4 accounts for coverage.

F.6 Per-tier Performance: Long-range Robustness

EntityBench is structured into three difficulty tiers based on episode length: easy (80 episodes, ≤ 14 shots each), medium (40 episodes, 14–30 shots), and hard (20 episodes, 50 shots each). Hard-tier episodes test long-range character recurrence specifically. We use this tier structure to ask whether EntityMem's advantage at the aggregate level reflects short-range performance or scales to long sequences.

Table 20: **DINOv2 face similarity by gap distance.** DINOv2 cosine similarity (mean of adjacent-pair sims to centroid) shows little gap effect across methods, consistent with embedding-similarity rewarding visual self-similarity rather than identity preservation.

Method	Gap 1–2	Gap 3–5	Gap 6–10	Gap 11–20	Gap 21–50
EntityMem	0.729 (376)	0.712 (80)	0.752 (27)	0.708 (6)	0.761 (1)
HoloCine	0.760 (124)	0.720 (26)	0.762 (11)	0.795 (5)	—
CineTrans	0.760 (61)	0.751 (41)	0.761 (7)	0.773 (2)	0.799 (3)

Table 21: **Full gap-decay table including StoryMem,** for completeness. StoryMem's per-bin estimates are based on a small, gate-selected subset and are not directly comparable across methods at the bin level.

Signal	Method	Gap 1-2	Gap 3-5	Gap 6-10	Gap 11-20	Gap 21-50
LLM	EntityMem	0.744 (250)	0.698 (126)	0.646 (76)	0.669 (36)	0.657 (14)
	StoryMem	0.830 (105)	0.759 (51)	0.763 (16)	0.950 (6)	0.650 (2)
	HoloCine	0.765 (82)	0.517 (36)	0.614 (22)	0.420 (15)	—
	CineTrans	0.371 (51)	0.408 (25)	0.333 (6)	0.600 (10)	0.457 (7)
DINOv2	EntityMem	0.729 (376)	0.712 (80)	0.752 (27)	0.708 (6)	0.761 (1)
	StoryMem	0.807 (113)	0.782 (26)	0.733 (6)	—	0.808 (1)
	HoloCine	0.760 (124)	0.720 (26)	0.762 (11)	0.795 (5)	—
	CineTrans	0.760 (61)	0.751 (41)	0.761 (7)	0.773 (2)	0.799 (3)

Character-related metrics: robustness improves with sequence length. Across the three character-centric metrics in Table 22, the head-to-head gap between EntityMem and StoryMem (the strongest baseline) is stable or grows as episodes get longer.

The pattern is sharpest on `intra_action_overall`: at easy tier the gap is $+0.028$ (essentially tied), but at hard tier it grows to $+0.146$ ($5\times$ larger). Inspection of the underlying numbers shows that EntityMem's score is roughly flat across tiers ($0.618 \rightarrow 0.626 \rightarrow 0.614$) while StoryMem drops by 21% from easy to hard ($0.590 \rightarrow 0.468$). On 50-shot episodes, where actions span longer, more diverse sequences with more potential for character drift, EntityMem's per-entity bank actively prevents the action-correctness collapse that other methods suffer.

A similar but smaller-magnitude pattern holds on `intra_character_presence`: the gap grows from $+0.077$ to $+0.155$, with EntityMem remaining essentially flat (0.96–0.97) while StoryMem drops from 0.88 to 0.81. On longer episodes, baselines start failing to render scheduled characters at all in some shots; EntityMem continues to render them.

LLM identity advantage compounds with longer sequences. On `llm_face_accuracy`, EntityMem's score itself *grows* with tier difficulty: $0.344 \rightarrow 0.393 \rightarrow 0.476$ (a 38% improvement from easy to hard). StoryMem also improves ($0.197 \rightarrow 0.306$), but EntityMem's growth is larger in absolute terms. We interpret this as a measurement effect: hard-tier (50-shot) episodes provide more pairs for the LLM to judge, and EntityMem's higher gate pass-rate (Appendix F.2) translates into a larger admitted pool where the LLM can identify successfully-preserved character identity.

Object trade-off shrinks at hard tier. The aggregate `llm_object_accuracy` loss to StoryMem ($d = -0.68$, Appendix F.3) is concentrated at easy tier (-0.073) and shrinks substantially at hard tier (-0.022, essentially tied). EntityMem's object accuracy itself grows with tier ($0.101 \rightarrow 0.169 \rightarrow 0.244$), suggesting that directly applying object entity bank is challenging to improve object consistency, but may help more when backbone's performance weakens.

DINOv2 deficit stays flat across tiers. In contrast, the `cs_face` gap to StoryMem is uniformly -0.04 to -0.05 across all three tiers, with *no* sequence-length effect. The flat DINOv2 gap, combined with a tier-dependent LLM gap on the same characters, supports the §4.3 / Appendix F.4 interpretation that DINOv2 cross-shot measures generic visual similarity rather than character identity. This is the gap-decay observation (Appendix F.5) reproduced at the tier level: same finding, different aggregation.

Table 22: **Per-tier breakdown of headline metrics**. EntityMem's advantage on character-related and action metrics is robust across all tiers, with several metrics showing the gap *widening* at hard tier (50-shot episodes). DINOv2 cross-shot face shows a flat gap across tiers, consistent with the embedding-vs-identity disagreement (Appendix F.4).

Metric	Tier	Ours	StoryMem	HoloCine	CineTrans
intra_character_presence	Easy	**0.961**	0.884	0.896	0.818
	Medium	**0.974**	0.836	0.845	0.788
	Hard	**0.968**	0.813	0.894	0.781
intra_face_fidelity	Easy	**0.738**	0.483	0.373	0.332
	Medium	**0.748**	0.438	0.315	0.314
	Hard	**0.736**	0.420	0.349	0.330
intra_action_overall	Easy	**0.618**	0.590	0.588	0.262
	Medium	**0.626**	0.563	0.585	0.256
	Hard	**0.614**	0.468	0.543	0.293
cs_face (DINOv2)	Easy	0.787	**0.822**	0.796	0.812
	Medium	0.744	**0.789**	0.765	0.792
	Hard	0.693	**0.746**	0.713	0.735
llm_face_accuracy	Easy	**0.344**	0.197	0.188	0.058
	Medium	**0.393**	0.220	0.223	0.042
	Hard	**0.476**	0.306	0.272	0.169
llm_object_accuracy	Easy	0.101	**0.174**	0.055	0.051
	Medium	0.169	**0.210**	0.074	0.119
	Hard	**0.244**	0.267	0.144	0.123

Table 23: **EntityMem-vs-StoryMem head-to-head gap by tier** on selected metrics. Positive values favor EntityMem.

Metric	Easy	Medium	Hard	Direction
intra_character_presence	$+0.077$	$+0.138$	$+0.155$	grows with tier
intra_face_fidelity	$+0.255$	$+0.310$	$+0.316$	grows slightly
intra_action_overall	$+0.028$	$+0.063$	$+0.146$	grows substantially
llm_face_accuracy	$+0.147$	$+0.173$	$+0.170$	grows then plateaus
llm_object_accuracy	-0.073	-0.041	-0.022	deficit shrinks
cs_face (DINOv2)	-0.035	-0.045	-0.053	flat

The hard tier contains only 20 episode, so per-tier estimates have wider confidence intervals than the aggregate. The directional claims above (gap-grows-with-tier, deficit-shrinks-at-hard) are robust to this in the sense that the differences exceed plausible standard errors at $n = 20$, but precise hard-tier values should be treated as approximate. The tier breakdown is intended primarily as a sanity check on the aggregate story, not as the basis for new claims.

G Additional Related Work

Identity-Preserving Video Generation. Maintaining consistent character appearance within generated video has been approached through frequency-domain identity decomposition [Yuan et al., 2025b], multi-subject reference conditioning [Liu et al., 2025b, Jiang et al., 2025], reinforcement learning with identity-aware rewards [Meng et al., 2025a], and training-free cross-shot feature sharing [Singh et al., 2025]. Other works explore zero-shot identity animation [He et al., 2024] and universal identity-preserving synthesis [Zhong et al., 2025]. However, these methods focus primarily on human facial identity for one or two subjects, leaving broader entity types, such as objects, locations, and character ensembles, largely unaddressed.

LLM-Directed Video Generation. LLMs have been used as video planners to produce scene descriptions with entity layouts and consistency groupings [Lin et al., 2023], decompose prompts into structured shot instructions [Chen et al., 2025], and combine coarse scene planning with fine-grained

object-level layout control [Wang et al., 2026]. Multi-agent frameworks further coordinate specialized modules for long video planning [Xie et al., 2024, Huang et al., 2025a]. These methods demonstrate the value of LLM-guided planning for structural coherence but treat entity consistency as a byproduct of shared embeddings or layout constraints. Our multi-agent system differs by maintaining persistent per-entity memory banks for both visual and textual information and injecting the retrieved entity memory as context for consistent cross-shot video generation.

H Broader Impact

EntityBench and EntityMem operate in cross-shot long video generation with both clear creative-tool applications and well-documented misuse risks. We discuss both, along with limitations of the proposed evaluation framework.

Beneficial applications. Reliable character consistency in multi-shot generation lowers the barrier for creators (independent animators, educators, accessibility advocates) to produce longer-form visual narratives without large production teams. Narrative video that preserves character identity across shots is a precondition for accessible storytelling, educational content with recurring characters, and rapid prototyping in animation and storyboarding.

Misuse risks. The same capabilities enable the generation of synthetic videos depicting real people in fabricated scenarios, with applications including non-consensual deepfakes, defamation, and political disinformation. EntityMem is built on pretrained text-to-video backbone, LLM, text-to-image generation model and inherits any safety properties of that backbone.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The dataset and evaluation details are in Section 3.1 and Appendix F.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations are discussed in Section F.4.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: This paper does not include theoretical results.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The data curation and evaluation details are provided in Section 3 and the method details are provided in Section C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Section C provides all the details for reproducing the results and the code will be released upon publication.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All the implementation details are provided in Section 5.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We provide accurate results and analysis in Sections 5.2 and Appendix F.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Compute resource information is disclosed in Section 5.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The authors have followed the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The broader impact is discussed in Section H.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited relevant models through out the paper and use them under their license.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The benchmark is presented in Section 3 and Appendix A.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [No]

    Justification: The LLM is used as a part of multi-agent system.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.