

VISTA: A Controllable Platform for Generating and Auditing Egocentric Assistance Scenarios

Yu-Hsiang Liu, Yu-Chien Tang, An-Zi Yen

Department of Computer Science, National Yang Ming Chiao Tung University, Taiwan
ivesliu.ee10@nycu.edu.tw, tommytyc.cs10@nycu.edu.tw, azyen@nycu.edu.tw

Abstract

First-person assistance situations are difficult to collect with controlled timing and evidence: rare hazards should not be staged, while ordinary needs often depend on subtle visual and conversational context. We present VISTA, an interactive platform that treats scenario creation as an inspectable authoring workflow rather than a one-shot video prompt. A six-stage compiler turns a seed into typed intent, scene, event, interaction, rendering, and media artifacts; human reviewers can inspect revisions and provider requests before approving execution. We evaluate the rendered outputs, not the authoring interface. In a blinded study of 60 scenarios, 11 evaluators supplied 165 judgments. They selected the human-reviewed VISTA output as the best match to the seed in 52.1% of judgments, versus 22.4% for a one-pass seed-plus-assets baseline and 25.5% for seed-only generation. Case-balanced seed-match scores were 3.65, 3.27, and 3.22, respectively. These results support the complete workflow under the tested renderers; they do not establish interface usability, authoring efficiency, or a novel planning algorithm. The public artifact is a credential-free prepared replay with a narrated screencast.

1 Introduction

Assistance in daily first-person settings is expressed in different ways. A person may directly ask for help, mention a difficulty without requesting an intervention, or provide no verbal signal at all. The underlying event may be hazardous, such as using a sharp tool improperly, or mundane, such as misplacing an object or omitting a task step. Studying this variation requires control over what is visible, when the event unfolds, what is said, and whether assistance is warranted.

Large egocentric datasets provide authentic recordings of daily and procedural activity (Grauman et al., 2022; Wang et al., 2023; Ragusa et al.,

2026), and recent work synthesizes proactive dialogue from streaming first-person video (Zhang et al., 2025). Yet a collected clip cannot readily be retargeted by changing one cue, event time, or outcome while holding the rest of the scene fixed. Rare hazards also should not be intentionally staged. Generative video can fill targeted gaps, but direct prompting hides design decisions in opaque requests and makes failed outputs difficult to diagnose.

We introduce **VISTA**, a platform for generating and auditing controlled egocentric assistance scenarios. VISTA treats video generation as a structured authoring workflow: a scenario seed is transformed into inspectable artifacts describing intent, scene affordances, temporal beats, interaction mode, camera constraints, and expected media evidence. Users may edit artifacts or review a scoped proposal from the human-guided VISTA Agent before authorizing a provider call. Each candidate remains linked to its planning and review record. The contribution is this inspectable, controllable workflow and system integration; VISTA is not presented as a new planning or generation algorithm.

VISTA organizes scenarios along two orthogonal axes. The interaction axis comprises *reactive*, *explicit proactive*, and *implicit proactive* modes. The consequence axis distinguishes *safety-critical* from *everyday-inconvenience* cases. *No-assistance* controls represent normal or resolved scenes. The evaluation unit in this paper is therefore the fidelity of an authored scenario’s rendered evidence, not the behavior of a downstream assistant or the usability of the authoring interface.

Our contributions are threefold:

- a scenario model separating interaction mode from consequence type, with explicit no-assistance controls;
- an interactive platform exposing typed planning, revision, validation, provider gating, provenance, and export artifacts; and

- a blinded output study of 60 cases and 165 judgments comparing the human-reviewed VISTA workflow with two one-pass generation baselines.

The prepared public replay is available as a [live demonstration](#), accompanied by a [screencast](#).

2 Related Work

Egocentric assistance data. Ego4D established broad naturally occurring first-person video coverage (Grauman et al., 2022). HoloAssist records remote instructors guiding task performers (Wang et al., 2023), and Ego-EXTRA captures unscripted expert–trainee interaction from the trainee’s viewpoint (Ragusa et al., 2026). PROASSIST synthesizes proactive assistant dialogue from annotated streaming egocentric video (Zhang et al., 2025). These resources contribute authentic activity, dialogue, or labels. VISTA instead supports prospective authoring: a user specifies a target condition and can revise its visual and temporal realization before rendering.

Controlled video generation and evaluation. Video diffusion models enable text-conditioned temporal synthesis (Ho et al., 2022). EgoVid-5M targets egocentric video generation with action descriptions (Wang et al., 2024), while VBench++, T2V-CompBench, and VideoScore assess general quality, compositional faithfulness, and human-aligned quality (Huang et al., 2024; Sun et al., 2024; He et al., 2024). VISTA is complementary: its target is whether a clip preserves a specified assistance scenario, and its intermediate artifacts expose that target before and after generation.

Interactive research platforms. Thresh exposes configurable annotation schemas (Heineman et al., 2023), ChatHF integrates multimodal feedback into an interactive interface (Li et al., 2024), and EvalAssist supports iterative synthetic-data construction (Santillán Cooper et al., 2025). VISTA adopts this human-in-the-loop principle for first-person scenario authoring, with a provider gate and portable review package. Table 1 summarizes advertised primary functionality; it is not an empirical ranking, and our study does not compare authoring tools.

3 System Architecture

VISTA separates scenario intent from media rendering. As shown in Figure 1, a case first receives

Resource	FP	Dlg.	Ctrl.	Edit	VGen	Trail
Ego4D (Grauman et al., 2022)	✓	○	–	–	–	–
HoloAssist (Wang et al., 2023)	✓	✓	–	–	–	–
Ego-EXTRA (Ragusa et al., 2026)	✓	✓	–	–	–	–
PROASSIST (Zhang et al., 2025)	✓	✓	○	–	–	–
EgoVid-5M (Wang et al., 2024)	✓	–	○	–	✓	–
Thresh (Heineman et al., 2023)	–	–	✓	✓	–	○
EvalAssist (Santillán Cooper et al., 2025)	–	–	✓	✓	–	○
VISTA	✓	✓	✓	✓	✓	✓

Table 1: Representative resources. FP: first-person evidence; Dlg.: grounded dialogue; Ctrl.: user-level scenario control; VGen: video generation. Circles denote partial or adjacent functionality.

explicit categorical and temporal structure and is only then compiled into image, motion, and video requests. Every stage is serializable, revisable, and linked to the resulting media.

3.1 Scenario Taxonomy

Two labels define an assistance case before rendering. *Interaction mode* specifies how the need becomes observable: *reactive* cases contain a direct request, *explicit proactive* cases reveal a recognized need without addressing the assistant, and *implicit proactive* cases convey the need through visual evidence and event progression. *Consequence type* separates a plausible hazard from a lower-risk mistake, inefficiency, or missing object. Assigning both labels before rendering reduces the chance that visual details silently change the intended condition.

No-assistance controls depict normal or resolved situations and sit outside the assistance grid. They test whether authoring can preserve the absence of a problem instead of introducing a dramatic event into every clip. Figure 2 shows reviewed examples; dialogue is external text context rather than in-video speech.

3.2 Six-Stage Compiler

1. The *design brief* converts the seed and selected labels into a contract covering interaction mode, consequence, dialogue policy, required evidence, and forbidden dependencies.
2. *Scene affordance* specifies the setting, task frame, object inventory, spatial relations, and evidence that must be visually recoverable.
3. The *event skeleton* expands the scene into a 12-second sequence of setup, development, decision, and outcome beats. Each beat records duration, visible goal, action, camera behavior, evidence, and end state.

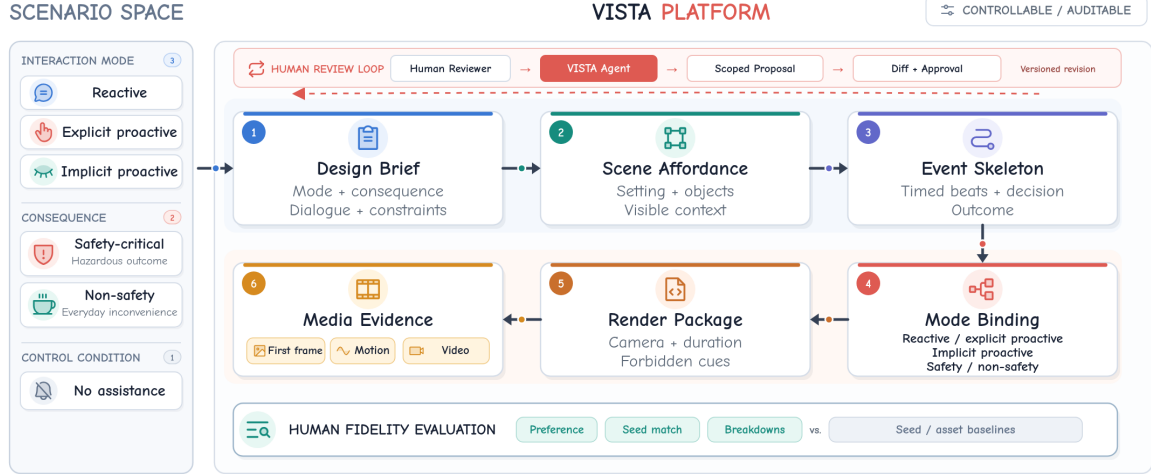


Figure 1: **VISTA architecture.** Scenario intent passes through six inspectable stages; scoped revision proposals require human approval before provider execution. The output study compares the resulting human-reviewed candidate with one-pass Seed+assets and Seed-only conditions.

Three assistance modes + no-assistance control

User state + observable evidence + annotated timing determine the expected policy

D dialogue onset I earliest useful intervention

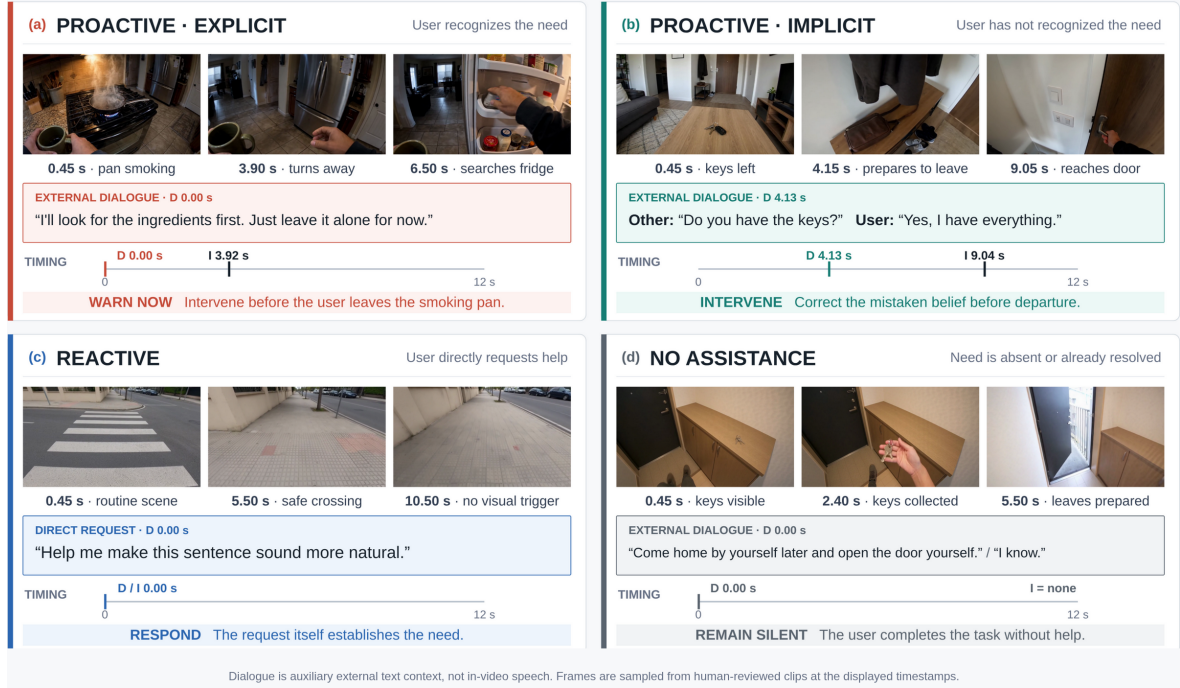


Figure 2: **Three assistance modes and a no-assistance control.** Each panel shows three frames, auxiliary dialogue, and the intended response policy. **D** marks dialogue onset and **I** the earliest useful intervention; no-assistance has no intervention target. Safety-critical and everyday-inconvenience consequences apply orthogonally to the three assistance modes.

4. *Mode binding* encodes the selected interaction mode through awareness, gaze, hand behavior, camera policy, dialogue affordance, and the useful intervention window while preserving the event state.
5. The *render package* compiles duration, ego-

centric camera constraints, timestamped actions, required and forbidden cues, audio policy, end state, and per-shot complexity limits into `render_script.yaml` and provider-shaped prompts.

6. *Media evidence* stores the first-frame prompt

Artifact	Engine / setting	Validation and record
Six planning stages	GPT-5.4 via OpenRouter; $T = .2$	strict JSON key; typed fields/enums; persisted prompt, request, response, and metadata
First frame and motion sheet	GPT Image 2; medium; 1280×720	prompt, image, usage, and attempt metadata
Video	Seedance 2.0 via OpenRouter or Sora 2 via OpenAI; 12 s, 1280×720, audio off	redacted request, job, polling history, output, and failure record

Table 2: Recorded generation configuration. The video backend is selected by the evaluation protocol rather than by a learned router.

and image, optional motion sheet, video request and candidate, model/provider metadata, and review status as separate artifacts.

Artifacts form a dependency chain. An accepted upstream change identifies downstream script, frame, motion, and video artifacts as stale before any regeneration. The review package retains the current script, parsed scene, prepared requests, attempt history, and human decisions rather than only the selected video.

3.3 Models, Schemas, and Failure Handling

Table 2 reports the archived execution configuration. The first six rows of the planning trace are separate calls to `openai/gpt-5.4` through the OpenRouter chat endpoint. Each call uses temperature 0.2 and a stage-specific template with upstream JSON inserted into named fields; neither top- p nor a maximum-token override is supplied. The expected top-level object matches the stage: design brief, scene affordance, visual event skeleton, mode binding, render compile, or first-frame anchor.

Pydantic schemas require typed fields and constrain interaction, safety, shot role, camera move, and assembly enums. After the design-brief call, a separate invariant check rejects drift from the preselected reasoning condition, interaction mode, or safety tag. The compiler normalizes duration and supported camera/role strings, validates each stage before it can feed the next, and halts rather than sending an invalid artifact to a media provider. Malformed planning output is tried at most three times; attempts two and three add a stricter “JSON only, required top-level key” instruction, and each failure is persisted.

GPT Image 2 requests retry transient HTTP failures up to three times with backoff. For the study runner, Sora submission likewise permits two retries after the initial attempt; Seedance can advance

to another configured credential on authentication, quota, or rate-limit failure. Both video paths persist polling histories and terminal failures. Semantic failures are never silently accepted or automatically rerendered: a reviewer chooses whether to revise, resubmit, keep, or discard a candidate.

The application is a React/TypeScript client over a FastAPI service using Pydantic and PyYAML. The camera-ready lockfiles pin React 19.2.1, TypeScript 5.9.3, FastAPI 0.123.5, Pydantic 2.12.5, and PyYAML 6.0.3. These dependencies implement the workflow and validation; they are not part of the paper’s scientific novelty claim.

4 Interface and Human-Guided Revision

The research interface targets multimodal researchers and evaluation designers who need to construct and inspect first-person cases. Its authoring workspace groups the compiler into persistent *Seed* → *Intent* → *Script* → *Artifacts* → *Trail* / *Export* views. The Intent view exposes setting, objects, event, interaction mode, and consequence before temporal expansion. The Script view exposes timed beats and decision points. Artifacts shows the first-frame, motion-sheet, and video requests without silently calling a provider, while Trail / Export retains attempts, revisions, and provenance.

4.1 VISTA Agent

Figure 3 shows the human-gated revision flow. VISTA Agent is a human-guided revision assistant, not an agent acting inside the generated video. The interface lets a reviewer scope feedback to the full chain, script, first frame, motion, or video plan. The backend default recorded in the camera-ready source is GPT-5.5 with high reasoning, invoked through an ephemeral Codex CLI run. The request carries the seed, interaction labels, active timing, current script revision, and optionally a copied video. The run receives an isolated workspace containing `context.json`, an immutable original script, and the current `render_script.yaml`; only the latter is writable. Its task instruction forbids provider calls, regeneration, package installation, and repository edits.

The assistant returns a message, a unified diff, and a draft script. The service parses the YAML and requires a mapping with `Global_Metadata`, `Scene`, and a nonempty `Scene.Actions` list before marking a draft usable. A reviewer must explicitly adopt and save the draft; only a later, separate gen-

eration action can contact a provider. The Agent itself does not retry a failed run: error or cancellation is surfaced in the conversation record and the reviewer may resubmit. Thus the UI’s artifact scope controls invalidation and follow-up work, while the Agent’s write authority remains limited to one copied YAML file.

4.2 Execution and Public Replay

Live authoring separates preparation from execution. Provider readiness, model identity, remote-call state, and available provider-reported usage remain visible; unavailable cost fields are not converted to zero. An image or video request can run only after credential readiness and explicit confirmation. Prepared requests and every attempt are preserved whether execution succeeds or fails.

The current public Space is deliberately narrower than the internal authoring workspace. It is a credential-free, read-only replay of 14 reviewed examples organized as Scenario → Direction → Output → Evidence, plus deterministic VISTA Agent proposal replays. It accepts no credentials, creates no shared sessions, makes no generation calls, and does not publish the underlying benchmark. Consequently, the screenshots demonstrate inspectability and approval flow; they are not evidence of authoring usability, speed, or learnability.

5 Output-Fidelity Evaluation

We evaluate the end-to-end media-fidelity question supported by the study: do the final VISTA videos preserve the user’s seed better than two one-pass generation conditions? Evaluators judged outputs only. They did not use the authoring interface, complete creation tasks, or report completion time, revision effort, satisfaction, or learnability.

5.1 Cases and Generation Conditions

We sampled 60 cases: 15 reactive, 15 explicit proactive, 15 implicit proactive, and 15 no-assistance controls. The 45 assistance cases comprise 14 safety-critical and 31 everyday-inconvenience cases. Each case contains one video from each condition:

- **VISTA**: the candidate selected during VISTA authoring after human review, using the compiled scenario artifacts;
- **Seed+assets**: a one-pass baseline using the seed plus a seed-derived first frame and, where supported, motion sheet generated with GPT Image 2, but no VISTA render script; and

- **Seed only**: a one-pass video from the seed and a common strict first-person instruction, with no image asset or script.

Within a case, all three conditions use the same video backend and target a 12-second, 1280×720, silent, first-person clip. Explicit and implicit cases use bytedance/seedance-2.0 through OpenRouter; reactive and no-assistance cases use sora-2 through the OpenAI video endpoint. For Sora 2, Seed+assets receives only the first-frame reference because the study runner does not send a motion-sheet reference to that endpoint. This design controls the backend within a case, but ties backend family to interaction mode and therefore does not support causal comparisons between modes.

5.2 Output-Only Annotation Protocol

Eleven CS-majoring students contributed 165 blinded case judgments. For each case, an evaluator saw the original seed and three candidates labeled *Video 1–3*. Condition identities were hidden, and the six possible display orders were balanced across cases. Evaluators (i) selected the video that best matched the scenario and (ii) rated each candidate’s seed match from 1 (poor) to 5 (excellent). Forty-five cases received three judgments and 15 received two; we retain every complete judgment rather than discard the partially triplicated cases.

Best-video preference is the fraction of all 165 judgments selecting each condition. For ratings, we first average evaluators within a case and then average the 60 case means, giving cases with two and three reviews equal weight. Confidence intervals use 20,000 case-cluster bootstrap samples, keeping all judgments for a sampled case together. We compare case means using two-sided paired Wilcoxon signed-rank tests and Holm-correct the two VISTA-baseline tests. On the 45 triplicated cases, Fleiss’ $\kappa = 0.215$ for the best-video choice, indicating low-to-modest agreement; this motivates reporting all judgments, clustered uncertainty, and continuous ratings rather than only unanimous cases.

The annotation store retains legacy condition-slot keys from an upstream bookkeeping error. Before aggregation, the analysis verifies all 180 condition paths and applies the frozen correction: the legacy “seed only” slot identifies VISTA, the legacy “VISTA pipeline” slot identifies Seed only, and the assets slot is unchanged. Raw judgments are not modified.

(a) Reviewer feedback and target scope.

(b) Accepted proposal and revision receipt.

Figure 3: **Human-gated revision in the prepared demo replay.** A reviewer supplies a correction and selects the affected scope (a), then inspects and accepts a structured proposal (b). The replay records a revision receipt but performs no model or media-provider call and does not claim a newly rendered video.

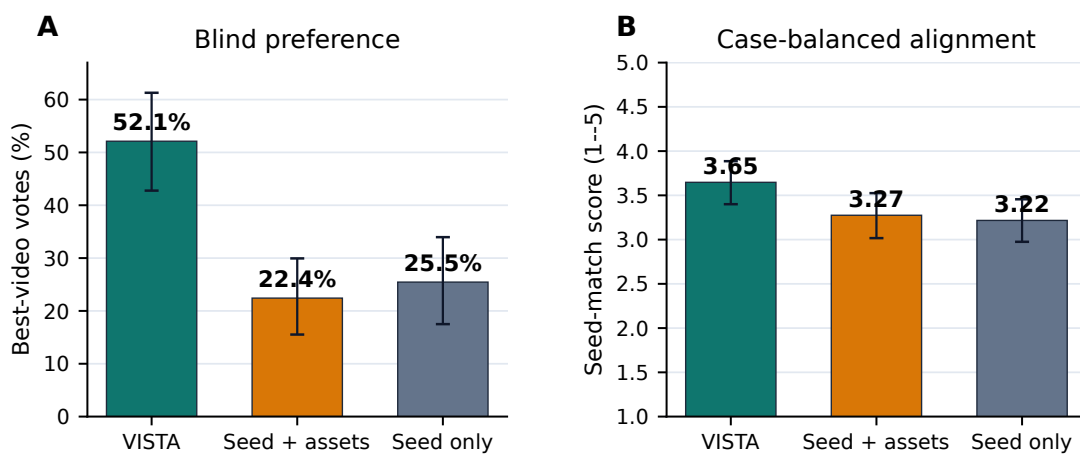


Figure 4: **Blinded output evaluation over 60 cases and 165 judgments.** (A) Best-video vote share. (B) Case-balanced mean seed-match score (1–5). Error bars are 95% case-cluster bootstrap intervals.

Group	Cases	VISTA votes (95% CI)
Reactive	15	45.2% (30.2–61.9)
Explicit proactive	15	48.8% (29.3–68.4)
Implicit proactive	15	63.4% (46.3–79.1)
No assistance	15	51.2% (31.7–71.1)
Safety-critical	14	44.7% (27.3–63.2)
Everyday inconvenience	31	55.8% (43.5–68.2)

Table 3: Descriptive subgroup coverage. Intervals overlap, samples are small, and backend varies with interaction mode; these are not tests of category effects.

5.3 Results

Figure 4 shows that VISTA receives 86/165 best-video votes, or 52.1% (95% CI 42.8–61.3), compared with 37/165, or 22.4% (15.5–29.9), for Seed+assets and 42/165, or 25.5% (17.5–34.0), for Seed only. The corresponding case-balanced scores are 3.65 (3.40–3.89), 3.27 (3.02–3.53), and 3.22 (2.97–3.46). The VISTA differences are +0.37 (0.06–0.67; Holm-adjusted $p = 0.021$) against Seed+assets and +0.43 (0.12–0.72; $p = 0.014$) against Seed only.

The aggregate result supports the complete human-reviewed workflow under the tested renderers. It does not isolate scenario compilation: VISTA includes human review and candidate selection, whereas both baselines are one-pass. It also does not measure physical realism, constraint-by-constraint control, or downstream assistant behavior.

6 Audit Record and Replay Boundaries

VISTA distinguishes a reproducible request from a reproducible provider output. Table 4 lists the paper-facing record retained at each boundary. For a planning call, the trace stores the template name, filled prompt, upstream artifact replacements, model and endpoint, temperature, raw response, generation identifier, reported cost, and typed canonical object. For media, the archived request records dimensions, duration, audio flag, reference assets, and model/provider identity; embedded image bytes and secrets are redacted, while

Boundary	Persisted record
Planning stage	template and filled prompt; upstream JSON; model, endpoint, temperature; raw response and typed artifact; cost metadata
Human revision	base revision and target scope; reviewer request; proposed YAML and unified diff; validation; acceptance/discard and invalidations
Media attempt	redacted request; model/provider; duration, size, audio and reference IDs; job and polling history; usage, failure, and candidate
Evaluation	snapshot time; 180-path condition-map audit; case/reviewer coverage; bootstrap seed/count; paired-test and agreement outputs

Table 4: Audit records separate model proposals, human decisions, provider execution, and paper analysis.

the job, polling history, usage, terminal status, and output path remain separate.

Existing typed planning artifacts can be compiled again into `render_script.yaml`, first-frame prompts, and provider request plans in dry-run mode without calling image or video services. The VISTA Agent workspace is likewise a disposable copy: its conversational events and diff can be inspected without granting write access to source media or repository files. Reproducing pixels is a different claim because it requires commercial providers whose behavior can evolve; the record therefore supports diagnosis and request-level replay, not bit-identical video regeneration.

Figure 3 illustrates the public receipt boundary. The prepared example asks to reframe a reactive case as explicit proactive while preserving its visible objects, first-person viewpoint, and reviewed outcome. The replay exposes the requested full-chain scope, current and proposed labels, human-accepted status, revision identifier, and the fact that remote execution is absent. It does not present the proposed label change as a newly rendered or validated video.

7 System Availability

The [public Hugging Face Space](#) and [direct route](#) provide a prepared replay of 14 reviewed cases and a [screencast](#). Its repository exposes only the curated replay files. The artifact accepts no provider keys, executes no models, creates no shared sessions, and releases neither the 60-case videos nor annotations. Because the broader code has no explicit software license, we call this an inspectable hosted preview rather than an open-source release.

8 Conclusion

VISTA makes egocentric assistance-scenario generation an inspectable authoring workflow. Its scenario model separates interaction mode from consequence, its compiler exposes typed intermediate artifacts, and its human gate preserves a trace from seed to selected media. In the output-only study, evaluators preferred the human-reviewed VISTA candidate and rated it as more faithful to the seed than two one-pass baselines under the tested renderers. The evidence supports the integrated workflow, not a novel planning algorithm or a claim about interface usability. Future work should isolate pipeline stages, hold renderer choice fixed, and evaluate authoring tasks and downstream assistance directly.

Acknowledgments

This research was partially supported by National Science and Technology Council, Taiwan, under grants NSTC 114-2221-E-A49-057-MY3, NSTC 114-2221-E-002-070-MY3, and NSTC 115-2634-F-002-012-, and Ministry of Education (MOE) in Taiwan, under grants 115L900901.

Limitations

What the study measures. The experiment is an output-fidelity study, not a system-usability study. Evaluators never authored a case in VISTA, so the paper provides no evidence about task completion, authoring time, learning curve, revision count, proposal-acceptance rate, satisfaction, or comparative efficiency. Seed-match judgments also do not directly test per-constraint controllability, physical realism, temporal precision, or the quality and timing of a downstream assistant’s intervention. We did not run an ablation or a head-to-head study with another scenario-authoring tool. Accordingly, the results should not be interpreted as showing that VISTA is easier to use, faster, or safer than an alternative platform.

Scale and agreement. The 60 cases and 11 CS-majoring evaluators do not exhaust daily assistance needs, author backgrounds, cultures, or environments. Review counts vary between two and three per case, and Fleiss’ $\kappa = 0.215$ on triplicated best-video choices indicates low-to-modest agreement. Subgroup samples are small and their bootstrap intervals overlap; their reported differences are descriptive coverage checks, not category effects. A

larger, more diverse pool of scenario authors and evaluators is needed.

Comparison and renderer confounds. The VISTA condition includes authoring-stage human review and candidate selection, while Seed+assets and Seed only are one-pass. The study therefore measures the end-to-end workflow rather than the isolated effect of compilation or any one stage. Although all conditions share a backend within each case, explicit and implicit cases use Seedance 2.0 whereas reactive and no-assistance cases use Sora 2. Renderer and interaction mode are consequently confounded. Future experiments should hold the renderer fixed, match human selection effort across conditions, and ablate planning, assets, revision, and selection.

Reproducibility and release. Commercial provider behavior can change even when model strings remain fixed. The archived planning calls record model, endpoint, temperature, prompt, response, and cost metadata, but media endpoints expose no temperature or top- p in the study payload and provider-side defaults or silent revisions are not under our control. The public Space is a 14-case prepared replay: it does not run generation or publish the 60 evaluation videos, anonymized judgment snapshot, complete benchmark, or a licensed full-code release. This protects credentials and unreleased research artifacts but limits independent reuse. A versioned, licensed software snapshot and appropriately consented evaluation artifacts would materially improve reproducibility.

Ethical Considerations

Synthetic scenarios avoid deliberately staging hazards with real participants, but generated depictions can reproduce social, cultural, environmental, and provider biases. Safety cases are fictional test scenarios, not evidence about the frequency of real hazards or certification of an assistive system. Everyday-inconvenience cases and no-assistance controls reduce the incentive to make every scene dangerous or intervention-worthy.

The authoring system separates request preparation from a provider call, requires explicit authorization, and retains the provider boundary and attempt history. Reviewers remain responsible for provider terms and for screening media before distribution. The paper-facing evaluation export is

anonymized: it contains case labels and judgments but excludes reviewer names, email addresses, credentials, hidden prompts, and local paths. The public replay similarly accepts no credentials and exposes no shared session state or underlying benchmark files.

References

- Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, and 66 others. 2022. [Ego4D: Around the world in 3,000 hours of egocentric video](#). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012.
- Xuan He, Dongfu Jiang, Ge Zhang, Max Ku, Achint Soni, Sherman Siu, Haonan Chen, Abhranil Chandra, Ziyang Jiang, Aaran Arulraj, Kai Wang, Quy Duc Do, Yuansheng Ni, Bohan Lyu, Yaswanth Narsupalli, Rongqi Fan, Zhiheng Lyu, Bill Yuchen Lin, and Wenhu Chen. 2024. [VideoScore: Building automatic metrics to simulate fine-grained human feedback for video generation](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2105–2123. Association for Computational Linguistics.
- David Heineman, Yao Dou, and Wei Xu. 2023. [Thresh: A unified, customizable and deployable platform for fine-grained text evaluation](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 336–345. Association for Computational Linguistics.
- Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. 2022. [Video diffusion models](#). *Preprint*, arXiv:2204.03458.
- Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, Yaohui Wang, Xinyuan Chen, Ying-Cong Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. 2024. [VBench++: Comprehensive and versatile benchmark suite for video generative models](#). *Preprint*, arXiv:2411.13503.
- Andrew Li, Zhenduo Wang, Ethan Mendes, Duong Minh Le, Wei Xu, and Alan Ritter. 2024. [ChatHF: Collecting rich human feedback from real-time conversations](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 270–279. Association for Computational Linguistics.
- Francesco Ragusa, Michele Mazzamuto, Rosario Forte, Irene D’Ambra, James Fort, Jakob Engel, Antonino Furnari, and Giovanni Maria Farinella. 2026. [Ego-EXTRA: Video-language egocentric dataset for](#)

[expert-trainee assistance](#). In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4438–4450.

Martín Santillán Cooper, Zahra Ashktorab, Hyo Jin Do, Erik Miehl, Werner Geyer, Jasmina Gajcin, Elizabeth M. Daly, Qian Pan, and Michael Desmond. 2025. [Synthetic data for evaluation: Supporting LLM-as-a-judge workflows with EvalAssist](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 1–11. Association for Computational Linguistics.

Kaiyue Sun, Kaiyi Huang, Xian Liu, Yue Wu, Zihan Xu, Zhenguo Li, and Xihui Liu. 2024. [T2V-CompBench: A comprehensive benchmark for compositional text-to-video generation](#). *Preprint*, arXiv:2407.14505.

Xiaofeng Wang, Kang Zhao, Feng Liu, Jiayu Wang, Guosheng Zhao, Xiaoyi Bao, Zheng Zhu, Yingya Zhang, and Xingang Wang. 2024. [EgoVid-5M: A large-scale video-action dataset for egocentric video generation](#). *Preprint*, arXiv:2411.08380.

Xin Wang, Taein Kwon, Mahdi Rad, Bowen Pan, Ishani Chakraborty, Sean Andrist, Dan Bohus, Ashley Feniello, Bugra Tekin, Felipe Vieira Frujeri, Neel Joshi, and Marc Pollefeys. 2023. [HoloAssist: An egocentric human interaction dataset for interactive AI assistants in the real world](#). In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20270–20281.

Yichi Zhang, Xin Luna Dong, Zhaojiang Lin, Andrea Madotto, Anuj Kumar, Babak Damavandi, Joyce Chai, and Seungwhan Moon. 2025. [Proactive assistant dialogue generation from streaming egocentric videos](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 12044–12068. Association for Computational Linguistics.