SEA-BENCH: SELF-EVOLVING AGENTIC BENCHMARK FOR VIDEO GENERATION

Anonymous authors
Paper under double-blind review

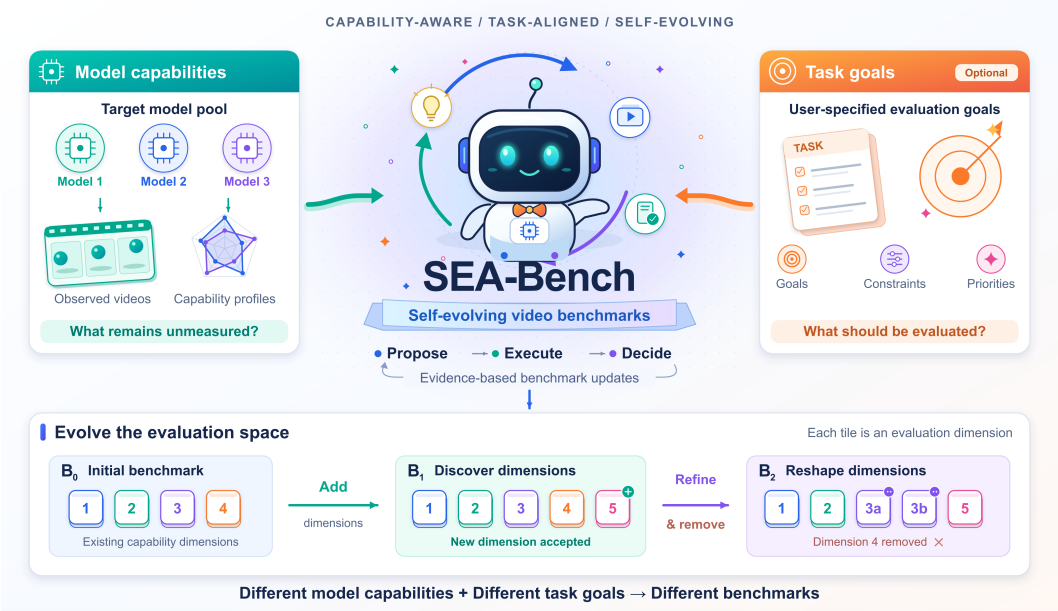

Figure 1: **SEA-Bench** constructs self-evolving video-generation benchmarks that adapt to model capabilities and task-specific goals. It evolves the evaluation space itself through evidence-based additions, refinements, and deletions of evaluation dimensions.

ABSTRACT

Video generation models are advancing rapidly, while their evaluation benchmarks remain largely static. Existing benchmarks typically fix the evaluated capability dimensions and task distributions, causing previously informative tests to saturate and newly emerging capabilities to remain unmeasured. We introduce SEA-Bench, a self-evolving benchmark construction framework that enables agents to discover and operationalize evaluation dimensions from observed model behaviors. Starting from an existing benchmark, SEA-Bench iteratively proposes capability hypotheses, designs executable prompts and evaluation procedures, validates them on target models, and uses an evidence-based decision agent to accept benchmark updates supported by the resulting evaluation evidence. Unlike prior dynamic evaluation methods that adapt instances or evaluation trajectories within a predefined capability space, SEA-Bench evolves the evaluation space itself. We instantiate SEA-Bench for text-to-video generation and evolve benchmarks for model pools with different capability profiles. Retrospective comparisons with independently developed video benchmarks assess the community alignment of retained dimensions, while human studies validate their meaningfulness, prompt faithfulness, and agreement between executable evaluators and human preferences. We compare aggregate model discrimination before and after benchmark evolution in the two evaluated model pools.

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

1 INTRODUCTION

Benchmarks are essential for measuring and comparing the rapidly advancing capabilities of modern AI systems. However, most benchmarks are static: after their release, both the capabilities they evaluate and the distributions of their evaluation tasks remain largely fixed. As models improve, such benchmarks face two distinct failure modes. First, existing tasks may saturate: models obtain similar high scores on the original prompts, so harder instances are needed within the same capability dimension. Second, and more fundamentally, the benchmark may miss emerging dimensions: two models can appear similar under existing metrics while exhibiting stable differences in behaviors that the benchmark was not designed to measure. The first problem calls for harder tasks; the second calls for expanding the evaluation space.

This distinction is especially important for video generation. The capability space of video models spans visual fidelity, temporal dynamics, physical consistency, controllability, compositionality, long-range consistency, and interaction. As models become more capable, their differences are not always captured by the dimensions emphasized in earlier benchmarks. A benchmark may therefore remain useful for measuring known capabilities while still failing to expose new failure modes that matter for comparing current models. Such gaps can arise in capabilities absent from the initialization; for example, text rendering emerged as a distinct evaluation dimension in our evolution of the higher-capability model pool, although it is not directly captured by the original quality and consistency axes.

Recent works have automated different parts of benchmark construction and evaluation. HarnessEval-W (Chen et al., 2026) constructs executable evaluation cases by combining scene taxonomies, probe categories, planning agents, and validation agents, and then selects skills and tools to evaluate visual-world capabilities. Benchmark Everything Everywhere All at Once (Xiong et al., 2026) automates the construction of benchmark instances by decomposing instructions, retrieving data, and generating and filtering multiple-choice questions. TRACE (Guo et al., 2026) evolves existing tasks through test-time exploration and uses reproducible execution trajectories to validate the resulting tasks. Evaluation Agent (Zhang et al., 2025) adapts the evaluation procedure by decomposing user-specified criteria into sub-aspects, generating prompts, and using intermediate evaluation feedback to refine the assessment process.

Existing frameworks support several forms of adaptivity, including query-conditioned refinement of evaluation sub-aspects, requirement-driven design of benchmark dimensions, context-dependent evidence gathering, and trajectory-based task evolution. However, these mechanisms do not directly address how to update a benchmark's capability dimensions using observed behavior across a target model pool. In particular, they do not validate proposed dimensions through target-model execution before deciding whether the benchmark itself should be updated. Table 1 summarizes this distinction.

In this work, we introduce **SEA-Bench**, a self-evolving benchmark construction framework for video generation. SEA-Bench treats evaluation dimensions as objects of evolution. Starting from an existing benchmark, a proposal agent analyzes model outputs and evaluation feedback to formulate candidate capability hypotheses. An execution agent then designs prompts and executable evaluation procedures, generates target-model videos, and collects evaluation evidence. Finally, an evidence-based decision agent determines whether each proposed addition, refinement, or deletion should update the benchmark. The framework can operate in an open-ended setting, where the agent searches for capability dimensions that characterize a target model pool, or in a task-specified setting, where a user provides an evaluation goal. We further explore the task-specified setting through a memory-focused benchmark evolution experiment.

We evaluate SEA-Bench through controlled experiments in text-to-video generation. We first test whether SEA-Bench discovers dimensions beyond its initialization and whether the discovered dimensions differ across model pools and user-specified goals. We then assess benchmark quality through community alignment with independently developed video-generation benchmarks, human validation of discovered dimensions, and aggregate model discrimination given the same number of iterations. Our contributions are as follows:

- We formulate **self-evolving benchmarking** as a benchmark paradigm in which the evaluation space itself can evolve in response to observed model behaviors, rather than remaining fixed after initialization.
- We develop **SEA-Bench**, an agentic benchmark construction framework that proposes capability hypotheses, designs executable evaluations, and iteratively updates a benchmark through addition, refinement, and deletion. Evaluation results, agent rationales, model discrimination, and construction context are provided to a decision agent to determine whether each proposed benchmark update should be accepted.
- We validate SEA-Bench through controlled text-to-video experiments, demonstrating its **adaptability to model pools and task goals**, the community alignment of retained dimensions, and human agreement with their executable evaluations.

2 Related Work

Video generation benchmarks. Early evaluation of generative video models relied on distributional and perceptual metrics such as IS, FID, and FVD (Salimans et al., 2016; Heusel et al., 2017; Unterthiner et al., 2018), which support scalable comparison but provide limited diagnostic information about model capabilities. Recent video-generation benchmarks therefore decompose evaluation into capability-oriented dimensions. VBench (Huang et al., 2024), EvalCrafter (Liu et al., 2024), and FETV (Liu et al., 2023b) evaluate visual quality, temporal consistency, motion, text-video alignment, and related aspects. Subsequent benchmarks further expand the evaluation space toward compositionality, physical plausibility, long-horizon consistency, narrative coherence, controllability, and world-level behaviors (Sun et al., 2025; Meng et al., 2024; Bansal et al., 2025; Wang et al., 2025b; Zheng et al., 2025; Huang et al., 2025; Ying et al., 2026). These works show that video-generation evaluation is inherently multi-dimensional, but their capability dimensions and evaluation protocols are primarily human-specified.

Model-based and agentic evaluation. Model-based evaluation uses learned metrics, vision-language models, or LLM judges to complement human assessment (Hessel et al., 2021; Lin et al., 2024; Liu et al., 2023a; Zhu et al., 2025). More recent agentic evaluators automate parts of the evaluation workflow. Evaluation Agent (Zhang et al., 2025) decomposes user-specified criteria into sub-aspects, generates prompts, invokes evaluation tools, and refines the evaluation plan using intermediate feedback. Open Evaluation Agent (Tian et al., 2026) studies history-conditioned evaluation planning, and HarnessEval-W (Chen et al., 2026) coordinates specialized agents and diagnostic tools for visual-world evaluation. These methods adapt how an evaluation is executed for a given objective, whereas SEA-Bench studies how the benchmark's evaluation dimensions themselves can be proposed, tested, and revised from model feedback.

Automated and evolving benchmarks. Another line of work automates benchmark construction by generating, filtering, or transforming evaluation instances. Dynabench (Kiela et al., 2021), DyVal (Zhu et al., 2024a), Meta Probing Agents Zhu et al. (2024b), and Benchmark Everything Everywhere All at Once (Xiong et al., 2026) improve scalability by constructing diverse or adaptive test instances. Recent self-evolving benchmark methods further modify task difficulty, prompts, contexts, or executable trajectories (Wang et al., 2025a; Guo et al., 2026; Jia et al., 2026). These approaches demonstrate that benchmark content can evolve, but they usually operate within predefined task families or capability taxonomies. SEA-Bench instead treats capability dimensions as hypotheses: each candidate dimension is operationalized as an executable evaluation and reviewed by an independent Decision Agent using its execution evidence and the current benchmark context.

Self-improving agents. SEA-Bench is also related to agents that improve from experience without updating model parameters. Reflexion (Shinn et al., 2023), Self-Refine (Madaan et al., 2023), Voyager (Wang et al., 2023), and ExpeL (Zhao et al., 2024) show that agents can store reflections, revise outputs, or reuse learned skills across tasks. These mechanisms are complementary to benchmark evolution: in SEA-Bench, validated evaluation procedures, implementation fixes, and rejected hypotheses can be summarized into reusable experience for future benchmark construction. Thus, SEA-Bench studies both benchmark-level evolution and, when enabled, agent-level improvement in the construction process.

Table 1: Capability-level comparison with related agentic evaluation and benchmark-construction frameworks.

Method	Automated prompts	Execution feedback	Benchmark update	Dimension discovery	Pool-aware evolution
Evaluation Agent (Zhang et al., 2025)	✓	✓	–	–	–
Benchmark Everything Everywhere All at Once (Xiong et al., 2026)	✓	–	–	✓	–
HarnessEval-W (Chen et al., 2026)	✓	–	–	–	–
TRACE (Guo et al., 2026)	✓	–	✓	–	–
SEA-Bench (Ours)	✓	✓	✓	✓	✓

3 SEA-BENCH

3.1 OVERVIEW

SEA-Bench iteratively revises a benchmark using evaluation feedback from a target model pool $\mathcal{M} = \{f_1, \ldots, f_M\}$. At iteration t, the benchmark B_t contains a set of capability dimensions, each associated with generation prompts,and an executable evaluation procedure. As illustrated in Figure 2, the framework follows a propose–execute–decide loop: agents formulate capability hypotheses, construct and run experiments, and decide whether the resulting evidence supports an update to the benchmark. Experimental outcomes are retained to guide subsequent iterations.

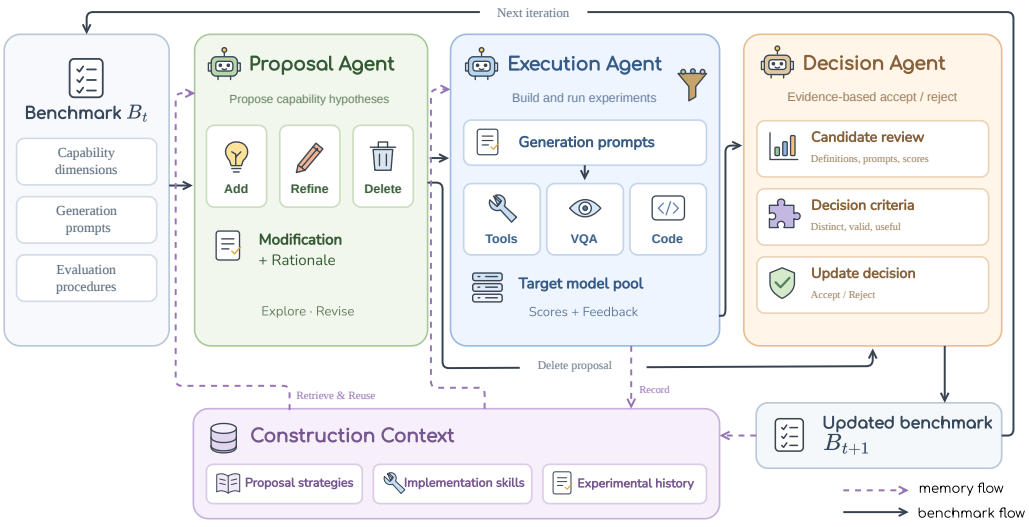

Figure 2: **The overall framework of SEA-Bench.** SEA-Bench iteratively evolves a benchmark through a propose–execute–select loop. The **Proposal Agent** proposes adding, refining, or deleting capability dimensions based on the current benchmark and experimental history. The **Execution Agent** operationalizes addition and refinement proposals by constructing generation prompts and evaluating target video generation models using existing tools, VQA, or customized code. **Decision Agent** reviews proposal rationales, executable evaluation results, model discrimination, and construction context to decide whether each proposed benchmark update should be accepted or rejected. **Construction Context** retains the current benchmark, predefined evaluation tools, prior proposals, execution outcomes, evaluator revisions, and update decisions for subsequent iterations.

3.2 CAPABILITY PROPOSAL AND EVALUATION

The Proposal Agent receives the current benchmark and experimental history, including previous hypotheses, evaluation scores, and selection decisions. It proposes three types of modifications: ADD introduces a dimension beyond the current coverage; REFINE revises an existing dimension or

its evaluation procedure; and DELETE removes a dimension or evaluation component. Each proposal specifies the intended modification and its rationale. The final set of dimensions is not prescribed in advance.

The Execution Agent operationalizes each ADD or REFINE proposal by constructing generation prompts and selecting an evaluation mechanism: an existing tool, visual question answering (VQA), or customized code. The resulting procedure is applied consistently across target models, producing a score vector

$$\mathbf{s}_d = (s_{1,d}, \ldots, s_{M,d})^\top \tag{1}$$

for candidate dimension d. Scores, evaluator rationales, and execution feedback are recorded as evidence for update decisions and subsequent proposal generation. Agent instructions and execution procedures are detailed in Appendix A.

3.3 EVIDENCE-BASED BENCHMARK REVISION

SEA-Bench updates the benchmark only after target-model execution. For an ADD or REFINE proposal, the Execution Agent produces a model score vector $\mathbf{s}_d = (s_{1,d}, \ldots, s_{M,d})^\top$ and associated evaluation evidence. Together with the current benchmark context, this evidence is reviewed by an independent Decision Agent, which returns a binary ACCEPT/REJECT decision with a rationale. Accepted updates produce B_{t+1}; full agent specifications and decision criteria are given in Appendix A.

3.4 CONSTRUCTION CONTEXT

SEA-Bench maintains a construction context across iterations. It contains the current benchmark, predefined evaluation tools, prior proposals, execution records, evaluator revisions, and accepted or rejected update decisions. The Proposal and Decision Agents retrieve relevant entries from this context when formulating and assessing subsequent benchmark updates. This persistent record prevents the agents from treating each iteration as an isolated construction task.

4 EXPERIMENTS

We evaluate SEA-Bench through four questions:

1. Can SEA-Bench discover new or substantively refined evaluation dimensions beyond its initialization?
2. Does SEA-Bench adapt its discovered benchmark to different target-model pools?
3. Do the evolved benchmarks provide more discriminative and valid evaluation than the initial benchmark?
4. Can SEA-Bench construct a benchmark for a user-specified evaluation goal?

4.1 EXPERIMENTAL SETUP

Task and initialization. We instantiate SEA-Bench in text-to-video generation, using VBench as the initial benchmark unless otherwise specified. In the default setting, the agent explores evaluation dimensions without a prescribed capability target. We additionally consider a task-specified setting in which the agent is instructed to construct a benchmark for video-model memory. In both settings, the agent designs the dimensions and evaluation procedures, guided by the current benchmark and accumulated experimental feedback. The evaluation options comprise existing tools, vision-language-model-based question answering (VQA), and customized code.

Model pools. We study two pools of four commercial video-generation models, selected to contrast cost-efficient and higher-capability systems. Group A is a cost-efficient pool comprising `happyhorse-1.1-t2v`, `wanx2.1-t2v-turbo`, `pixverse/pixverse-c1-t2v`, and `vidu/viduq2_text2video`. Group B is a higher-capability pool comprising `kling/kling-v3-video-generation`, `wan2.5-t2v-preview`, `pixverse/pixverse-v5.6-t2v`, and `MiniMax/MiniMax-H3`. We evolve a separate



(a) Group A (b) Group B

Figure 3: **Dimension-level profiles of the evolved benchmarks.** Each axis reports the normalized score of one final benchmark dimension. The two panels use the same score range $[0, 1]$; dimensions are shown separately because the evolved benchmarks contain pool-specific dimensions.

benchmark for each pool to examine how evaluation priorities differ across model capability profiles.

Agent and evaluator. The benchmark evolution agent is powered by `deepseek-v4.1-flash`. For VQA-based evaluation, we use `Qwen3.8-Flash` as the vision-language evaluator. The same agent and evaluator configuration is used across all evolution settings.

4.2 RQ1: Discovered Evaluation Dimensions

We examine whether SEA-Bench expands the evaluation space beyond its VBench initialization. Starting from the same initial benchmark, we independently run open-ended evolution on Group A and Group B. We report the new or substantively refined dimensions retained in the final benchmarks.

Table 2 summarizes the discovered capability families. The two model pools share several directions, including physical interaction, attribute composition, and social interaction, while also yielding pool-specific dimensions. Group A emphasizes counting, constraint following, and fine-grained motion. Group B emphasizes camera control, text rendering, temporal structure, human embodiment, and appearance-related effects.

These results show that SEA-Bench expands the evaluation space beyond the initial benchmark rather than only generating harder instances within its original dimensions. Whether the discovered dimensions provide valid and diagnostic measurements is evaluated separately in RQ3.

4.3 RQ2: Adaptation to Target Model Pools

We examine whether SEA-Bench produces pool-specific evaluation spaces rather than converging to the same set of dimensions for every target model pool. Group A and Group B differ not only in their accepted dimensions, but also in the types of capabilities emphasized by those dimensions.

The Group-A-specific dimensions emphasize overall visual quality, constraint following, counting, localized motion, and cyclic-motion consistency. In contrast, the Group-B-specific dimensions emphasize camera control, text rendering, hand and full-body articulation, event sequencing, material and fluid dynamics, lighting and transparency, weather, cinematographic rendering, and

Table 2: Capability families discovered beyond the VBench initialization. A checkmark indicates that the final benchmark contains a newly added or substantively refined dimension in the corresponding family.

Capability family	Group A	Group B
Physical interaction	✓	✓
Attribute composition	✓	✓
Social interaction	✓	✓
Counting and quantity	✓	–
Constraint following	✓	–
Fine-grained motion	✓	–
Camera control and cinematography	–	✓
Text and temporal structure	–	✓
Human embodiment	–	✓
Appearance and environmental effects	–	✓

Table 3: Cross-pool diagnostic value of Group-B-origin dimensions under identical evaluation specifications. ΔDisc. denotes the difference between Group B and Group A discrimination.

Dimension	Group A Disc.	Group B Disc.	ΔDisc.
Camera motion control	0.0982	0.1830	+0.0848
Text rendering in video	0.0663	0.1785	+0.1122
Temporal event sequencing	0.0872	0.4001	+0.3129

color grading. Thus, the two evolved benchmarks focus on different capability profiles: Group A emphasizes perceptual and constraint-oriented evaluation, whereas Group B covers more specialized cinematic, physical, embodied, and event-level capabilities.

To quantify this difference, we re-evaluate three Group-B-origin dimensions on Group A using the same frozen prompts, evaluators, score normalization, and generation protocol. As shown in Table 3, all three dimensions have higher discrimination in Group B than in Group A. Discrimination increases from 0.0982 to 0.1830 for camera motion control, from 0.0663 to 0.1785 for text rendering, and from 0.0872 to 0.4001 for temporal event sequencing. The largest gap occurs for temporal event sequencing.

These results show that the dimensions discovered for Group B are not uniformly diagnostic across model pools. Together with the pool-specific capability profiles, the cross-pool evaluation supports the interpretation that SEA-Bench adapts benchmark construction to the observed target-model population.

4.4 RQ3: Benchmark Quality

We evaluate the quality of evolved benchmarks using three complementary forms of evidence: community alignment, human validation, and aggregate model discrimination.

Community alignment. To assess whether the discovered dimensions correspond to meaningful directions recognized by the community, we compare them with independently developed video generation benchmarks, including VBench-2.0 (Zheng et al., 2025), VBench++ (Huang et al., 2025), and W-Bench (Ying et al., 2026). The comparison is based on capability definitions and evaluation tasks rather than names alone. These external benchmarks are withheld from the evolution process and are used only for post-hoc analysis. Community alignment is treated as supporting evidence of relevance, rather than as a necessary condition for a valid discovery.

Table 4 shows that several accepted dimensions introduced beyond the VBench initialization align with capability directions independently emphasized by later benchmarks, including physical causality, attribute binding, human interaction, temporal event order, camera control, human anatomy, and material dynamics. Other accepted dimensions have no close counterpart in the

Table 4: Community alignment of retained post-initialization dimensions.

Capability family	VBench	VBench++	VBench-2.0	W-Bench	Group A	Group B
Physical interaction and causality	–	–	✓	✓	✓	✓
Attribute binding and composition	△	–	✓	–	✓	✓
Multi-agent social interaction	–	–	✓	△	✓	✓
Temporal event order	–	–	✓	△	–	✓
Camera-motion control	–	✓	✓	✓	–	✓
Human anatomy and articulation	–	–	✓	△	–	✓
Material, fluid, and particle dynamics	–	–	✓	✓	✓	✓
Lighting, shadow, reflection, and transparency	–	–	△	✓	–	✓
Weather and atmosphere	–	–	△	✓	–	✓
Text rendering	–	–	–	–	–	✓
Counting and quantity	△	–	–	–	✓	–
Negative constraints	–	–	–	△	✓	–
Localized part motion	–	–	–	–	✓	–
Periodic motion consistency	–	–	–	△	✓	–
Cinematographic lens rendering	–	–	–	–	–	✓
Color grading and tonal directives	△	–	–	–	–	✓

Table 5: Aggregate human validation results for selected newly added dimensions. Positive proportion excludes responses marked "cannot determine." Agreement is computed over resolved non-tied pairs.

Validation criterion	Positive/agreement	Krippendorff's α	Evaluation unit
Dimension endorsement	100.0%	1.000	Annotated dimensions
Prompt faithfulness	100.0%	1.000	Annotated prompts
Evaluator–human agreement	72.3%	0.776	47 non-tied pairs

compared benchmarks, suggesting additional directions for evaluation rather than failed alignment. The corresponding SEA-Bench dimension names and detailed alignment rationales are provided in Appendix D.

Human validation. We validate a subset of newly added dimensions using independent human annotations on examples held out from benchmark evolution and candidate selection. Annotators assess whether each dimension represents a meaningful video-generation capability, whether its prompts instantiate the intended capability, and whether the executable evaluator agrees with human pairwise preferences. We report positive proportions, evaluator–human agreement, and Krippendorff's α for annotation reliability. Detailed annotation procedures and examples are provided in Appendix B.

As shown in Table 5, all annotators endorsed the validated dimensions and judged their prompts faithful to the intended capabilities. The executable evaluator agreed with human preferences on 72.3% of the 47 resolved non-tied pairs, with a nominal Krippendorff's α of 0.776. These results support both the semantic validity of the discovered dimensions and the reliability of their prompt instantiations, while indicating room for improving the executable evaluators. The remaining disagreement likely reflects limitations of the current evaluator implementations, which rely primarily on existing VBench tools and VQA. Future versions could provide the Execution Agent with a broader library of basic visual and temporal tools and stronger support for task-specific custom evaluators.

Aggregate model discrimination. We measure whether an evolved benchmark separates models more clearly than the initial benchmark. For a benchmark B with D dimensions, we compute the equally weighted aggregate score of model m:

$$S_m(B) = \frac{1}{D} \sum_{d=1}^{D} s_{m,d}, \tag{2}$$

Table 6: Aggregate model discrimination before and after benchmark evolution.

Model pool	Initialization	Evolved	Absolute change
Group A	0.0142	0.0171	+0.0029
Group B	0.0135	0.0808	+0.0673

where all dimension scores are mapped to a common range. We define aggregate discrimination as the coefficient of variation of the aggregate model scores:

$$\mathrm{Disc}(B, \mathcal{M}) = \frac{\sqrt{\frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} \left(S_m(B) - \overline{S}(B)\right)^2}}{|\overline{S}(B)|}, \tag{3}$$

where $\overline{S}(B)$ is the mean aggregate score over $\mathcal{M}$. If $\overline{S}(B) = 0$, we define the discrimination as zero. A larger value indicates greater separation relative to the mean aggregate score within the evaluated pool using the same score normalization and per-dimension evaluation protocol.

4.5 RQ4: Task-Specified Memory Benchmark Evolution

We finally evaluate whether SEA-Bench can direct benchmark evolution toward a user-specified goal. We instruct the agent to construct a memory benchmark for text-to-video generation while keeping the proposal–execution–decision procedure unchanged. The resulting benchmark contains eight newly evolved dimensions: *object permanence*, *temporal-order memory*, *counting memory*, *initial-state memory*, *dynamic spatial memory*, *agent–action binding memory*, *long-horizon memory decay*, and *identity re-identification under appearance change*. Each dimension is paired with an executable evaluation procedure.

The discovered dimensions recover several capability directions represented in MBench (Zhang et al., 2026), although the two benchmarks differ in data, model setting, and evaluation protocol. Thus, this experiment demonstrates that a user-provided goal can guide SEA-Bench toward a coherent set of executable memory evaluations. Detailed dimension definitions are provided in Appendix C. Table 7 compares their capability coverage with MBench. This comparison is qualitative and does not imply directly comparable benchmark scores.

5 Further Discussion

The experiments show that SEA-Bench evolves the evaluation space itself rather than only generating harder instances within the initial VBench dimensions. The two model pools produce different capability profiles, and representative Group-B-origin dimensions provide greater score separation within Group B than within Group A under identical evaluation specifications. This suggests that the diagnostic value of an evaluation dimension depends on the target model population.

The memory experiment further shows that task specifications can guide open-ended benchmark evolution. Given the goal of evaluating video-model memory, SEA-Bench constructs eight executable memory dimensions covering entity, state, spatial, temporal, causal, and long-horizon retention, several of which correspond to capability families studied in MBench. Thus, a task specification constrains the search toward a desired evaluation concept without fixing the final dimension taxonomy in advance.

Human validation supports the semantic validity of the selected dimensions and prompts: both dimension endorsement and prompt faithfulness receive 100% positive responses with Krippendorff's $\alpha = 1.000$. Evaluator–human agreement reaches 72.3% on 47 resolved non-tied pairs, with $\alpha = 0.776$. The remaining disagreement reflects the limitations of the current evaluator implementations, which rely substantially on existing tools and VQA. Broader visual and temporal primitives and stronger task-specific code evaluators are promising directions for improving evaluator fidelity.

AI USE STATEMENT

We used generative AI tools to assist with language editing, LaTeX formatting, visualization design, code debugging, and the organization of research analyses. Generative AI tools were not used to generate experimental data, model outputs, human annotations, or unverified scientific claims. All AI-assisted text, code, analyses, and figures were reviewed and verified by the authors. Generated code was tested against the implementation and experimental results, and literature-related suggestions were checked against the cited papers. The authors take full responsibility for the final content, claims, and artifacts of this work.

ETHICS STATEMENT

This work evaluates publicly accessible video-generation models and does not release personal data or sensitive content. The human validation study involved annotation of generated videos and evaluation prompts. We do not identify individual annotators in the paper or release personally identifying information.

REPRODUCIBILITY STATEMENT

We describe the SEA-Bench framework, proposal–execution–decision procedure, evaluation metrics, and experimental settings in the main paper. Appendix A provides the agent instructions and execution procedures, Appendix B describes the human validation protocol, Appendix C documents the task-specified memory benchmark, and Appendix D provides the community alignment analysis. We will release the implementation, configuration files, evaluation specifications, and anonymized experiment records through an anonymous repository where permitted by the evaluation platforms and model-provider licenses.

REFERENCES

Hritik Bansal, Zongyu Lin, Tianyi Xie, Zeshun Zong, Michal Yarom, Yonatan Bitton, Chenfanfu Jiang, Yizhou Sun, Kai-Wei Chang, and Aditya Grover. Videophy: Evaluating physical commonsense for video generation. In *International Conference on Learning Representations*, 2025.

Weiliang Chen, Haowen Sun, Jun Gao, Jiawei Chi, Hanyang Wang, Qiyu Dai, Yihao Li, Hao Li, Jingnan Gao, Yi-Hsin Hung, et al. Harnesseval-w: Agentifying the evaluation of visual worlds. *arXiv preprint arXiv:2608.16859*, 2026.

Dadi Guo, Tianyi Zhou, Dongrui Liu, Chen Qian, Qihan Ren, Shuai Shao, Zhiyuan Fan, Yi R Fung, Kun Wang, Linfeng Zhang, et al. Towards self-evolving agent benchmarks: Validatable agent trajectory via test-time exploration. In *The Fourteenth International Conference on Learning Representations*, 2026.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, 2021.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 2017.

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

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

Qi Jia, Ye Shen, Xiujie Song, Kaiwei Zhang, Shibo Wang, Dun Pei, Xiangyang Zhu, and Guangtao Zhai. One battle after another: Probing llms' limits on multi-turn instruction following with a benchmark evolving framework. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2026.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, et al. Dynabench: Rethinking benchmarking in nlp. In *Proceedings of the 2021 conference of the North American chapter of the Association for Computational Linguistics: human language technologies*, 2021.

Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. In *European Conference on Computer Vision*, 2024.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, 2023a.

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

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*, 2023b.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 2023.

Fanqing Meng, Jiaqi Liao, Xinyu Tan, Wenqi Shao, Quanfeng Lu, Kaipeng Zhang, Yu Cheng, Dianqi Li, Yu Qiao, and Ping Luo. Towards world simulator: Crafting physical commonsense-based benchmark for video generation. *arXiv preprint arXiv:2410.05363*, 2024.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 2016.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 2023.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Shulin Tian, Ziqi Huang, Fan Zhang, Hongyuan Zhu, Yu Qiao, and Ziwei Liu. Open evaluation agent: Efficient and promptable evaluation of visual generative models. *arXiv preprint arXiv:2608.09666*, 2026.

Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Siyuan Wang, Zhuohan Long, Zhihao Fan, Xuan-Jing Huang, and Zhongyu Wei. Benchmark self-evolving: A multi-agent framework for dynamic llm evaluation. In *Proceedings of the 31st international conference on computational linguistics*, 2025a.

Yiping Wang, Xuehai He, Kuan Wang, Luyao Ma, Jianwei Yang, Shuohang Wang, Simon Shaolei Du, and Yelong Shen. Is your world simulator a good story presenter? a consecutive events-based benchmark for future long video generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025b.

Shiyun Xiong, Dongming Wu, Peiwen Sun, Yuang Ai, Bokang Yang, Wencheng Han, Xiao-Hui Li, and Xiangyu Yue. Benchmark everything everywhere all at once. *arXiv preprint arXiv:2606.06462*, 2026.

Kaining Ying, Hengrui Hu, Siyu Ren, Jiamu Li, Fengjiao Chen, Ziwen Wang, Xuezhi Cao, Xunliang Cai, and Henghui Ding. Wbench: A comprehensive multi-turn benchmark for interactive video world model evaluation. *arXiv preprint arXiv:2605.25874*, 2026.

Fan Zhang, Shulin Tian, Ziqi Huang, Yu Qiao, and Ziwei Liu. Evaluation agent: Efficient and promptable evaluation framework for visual generative models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025.

Shengjun Zhang, Zhang Zhang, Simin Huang, Zhenyu Tang, Hanyang Wang, Chensheng Dai, Min Chen, Yifan Li, Yuxin Li, Yingjie Chen, et al. Mbench: A comprehensive benchmark on memory capability for video world models. *arXiv preprint arXiv:2606.00793*, 2026.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2024*, 2024.

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.

Kaijie Zhu, Jiaao Chen, Jindong Wang, Neil Gong, Diyi Yang, and Xing Xie. Dyval: Dynamic evaluation of large language models for reasoning tasks. In *International Conference on Learning Representations*, volume 2024, 2024a.

Kaijie Zhu, Jindong Wang, Qinlin Zhao, Ruochen Xu, and Xing Xie. Dynamic evaluation of large language models by meta probing agents. *arXiv preprint arXiv:2402.14865*, 2024b.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. Judgelm: Fine-tuned large language models are scalable judges. In *International Conference on Learning Representations*, 2025.

A Agent Instructions and Execution Procedures

A.1 Construction Context

At iteration t, SEA-Bench provides the agents with the current benchmark B_t, the target model pool, and the construction history accumulated in previous iterations. The current benchmark contains the retained evaluation dimensions, their definitions, generation prompts, and executable evaluation procedures. The construction history contains previous hypotheses, evaluation results, proposal rationales, and accepted or rejected update decisions.

The construction context serves two purposes. First, it allows the Proposal Agent to identify uncovered capabilities and avoid repeatedly proposing previously rejected or duplicated hypotheses. Second, it allows subsequent agents to interpret new evaluation results relative to the current benchmark. The context is passed to the agents as structured records rather than as unstructured conversation history.

A.2 PROPOSAL AGENT

The Proposal Agent is prompted as an expert in video-generation model evaluation and benchmark design. It receives the current benchmark, the target model pool, and the accumulated hypothesis context. Its task is to identify uncovered capabilities, formulate testable hypotheses, and propose concrete benchmark modifications.

For ADD operations, the agent first performs a divergent search over candidate capabilities, including untested behaviors, edge cases, and fine-grained distinctions. It generates at most five candidates and then selects up to three candidates for downstream execution. The intermediate selection is used only to limit the number of expensive executions; it is not the final benchmark update decision. Each candidate contains a name, a capability gap, a capability definition, a proposal rationale, and an initial assessment of novelty, feasibility, and expected discrimination.

For REFINE and DELETE operations, the agent examines the retained benchmark dimensions and may propose at most one refinement and one deletion per iteration. A refinement specifies the target dimension, the new dimension name, and the intended improvement. A deletion specifies the target dimension and the rationale for removing it.

The Proposal Agent is instructed to avoid repeating previously rejected hypotheses and to use prior evaluation outcomes when identifying new opportunities. Its predicted novelty and discrimination are treated as proposal metadata only. Decisions about benchmark updates are made after execution by the Decision Agent.

A.3 EXECUTION AGENT

The Execution Agent converts each selected hypothesis into an executable evaluation procedure. It first chooses one of three evaluation options: an existing tool, visual question answering (VQA), or customized code.

Existing tools are preferred when the hypothesis is already covered by an available standardized metric. VQA is used for semantic or visual judgments that require flexible interpretation of the generated video. Customized code is used when the hypothesis requires combining multiple tools, extracting structured evidence, or applying a task-specific scoring procedure.

For each hypothesis, the agent returns the selected evaluation option and a choice rationale. It then generates the corresponding evaluation specification, including the capability definition, generation prompts, evaluation questions or code, scoring criteria, and output format. The procedure is executed consistently across all models in the target pool.

For a candidate dimension d, the execution stage produces a model score vector

$$\mathbf{s}_d = (s_{1,d}, \ldots, s_{M,d})^\top, \tag{4}$$

together with the generated videos, evaluator outputs, evaluator rationales, execution logs, and any implementation diagnostics. Scores are mapped to a common predefined range before they are passed to the Decision Agent.

If a customized evaluator fails during execution, the agent may revise the implementation and rerun the evaluation. Scores produced by an outdated evaluator implementation are invalidated. After an evaluator revision, all videos associated with the affected candidate are rescored using the same evaluator version before the candidate is considered for a benchmark update.

A.4 DECISION AGENT

The Decision Agent evaluates every proposed ADD, REFINE, and DELETE operation independently. It is instantiated with temperature 0 and no conversational history. Its system instruction is:

> You are the decision agent for an evolving video benchmark. For each candidate, use the criteria for its stated operation. Do not invent evidence that is absent from the input. Return JSON only with exactly these fields: `accept` (boolean) and `reasoning` (brief string).

For an ADD candidate, the agent checks whether the proposed dimension targets a meaningful capability or failure mode that is not already adequately covered, whether its prompts and evaluator operationalize that capability, and whether its results provide sufficiently stable model discrimination. Similar score patterns alone are explicitly insufficient to establish redundancy; the agent is instructed to compare evaluation targets, required evidence, and diagnosed failure modes.

For REFINE, the agent checks whether the revision improves the definition, prompts, evaluator, or scoring procedure of an existing dimension; preserves the intended evaluation target or justifies a necessary correction; and is at least as operationally valid as the original version. Improved diagnostic value, discrimination, or measurement reliability provide supporting evidence.

For DELETE, the agent checks whether a dimension no longer provides a distinct and useful target, is operationally invalid, or is adequately covered by another retained dimension. Score similarity alone is insufficient. When available, the effect of deletion on held-out discrimination and evaluation coverage is used as supporting evidence.

For each decision, the agent receives the operation type, the complete current benchmark definitions, and the complete candidate definition and rationale. Runtime artifacts `video_paths` and `prompt_scores` are removed from both views. Thus, the candidate may include its hypothesis, choice rationale, prompts, cases, evaluator definition, aggregate score vector, and any other non-runtime fields produced upstream. The decision record stores the operation, candidate name, target name, evaluation ID, hypothesis ID, Boolean acceptance result, and textual reasoning.

The output is a JSON object of the form {`"accept": true/false, "reasoning": "..."`}. It therefore realizes an explicit ACCEPT/REJECT decision, where `accept=true` denotes acceptance.

The implementation also defines a scalar discrimination statistic for a score vector $\mathbf{s} = (s_1, \ldots, s_M)$:

$$\mathrm{disc}(\mathbf{s}) = \begin{cases} 0, & M < 2 \text{ or } \bar{s} = 0, \\ \dfrac{\mathrm{std}(\mathbf{s})}{|\bar{s}|}, & \text{otherwise}, \end{cases} \tag{5}$$

where $\bar{s}$ is the mean score. This is the coefficient of variation of the model scores. In the provided implementation, it is only a utility function: it is not directly invoked by the decision loop and does not override the Decision Agent's judgment. It can be included in a candidate as supporting evidence.

Accepted updates are applied as follows. An accepted ADD candidate is appended to the benchmark, and the candidate buffer is cleared when at least one addition is accepted. An accepted REFINE replaces the benchmark entry whose name matches the candidate's target name. An accepted DELETE removes the entry with the specified target name. If the same target is both refined and deleted, the original entry is removed and the refined candidate is appended as a new entry.

B Human Validation Protocol

Dimension selection. We select 8 newly added dimensions for human validation, covering dimensions unique to Group A, unique to Group B, and dimensions shared across pools. The selected dimensions span the evaluation types used by SEA-Bench. No validation prompt or video is used during benchmark evolution, candidate selection, or evaluator development.

Annotation tasks. For each selected dimension, annotators complete three independent tasks.

Dimension endorsement. Annotators are shown the dimension definition and representative positive and negative examples. They answer: "Does this dimension capture a meaningful capability for evaluating text-to-video generation?"

Prompt faithfulness. Annotators are shown the dimension definition and a held-out prompt. They answer: "Does this prompt require the capability described by the dimension?"

Evaluator–human agreement. Annotators are shown a held-out prompt and a randomized pair of videos generated from that prompt. They answer: "Which video performs better with respect to the

Table 7: Memory-focused dimensions discovered in the task-specified evolution and their correspondence to MBench capability families. The comparison is qualitative and does not imply directly comparable benchmark scores.

Dimension	Memory target	Verification structure	Evaluator	Related MBench family
Object permanence	Entity retention	An object is occluded or leaves the view and later reappears.	VQA	Entity consistency
Temporal-order memory	Event-order retention	Multiple events occur before a delayed order-sensitive verification.	VQA	Causal consistency
Counting memory	Entity-count retention	Objects are counted before and after movement, occlusion, or transitions.	VQA	Entity consistency
Initial-state memory	State retention	A later scene requires recovering an earlier object or scene state.	VQA	Environment consistency
Dynamic spatial memory	Spatial-relation retention	Camera or object motion occurs before a delayed spatial verification.	VQA	Environment consistency
Agent–action binding memory	Agent–action association	Multiple agents act before a delayed verification of who performed which action.	VQA	Causal consistency
Long-horizon memory decay	Retention over temporal distance	The delay or intervening content increases before the verification event.	VQA	Cross-cutting
Identity re-identification	Identity retention	An entity reappears after changes in viewpoint, lighting, pose, or appearance.	VQA	Entity consistency

stated dimension?" with options Video A, Video B, and Tie / Cannot determine. We compare the majority human preference with the preference returned by the executable evaluator.

Protocol and aggregation. Model identities, pool labels, SEA-Bench selection outcomes, and evaluator scores are hidden during annotation. For the first two tasks, we report the proportion of positive responses after excluding "Cannot determine" responses. For pairwise evaluation, we report agreement with the majority human preference on non-tied, human-resolved pairs. We report inter-annotator agreement using Krippendorff's α.

C Memory Benchmark Details

Table 7 summarizes the memory-focused dimensions discovered in the task-specified evolution, including their memory targets, verification structures, and evaluators. We also report their qualitative correspondence to MBench capability families; this comparison does not imply directly comparable benchmark scores.

D Community Alignment Details

Table 8: Detailed community alignment of post-initialization SEA-Bench dimensions. Alignments follow capability definitions and evaluation targets rather than name matching.

Capability family	SEA-Bench dimensions	Related external capability directions
Physical interaction and causality	Group A: `physical_plausibility_and_contact` Group B: `physical_interaction_causality`	VBench-2.0: Mechanics; Motion Rationality. W-Bench: Causal Fidelity; Visual Plausibility.
Attribute binding and composition	Group A: `attribute_binding_and_composition_v2` Group B: `attribute_binding`	VBench-2.0: Composition. Related VBench components: Color, Multiple Objects, and Spatial Relationship.
Multi-agent social interaction	Group A: `social_interaction_and_multi_agent_coordination` Group B: `multi_agent_social_interaction`	VBench-2.0: Human Interaction. W-Bench: Subject Action Adherence; Causal Fidelity.
Temporal event order	Group B: `event_sequencing_temporal_order`	VBench-2.0: Motion Order Understanding. W-Bench: Event Editing Adherence.
Camera-motion control	Group B: `camera_motion_control`	VBench++: Camera Motion. VBench-2.0: Camera Motion. W-Bench: NavScore; Perspective Switching Adherence.
Human anatomy and articulation	Group B: `hand_articulation_dexterity` Group B: `human_full_body_kinematics`	VBench-2.0: Human Anatomy. W-Bench: Causal Fidelity includes human motion and expression checks.
Material, fluid, and particle dynamics	Group A: `environmental_effects_and_particle_systems_v2` Group B: `material_and_fluid_dynamics`	VBench-2.0: Material; Thermotics. W-Bench: Causal Fidelity includes fluid, smoke, wind, and environmental forces.
Lighting, shadow, reflection, and transparency	Group B: `dynamic_lighting_and_shadow_consistency` Group B: `reflection_and_transparency_consistency`	W-Bench: Causal Fidelity includes reflection and lighting checks.
Weather and atmosphere	Group B: `weather_and_atmosphere`	W-Bench: Event Editing Adherence includes weather changes; Causal Fidelity includes environmental forces.
Text rendering	Group B: `text_rendering_in_video`	No direct counterpart in the compared benchmarks.
Counting and quantity	Group A: `counting_and_quantity`	No direct counterpart in the compared benchmarks.
Negative constraints	Group A: `negative_constraint_adherence`	Related to W-Bench Event Editing Adherence, which checks for anomalous unrelated entities.
Localized part motion	Group A: `localized_part_motion_disentanglement`	No direct counterpart in the compared benchmarks.

Capability family	SEA-Bench dimensions	Related external capability directions
Periodic motion consistency	Group A: `periodic_cyclic_motion_consistency`	Related to W-Bench NavScore consistency across repeated or symmetric actions.
Cinematographic lens rendering	Group B: `cinematographic_lens_rendering`	No direct counterpart in the compared benchmarks.
Color grading and tonal directives	Group B: `color_grading_and_tonal_directives`	Related to VBench Color, but evaluates global grading and tonal directives rather than object–color binding.