Bonsai: Cultivating Author Intent in LLM-Based Interactive Digital Narratives

Tiffany Wang [*1] **Max Kreminski** [*2]

Abstract

Authors of LLM-based interactive digital narratives (IDNs) struggle to preserve creative intent as player choices and real-time generation pull storylines in unpredictable directions. Existing frameworks treat IDNs as static once published, limiting authors' insight into and control over the storylines that emerge from unexpected player input in the wild. We propose *cultivation*: a design metaphor in which LLM-generated branches are stored as persistent material for authors to shape through iterative curation. Authors seed an initial scenario; the system grows new branches in response to player exploration; authors prune and revise what emerges, accumulating preference data that steers future generation. We complement this with three empirical probes showing that learned preferences transfer to unseen scenes, are project-specific rather than portable, and improve substantially when extraction is structured around IDN authoring categories. We demonstrate cultivation through *Bonsai*, an IDN authoring tool, and reflect on how this metaphor reframes human-AI creative collaboration: authors become gardeners, tending ever-growing branches rather than constraining ephemeral outputs.

1. Introduction

Designers of interactive digital narratives (IDNs) must negotiate a three-way tension between author, player, and system control over the narrative's direction (Jones & Millard, 2024; Mason, 2021). Imagine you've written a sword-and-sorcery fantasy where the player must defeat a dragon—but during playtesting, a player decides they'd rather befriend the dragon than slay it. To stray from the author's pre-defined paths is fruitless in conventional IDNs such as *Choose Your Own Adventure*, but empowered in LLM-based IDNs such as *AI Dungeon* (Walton, 2019). LLMs grant IDNs the ability to improvisationally "play along" to the player's potentially unusual requests. However, this comes at a cost to authorial influence—the author who dislikes the improvisation finds the LLM difficult to constrain (Toyer et al., 2023), and the author who likes the improvisation is not consulted, and cannot preserve it consistently for future playthroughs.

Pure LLM improvisation instantiates divergent, parallel universes with no shared reality underneath. For authors, the gap between their intent and realized player experience expands. LLM stochasticity produces outputs that are neither stable nor predictable; LLMs drift, pulled by a mix of player tangents and their own improvisation (Beguš, 2024; Tian et al., 2024), making it difficult for authors to anticipate what may emerge across playthroughs. Steering through prompting alone is cognitively burdensome and relies on brittle heuristics (Zamfirescu-Pereira et al., 2023). For players, an IDN that inconsistently responds the same inputs erodes both the value of replaying (Mitchell, 2025) and the social practice of sharing experiences through retellings (Eladhari, 2018; Kreminski et al., 2019). The stochastic, emergent nature of playthroughs undermines both author and player trust; the *choice poetics* (Mawhorter et al., 2014) the author constructs and the predictability of consequences experienced by the player (Wardrip-Fruin et al., 2009).

Existing authoring frameworks treat IDNs as static artifacts once they are published, limiting authors' insight into and control over the storylines that emerge from player interaction. Adapting Kreminski & Wardrip-Fruin's delineation between *mining* and *gardening* dynamics in procedural generation, we argue that current LLM-based IDNs follow mining dynamics—where playthrough content is generated on-demand and discarded upon consumption. Gardening, alternatively, treats generated artifacts as *non*-disposable, fostering instead sustained, deep engagement with a single, gradually evolving artifact (Kreminski & Wardrip-Fruin, 2018a). We propose *cultivation* as a gardening-oriented alternative design metaphor: authors correct generations as they emerge through play, edits accumulate as intent rules, and alignment with authorial intent improves over time—shaping not only individual branches but future improvisations.

We make four contributions:

[*]Equal contribution [1]Midjourney, San Francisco, CA [2]Cornell Tech, New York, NY. Correspondence to: Tiffany Wang <twang@midjourney.com>, Max Kreminski <mkremins@cornell.edu>.

ICML'26 Workshop on Human-AI Co-Creativity, Seoul, South Korea.

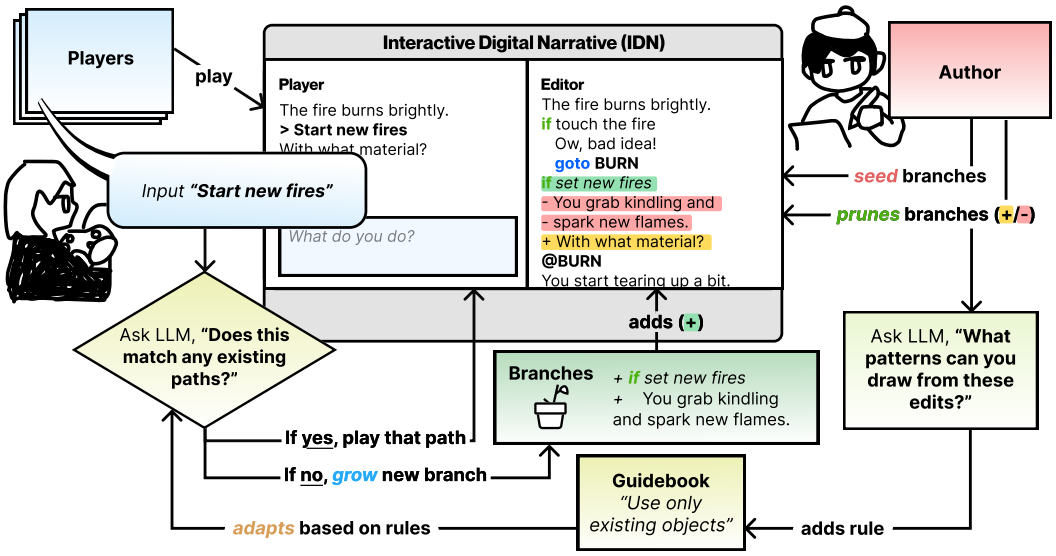

Figure 1. Overview of Bonsai's cultivation cycle. Players input natural language actions (e.g., *"Start new fires"*), triggering similarity matching against existing paths. Novel actions generate new branches guided by the accumulated rule set. Authors asynchronously review generated content in the Editor, editing or resolving branches. Inline author edits create new versions and add learned patterns to the rule set (e.g., *"Use only existing objects"*), which inform future generation.

1. *Bonsai*, an IDN authoring tool instantiating cultivation through four phases (seeding, growing, pruning, adapting), with analysis of how each addresses IDN authoring challenges;
2. *Cultivation*, a design metaphor for LLM-based authoring in which narratives remain live after publication, accumulating design knowledge through use rather than calcifying at deployment;
3. Three simulated experiments probing edit-based preference learning in IDN authoring;
4. Design implications of the cultivation metaphor for authoring LLM-based interactive systems within and beyond IDN.

2. Related Work

Interactive digital narratives (IDNs) grant players what Murray calls "the satisfying power to take meaningful action and see the results of your decisions and choices" (Murray, 1997). Yet player agency is fundamentally limited by what different IDN authoring forms allow. For example, conventional branching IDNs (e.g. those implemented via Twine) disallow paths that were not explicitly pre-defined by the author. As narrative complexity expands, it combinatorially expands the number of paths an author must write (Jones & Millard, 2024), limiting the degree of comprehensive player agency supported by the system. Modular content structures like storylets enable easier refactoring and emergent interaction design (Kreminski & Wardrip-Fruin, 2018b; Short, 2016), but still leave authors solely responsible for defining the choices that lead to meaningful progression.

LLMs offer a potential solution to these challenges through their ability to improvise coherent responses to arbitrary player input. Early systems like AI Dungeon (Ferreira, 2025) use LLMs to generate narrative content on demand. Pure prompt-based generation introduces failure modes: models hallucinate (Kalai et al., 2025), misinterpret inputs (Liu et al., 2023), get derailed by player tangents (Gallotta et al., 2024), and lack understanding of narrative structure (Chakrabarty et al., 2024; Tian et al., 2024).

To address these consistency issues, recent systems impose additional structure on LLM generation. *Dramamancer* combines modular storylets with authored preconditions that gate critical state changes while generating narration and dialogue in real-time (Wang et al., 2025). Orchid uses upfront specifications to constrain narrative progression (Wu et al., 2025). These approaches improve coherence over pure prompt-based generation, yet require authors to anticipate player choices and specify all constraints upfront, giving authors no mechanism to learn from or correct what actually emerges across playthroughs.

Beyond IDN, creativity support tools have explored author-system alignment through preference learning from edits (Gao et al., 2024), edit-based writing rewards (Chakrabarty et al., 2025), revision-focused interfaces (Zhou & Sterman, 2024), and multimodal intent capture (Chung et al., 2022; 2025). These improve creative preference alignment in solo creative writing, but none ad-

dresses the triadic challenge of IDN authoring: balancing author intent, player agency, and system behavior simultaneously, across multiple playthroughs. No prior work has applied Kreminski & Wardrip-Fruin's gardening orientation (Kreminski & Wardrip-Fruin, 2018a) to IDN authoring with generative AI.

3. Design of Bonsai

Bonsai is a web-based IDN authoring system built around *cultivation*, using Claude 3.5 Haiku for all generation tasks within the deployed tool. Later empirical evaluations use larger models (Sonnet 4.6, Opus 4.7, GPT-5) to isolate the contribution of the cultivation mechanism; results may differ under the deployed model.

Seeding from initial author content. Authors construct narrative material using a scripting language inspired by Ink (Inkle Studios, 2016), a widely-adopted markup language for narrative games used in numerous award-winning titles (Inkle Studios, 2026). Scripts are composed of scenes and *branches*, which are pairs of conditions and consequences, e.g. "if Alice eats the cake *then* Alice grows larger". The language also supports basic narrative control flow including state and scene changes.

Growing from player interaction. When a player submits an input, an LLM is used to fuzzily match it against existing branch conditions within the current scene. If confidence exceeds 0.7, the player is routed to the matching branch, experiencing it exactly as written. Below that threshold, the LLM generates a new branch conditioned on narrative context and player input, which immediately persists to the artifact. Generations are strictly additive, and existing authored content is never overwritten.

Pruning generated branches. Authors review generated content via a unified version history with attribution ("AI" vs. "author"). In the authoring interface, freshly generated lines are highlighted in yellow until they are edited by the author (highlighted in green) or resolved.

Learning from author edits. When authors edit generated content, an LLM extracts a natural-language author-intent rule from the diff (e.g. softening a violent outcome produces "prefer non-violent resolutions"). Rules accumulate in a visible guidebook and condition future branch generation, so each author edit changes how the artifact grows.

To illustrate an example of a full cultivation cycle: A player types *"pick the lock"*, to which the LLM replies *"You don't have anything to pick a lock with yet."* The author deletes the *"yet"* as it creates a red herring, implying the player might find a lock-picking tool later. The system learns: *"Do not mention the absence of objects not already in the scene."* Every subsequent branch is conditioned on this rule, so future generations stay within the established object inventory. Section 5.1 traces two more full examples.

3.1. Design Process

We implemented Bonsai through three months of first-person research (Lucero et al., 2019) informed by autobiographical design (Zimmerman et al., 2007; Gaver, 2012), drawing on our experience as IDN authors with published works played tens of thousands of times. This process produced three design commitments: generated branches should persist as authoring material, authors need direct controls over generative scope, and adaptation should learn from inline rewrites rather than approvals or deletions.

3.2. Observations from Exploratory Testing

These qualitative observations come from formative, exploratory testing, rather than any controlled user study. Over three months of first-person and autobiographical design, we iterated on seeded IDN scripts with three testers with backgrounds in AI and creative writing. Testers interacted with partially seeded projects, primarily `fireplace` and `escaperoom`, by entering player actions, inspecting generated branches, and discussing or revising system outputs.

Several observations were formative: they shaped the current Bonsai design rather than evaluating it after the fact. In `fireplace`, testers valued seeing when unusual player inputs matched authored branches, motivating visible semantic-match confidence. In early branch review, generated branches appeared separate from author edits, which made it hard to tell what was unresolved; this led to the unified version history with generated content highlighted. We also found that approvals and deletions were noisy preference-learning signals, because authors might approve provisionally or delete material for reasons unrelated to intent, so Bonsai learns only from inline rewrites.

Other observations expose limits that remain relevant to the current system. In `escaperoom`, unconstrained scene growth could undermine puzzle structure, motivating the freedom slider as a way to trade narrative flexibility for puzzle integrity. The same project also produced a failure we called "vector similarity hell": a compelling generated hidden compartment became overmatched to unrelated later actions, including inspecting unrelated objects and using the key. Manual pruning fixed that instance, but the broader problem remains only partially addressed; Bonsai still depends on authors noticing when semantic matching has made an appealing branch too broadly active. These observations suggest that Bonsai redistributes authorial labor over time rather than eliminating it. Asynchronous review made correction possible in these small seeded projects, but larger projects may require stronger triage for persistent generated branches at scale.

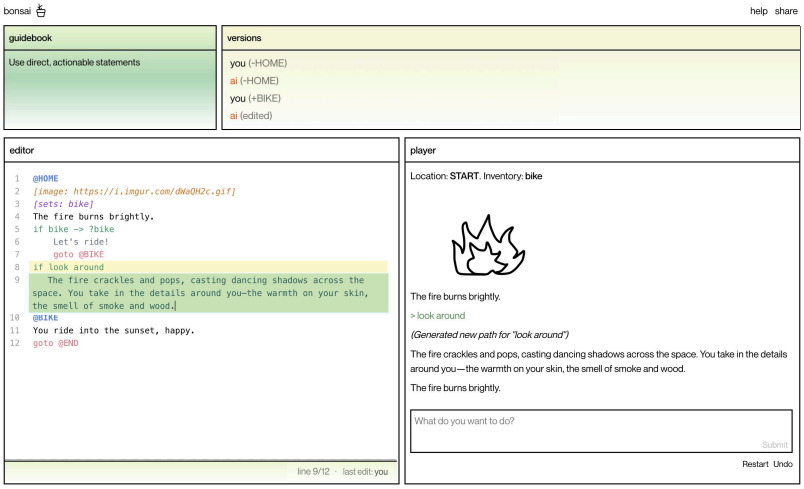

Figure 2. The Bonsai authoring interface. The Editor (left) shows markup with diff highlighting: yellow for LLM-generated content, green for author edits. The Player panel (right) renders the story with natural language input. The Versions panel (top-right) tracks changes chronologically; the Guidebook (top-left) holds accumulated author-intent rules.

4. Methods

We evaluate on three projects:

1. `fireplace`: absurdist, permissive world logic
2. `lifesim`: minimalist, variable-driven consequences
3. `escaperoom`: deadpan, hard puzzle constraints

Each contains a *seed script*—the initial scenario authored in markup before any play—and *simulated author preferences*: an informal intent document (see Appendix F) written in the voice of a real author specifying what kinds of branches are right or wrong for their world. Each project also has two evaluation scenes: a construction scene used for rule-building/edit examples, and a held-out scene used as a small transfer check.

We use Claude Sonnet 4.6 and GPT-5 for generation, Claude Haiku 4.5 for author edit simulation, and Claude Opus 4.7 and GPT-5 as author preference raters. Full configurations are in Appendix A and prompts are in Appendix G. We rely on LLM raters rather than human raters and mitigate this in three ways: dual cross-vendor raters (Opus + GPT-5) reduce single-model bias; the rater-leak diagnostic (Appendix I) shows that PREFERENCE scores reflect semantic alignment rather than lexical overlap with the preference document; and chance-corrected and leak-free variants are reported throughout to bound the contribution of evaluation artifacts. We treat these as upper-bound pilots motivating the future human-author studies described in Section 6.

RQ1: Do author-intent rules produce a meaningful, project-specific preference signal? We generate outputs under four conditions and rate each:

1. *none*: No rules are given.
2. *style*: Voice/tone rules extracted from the project's seed prose are given (Appendix E); no plot, mechanics, or intent rules are given.
3. *intent*: Author-intent rules parsed from the project's simulated author preferences are given; unlike *style*, these may include voice, world, plot, and interaction constraints. This condition tests whether explicit intent rules can steer generation, not whether those rules have been recovered from edits.
4. *wrong_intent*: Author-intent rules from a different author's project are given.

Each generation is rated on a 1–5 scale along three dimensions: *style*, how well it matches the author's *voice* shown in the seed script; *preference*, how well it follows the simulated author's stated preferences; and *consistency*, how well it fits the world established by the seed script. Only the *preference* rater sees the simulated author preferences; *style* and *consistency* are rated from the seed script alone. As a representation baseline, we also compare extracted rules against few-shot prompting with the same edit examples.

For each condition and scene split, we sample three player inputs per project and generate two responses per input, yielding 3 inputs $\times$ 2 trials $\times$ 3 projects $=$ 18 generations, each rated by two raters.

RQ2: Does the signal transfer to held-out scenes? Using the same rating pipeline, we compare the construction scene against one held-out scene for each project. This provides a small scene-level transfer check: whether rules improve ratings beyond the initial scene from which they

were inferred.

RQ3: How does extractor design shape preference recovery from edits? We first categorize the simulated author preference rules parsed from each preference document using Orchid (Wu et al., 2025), yielding a roughly balanced set across *world* (world-settings and characters), *narrator* (narrator behavior), and *player* (player interactions) rules (12, 14, and 9 rules, respectively). We then compare naive extraction with category-aware extraction. Naive extraction asks for one rule per edit; category-aware extraction separately asks whether each edit reveals a *world*, *narrator*, or *player* rule. Recovery is the fraction of simulated author preferences matched by extracted rules. Because category-aware extraction emits more rules, raw recovery can improve simply from having more opportunities to match a preference, so we compare recovery against a rule-count-matched chance baseline.

5. Results

5.1. Two Worked Traces

On the `fireplace` project, the simulated author's preferences specify that the fire is "a minor god" and that branches should avoid safe, deflecting responses. For the input *"talk to the fire"*, the system first generates a normal continuation about a fire that "doesn't have much to say." The simulated author rewrites this into an absurdist version where the fire shrieks and you "realize with horror that it's been WAITING for someone to acknowledge it." Bonsai extracts the rule: *"...escalate dramatically... give elements with agency explicit opinions and reactions rather than leaving them open-ended or restrained."* This illustrates how an edit can become a reusable constraint for later generation.

A second trace shows intent that is less stylistic. In `escaperoom`, the input *"pick the lock"* first produces: "You don't have anything to pick a lock with, and there's no lock in sight anyway." The simulated author deletes the second clause, leaving only: "You don't have anything to pick a lock with." Bonsai extracts the rule: *"Do not mention the absence of objects or game elements...this inadvertently introduces new concepts (like 'a lock')"*. This illustrates an IDN-specific preference: edits can encode interaction constraints, not only prose voice.

5.2. RQ1: Do author-intent rules produce a meaningful, project-specific preference signal?

Table 1 supports the preference signal in three ways: intent rules lift *preference* sharply, reaching +2.56 on held-out scenes; style rules produce the largest *style* lift (+0.64 held-out) but only a small *preference* lift (+0.39); and wrong-project rules hurt *preference* (-0.44 held-out), confirming the signal is project-specific. In a separate pairwise representation baseline, learned rules extracted from edit examples beat few-shot prompting with the same examples in 92% of comparisons (Appendix D). Intent rules reduce consistency (-0.81 / -0.33), revealing a voice–intent tradeoff: the seed script is an incomplete proxy for author intent, and explicit preferences can deliberately move generations beyond the seed's initial voice. For co-creative authoring, it marks the point where captured author intent can revise the artifact instead of extrapolating past it.

Effects vary by project. `fireplace` shows the largest lift (+3.83 / +3.67), consistent with expressive voice preferences being easy to express as rules. `lifesim` is smaller on the construction scene but strong held-out (+1.08 / +3.25), while `escaperoom` remains modest (+0.58 / +0.75), suggesting that mechanical puzzle constraints are harder to capture with global intent rules.

5.3. RQ2: Does the signal transfer to held-out scenes?

On the same rating grid, *preference* lift remains positive on the held-out set, even exceeding the construction scenes (+2.56 vs. +1.83). This suggests that the effect is not limited to the inputs used to construct the rules. Since this check covers only one held-out scene per project with hand-chosen inputs, we treat it as limited evidence of scene-level transfer rather than broad generalization.

5.4. RQ3: How does extractor design shape preference recovery from edits?

The recovery task spans IDN-specific preference types: *world* constraints, *narrator* voice preferences, and *player* interaction preferences. Naive extraction mostly recovers *narrator* preferences and nearly misses *world* constraints (64% vs. 8%). Category-aware extraction raises *world* recovery to 83% and *narrator* to 86%, but at the cost of emitting more rules (28 vs. 12).

Chance-corrected analysis (Appendix J) is essential here: because more rules create more chances to match a target, raw recovery can rise from rule-count inflation alone. The key metric is lift over the rule-count-matched chance ceiling. Naive extraction *undershoots* chance on world (-25 pp) and player (-22 pp) while exceeding it on narrator ($+35$ pp). Category-aware extraction recovers world above chance in both preference-doc-guided ($+5$ pp) and—crucially—leak-free ($+14$ pp, the strongest world lift of any configuration) settings, confirming that category prompting provides genuine structural recovery that survives removal of preference-document access.

Player interaction preferences remain the hardest category: they drop to 0% under leak-free naive extraction and only reach 44% under leak-free category-aware. These rules govern what player actions are possible, redirected, or

	PREFERENCE		STYLE		CONSISTENCY	
Condition	train	held-out	train	held-out	train	held-out
none (abs.)	3.17	2.31	3.97	3.22	4.42	3.75
style	+0.33	+0.39	**+0.42**	**+0.64**	**+0.28**	**+0.14**
intent	**+1.83**	**+2.56**	−0.61	0.00	−0.81	−0.33
wrong_intent	−0.86	−0.44	−0.72	−0.28	−0.86	−0.94

Table 1. Mean ratings by condition. The *none* row reports raw 1–5 scores; all other rows report change relative to *none*. Columns split construction and held-out scenes. Bold marks the best value within each metric.

prevented—the most IDN-specific constraint type—and are often implicit in prose edits (an author deletes a line rather than rewriting it, leaving little positive signal to extract). Improving player-preference recovery is an open problem we flag for future work.

6. Conclusion

We explore the design metaphor of *cultivation*—treating AI-generated IDN branches as persistent, editable material from which author preferences can be learned. Across three simulated projects, we parse intent rules from simulated per-project preferences, including both narrator voice and world logic constraints. Cultivation reframes correction as a reusable creative signal: authors directly and permanently correct unwanted improvisation, implicitly leaving alone what they'd like to preserve, and use edits to teach the artifact how to grow in a way that aligns to their creative intent. By reducing the friction of creative alignment, authors are free to explore richer generative possibility spaces without ceding control over the artifact's direction.

Limitations. The evaluations of author preferences and edits are simulated, the quantitative ratings are model-based, and the sample is small: three projects with one construction and one held-out scene each, each backed by a single fully-specified preference document rather than the varied, evolving, and possibly under-elicited preferences of real authors. The main recovery study is an upper bound, as the simulated author and extractor see the simulated author's ground-truth preferences. The leak-free variant partially addresses this.

Future work. Future work should formally evaluate Bonsai with a wider array of human IDN authors, measuring not only output alignment but creative control, ownership, effort, surprise, retention, trust, and willingness to keep using the system. Bonsai evaluations can also be technically scaled with multiple raters and preference-blind checks. Extractors must improve on recovering *player* interaction preferences; the rules governing what actions are possible, redirected, encouraged, or prevented are among the most IDN-specific and hardest to recover. Larger projects also require retrieval or routing over rules so accumulated preferences remain useful without over-conditioning every generation.

References

Beguš, N. Experimental narratives: A comparison of human crowdsourced storytelling and ai storytelling. *Humanities and Social Sciences Communications*, 11(1), October 2024. ISSN 2662-9992. doi: 10.1057/s41599-024-03868-8. URL http://dx.doi.org/10.1057/s41599-024-03868-8.

Chakrabarty, T., Laban, P., Agarwal, D., Muresan, S., and Wu, C.-S. Art or artifice? large language models and the false promise of creativity, 2024. URL https://arxiv.org/abs/2309.14556.

Chakrabarty, T., Laban, P., and Wu, C.-S. Ai-slop to ai-polish? aligning language models through edit-based writing rewards and test-time computation, 2025. URL https://arxiv.org/abs/2504.07532.

Chung, J. J. Y., Kim, W., Yoo, K. M., Lee, H., Adar, E., and Chang, M. Talebrush: Sketching stories with generative pretrained language models. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391573. doi: 10.1145/3491102.3501819. URL https://doi.org/10.1145/3491102.3501819.

Chung, J. J. Y., Roemmele, M., and Kreminski, M. Toyteller: Ai-powered visual storytelling through toy-playing with character symbols. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, pp. 1–23. ACM, April 2025. doi: 10.1145/3706598.3713435. URL http://dx.doi.org/10.1145/3706598.3713435.

Eladhari, M. P. Re-tellings: the fourth layer of narrative as an instrument for critique. In *International Conference on Interactive Digital Storytelling*, pp. 65–78. Springer, 2018.

Ferreira, C. Genre, Bias, and Narrative Logic in AI Dungeon: Generative AI as a Game-Based Storytelling Engine. *Hipertext.net*, (31):77–89, November 2025. ISSN 1695-5498. doi: 10.31009/hipertext.net.2025.i31.08. URL https://raco.cat/index.php/Hipertext/article/view/433301.

Gallotta, R., Todd, G., Zammit, M., Earle, S., Liapis, A., Togelius, J., and Yannakakis, G. N. Large language models and games: A survey and roadmap. *IEEE Transactions on Games*, pp. 1–18, 2024. ISSN 2475-1510. doi: 10.1109/tg.2024.3461510. URL http://dx.doi.org/10.1109/TG.2024.3461510.

Gao, G., Taymanov, A., Salinas, E., Mineiro, P., and Misra, D. Aligning llm agents by learning latent preference from user edits, 2024. URL https://arxiv.org/abs/2404.15269.

Gaver, W. What should we expect from research through design? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 937–946, 2012.

Inkle Studios. Ink: Inkle's narrative scripting language. https://www.inklestudios.com/ink/, 2016. Accessed: February 2, 2026.

Inkle Studios. inkle/ink-library: A collection of ink samples, tools and a list of projects that use ink. https://github.com/inkle/ink-library?tab=readme-ov-file#ink-games-and-non-games, 2026. Accessed: February 3, 2026.

Jones, J. D. and Millard, D. Experiencing the authorial burden. In *Proceedings of the 35th ACM Conference on Hypertext and Social Media*, HT '24, pp. 78–87, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400705953. doi: 10.1145/3648188.3675134. URL https://doi.org/10.1145/3648188.3675134.

Kalai, A. T., Nachum, O., Vempala, S. S., and Zhang, E. Why language models hallucinate, 2025. URL https://arxiv.org/abs/2509.04664.

Kreminski, M. and Wardrip-Fruin, N. Gardening games : An alternative philosophy of pcg in games. In *FDG Workshop on Procedural Content Generation in Games*, 2018a.

Kreminski, M. and Wardrip-Fruin, N. *Sketching a Map of the Storylets Design Space*, pp. 160–164. 12 2018b. ISBN 978-3-030-04027-7. doi: 10.1007/978-3-030-04028-4_14.

Kreminski, M., Samuel, B., Melcer, E., and Wardrip-Fruin, N. Evaluating AI-based games through retellings. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 15, pp. 45–51, 2019.

Liu, Y., Han, T., Ma, S., Zhang, J., Yang, Y., Tian, J., He, H., Li, A., He, M., Liu, Z., Wu, Z., Zhao, L., Zhu, D., Li, X., Qiang, N., Shen, D., Liu, T., and Ge, B. Summary of chatgpt-related research and perspective towards the future of large language models. *Meta-Radiology*, 1(2): 100017, 2023. ISSN 2950-1628. doi: https://doi.org/10.1016/j.metrad.2023.100017. URL https://www.sciencedirect.com/science/article/pii/S2950162823000176.

Lucero, A., Desjardins, A., Neustaedter, C., Höök, K., Hassenzahl, M., and Cecchinato, M. E. A sample of one: First-person research methods in HCI. In *Companion Publication of the 2019 Designing Interactive Systems Conference*, pp. 385–388, 2019.

Mason, S. *Responsiveness in Narrative Systems*. PhD thesis, UC Santa Cruz, 2021. URL https://escholarship.org/uc/item/9gq229h4. ProQuest ID: Mason_ucsc_0036E_12374.

Mawhorter, P., Mateas, M., Wardrip-Fruin, N., and Jhala, A. Towards a theory of choice poetics. In *In Proceedings of the 9th International Conference on the Foundations of Digital Games*, 2014. URL http://fdg2014.org/papers/fdg2014_paper_19.pdf.

Mitchell, A. Understanding the kaleidoscopic nature of interactive digital narratives through repeat experience. In *International Conference on Interactive Digital Storytelling*, pp. 1–14. Springer, 2025.

Murray, J. H. *Hamlet on the Holodeck: The Future of Narrative in Cyberspace*. The Free Press, New York, 1997.

Short, E. Beyond branching: Quality-based and salience-based narrative structures. https://emshort.blog/2016/04/12/beyond-branching-quality-based-and-salience-based-narrative-structures/, April 2016. Accessed: 2026-01-15.

Tian, Y., Huang, T., Liu, M., Jiang, D., Spangher, A., Chen, M., May, J., and Peng, N. Are large language models capable of generating human-level narratives?, 2024. URL https://arxiv.org/abs/2407.13248.

Toyer, S., Watkins, O., Mendes, E. A., Svegliato, J., Bailey, L., Wang, T., Ong, I., Elmaaroufi, K., Abbeel, P., Darrell, T., Ritter, A., and Russell, S. Tensor Trust: Interpretable prompt injection attacks from an online game, 2023. URL https://arxiv.org/pdf/2311.01011.pdf.

Walton, N. AI Dungeon 2, 2019. URL https://aidungeon.io/. Accessed: February 2, 2026.

Wang, T., Sun, Y., Wang, Y., Roemmele, M., Chung, J. J. Y., and Kreminski, M. Dramamancer: Interactive narratives with llm-powered storylets. In *Adjunct Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*, UIST Adjunct '25, New York, NY, USA, 2025. Association for Computing Machinery. ISBN

9798400720369. doi: 10.1145/3746058.3758995. URL https://doi.org/10.1145/3746058.3758995.

Wardrip-Fruin, N., Mateas, M., Dow, S., and Sali, S. Agency reconsidered. In *Proceedings of DiGRA 2009 Conference: Breaking New Ground: Innovation in Games, Play, Practice and Theory*, 2009.

Wu, Z., Kumyol, S., Wong, S. Y., Hu, X., Tong, X., and Braud, T. Orchid: A creative approach for authoring llm-driven interactive narratives. In *Proceedings of the 2025 Conference on Creativity and Cognition*, C&C '25, pp. 774–791, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400712890. doi: 10.1145/3698061.3726906. URL https://doi.org/10.1145/3698061.3726906.

Zamfirescu-Pereira, J., Wong, R. Y., Hartmann, B., and Yang, Q. Why johnny can't prompt: How non-ai experts try (and fail) to design llm prompts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450394215. doi: 10.1145/3544548.3581388. URL https://doi.org/10.1145/3544548.3581388.

Zhou, D. and Sterman, S. Ai.llude: Investigating rewriting ai-generated text to support creative expression. In *Creativity and Cognition*, pp. 241–254. ACM, June 2024. doi: 10.1145/3635636.3656187. URL http://dx.doi.org/10.1145/3635636.3656187.

Zimmerman, J., Forlizzi, J., and Evenson, S. Research through design as a method for interaction design research in hci. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pp. 493–502, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595935939. doi: 10.1145/1240624.1240704. URL https://doi.org/10.1145/1240624.1240704.

A. Per-experiment model configurations

The empirical results draw on several separate experiments, each backed by a JSON results file. Generator, rater, simulated-author, and extractor roles differ across experiments. The table below disambiguates which model played which role in each.

Experiment	Generator	Rater(s) / simulated author
RQ1–RQ2 rating evaluation	Claude Sonnet 4.6	Claude Opus 4.7 + GPT-5 (dual rater)
Recovery extraction, naive (§5.4, RQ3)	Claude Sonnet 4.6	Claude Haiku 4.5 (simulated author + rater)
Recovery extraction, category-aware (§5.4, RQ3)	Claude Sonnet 4.6	Claude Haiku 4.5 (simulated author + rater)
Rule-availability ablation	Claude Sonnet 4.6	Claude Haiku 4.5
Pairwise baseline (learned rules vs. few-shot vs. none)	Claude Sonnet 4.6	Claude Haiku 4.5 (pairwise judge)
Cross-vendor robustness	GPT-5	Claude Opus 4.7 + GPT-5
Orchid taxonomy classification (§5.4, RQ3)	—	Claude Opus 4.6 (classifier)

The headline rating results for RQ1–RQ2 use dual cross-vendor rating, prioritizing measurement quality. The recovery experiments for §5.4 (RQ3) use a single cheaper simulated author/rater (Haiku 4.5) because they require many iterative model calls per project. The Orchid taxonomy classifier and style-rule extractor use Claude Opus 4.6 (rather than 4.7), as these tasks were run earlier in the project timeline under the then-current model version; the core rating results use 4.7. The main recovery extractors are preference-document-guided: their prompts include the project preference document, as reproduced below. The deployed Bonsai application itself uses Claude 3.5 Haiku for low-latency runtime interaction, distinct from the evaluation harness models above.

B. Per-project Orchid breakdown

§5.4 (RQ3) reports recovery rates aggregated across the three projects. Per-project counts (target / naive recovered / category-aware recovered) appear below. Naive extraction systematically under-recovers WORLD across all three projects (≤ 1 rule per project); category-aware extraction recovers WORLD in all three.

	Target			Naive recovered			Cat.-aware rec.		
Project	W	N	P	W	N	P	W	N	P
`escaperoom`	5	3	5	0	3	1	2	3	0
`fireplace`	4	6	2	0	2	1	5	5	4
`lifesim`	3	5	2	1	4	0	3	4	2
Total	12	14	9	1	9	2	10	12	6
Recovery %	—	—	—	8	64	22	83	86	67

W = *world*, N = *narrator*, P = *player*. Numbers are from `analysis/orchid_taxonomy_results.json` and `analysis/loop_simulation_categoryaware_results.json`.

C. Full RQ1 condition $\times$ dimension means

Means $\pm$ SEM across all 36 ratings per (scene, condition) cell, headline configuration (Claude Sonnet 4.6 generator, Opus + GPT-5 raters).

D. Pairwise few-shot baseline

To test whether extracted rules add value beyond simply showing the model raw edits, we compare three conditioning schemes using the same underlying edit examples: no rules, few-shot edit examples, and learned rules extracted from

Scene	Condition	STYLE	PREFERENCE	CONSISTENCY
train	none	3.97 ± 0.12	3.17 ± 0.28	4.42 ± 0.13
train	style	4.39 ± 0.12	3.50 ± 0.30	4.69 ± 0.09
train	intent	3.36 ± 0.23	5.00 ± 0.00	3.61 ± 0.26
train	wrong_intent	3.25 ± 0.20	2.31 ± 0.29	3.56 ± 0.27
held-out	none	3.22 ± 0.17	2.31 ± 0.27	3.75 ± 0.21
held-out	style	3.86 ± 0.19	2.69 ± 0.29	3.89 ± 0.20
held-out	intent	3.22 ± 0.21	4.86 ± 0.06	3.42 ± 0.25
held-out	wrong_intent	2.94 ± 0.19	1.86 ± 0.20	2.81 ± 0.25

those examples. A Claude Haiku 4.5 pairwise judge chooses the better generation for each input pair. Results are from `analysis/baseline_results.json`.

Comparison	Wins A	Wins B	A win rate
learned rules vs. none	11	2	85%
learned rules vs. few-shot	12	1	92%
few-shot vs. none	10	3	77%

E. Style rules used as the *style* condition comparator

The *style* condition uses imperative voice/tone rules extracted from each project's seed prose by Claude Opus 4.6 (with explicit instructions to avoid plot, mechanics, or intent). These rules are the comparator against which intent rules' STYLE-vs-PREFERENCE dissociation is measured.

`escaperoom`.

- Write in second-person present tense.
- Keep descriptions brief and plainspoken; favor short declarative sentences.
- Inject casual, playful interjections (e.g., "Oh lookee," "Yoink!") for a lighthearted tone.
- Use trailing ellipses and sentence fragments to create a relaxed rhythm.
- Vary formality fluidly between neutral description and breezy colloquialism.

`fireplace`.

- Write in second-person present tense.
- Keep sentences short, declarative, and unadorned.
- Use warm, sensory language for environments (light, sound, movement).
- Maintain a calm, inviting, gently optimistic tone.
- Alternate terse action statements with slightly longer descriptive passages.

`lifesim`.

- Write in second-person present tense.
- Keep sentences short, declarative, unadorned; avoid subordinate clauses.
- State life events in plain, concrete diction (no figurative or emotional language).
- Maintain a calm, matter-of-fact tone for both major and minor moments.
- Favor simple subject–verb–object structures with no connective tissue.

F. Per-project simulated author preferences

These are the hidden authorial intents that the *intent* condition parses into rules and that the preference-document-guided recovery setup uses to simulate author edits and extract upper-bound rules. They are short, informal documents, intentionally written in the voice of a real author rather than as a structured rule list.

`fireplace`.

This author wants the world to be unhinged but internally consistent. The fire is a minor god. The woods are suspicious. Player actions should be met with full commitment—if you try to befriend the fire, the fire has opinions. Tone is loud and absurdist; humor is explicit, not dry. The author wants the LLM to swing for comedy: unexpected escalations, dramatic overreactions, things going catastrophically or wonderfully wrong. Darker inputs should get darker outputs. The author actively dislikes safe, deflecting responses. A branch is wrong if it's measured, neutral, or resets to ambient description.

`escaperoom`.

This author cares about puzzle integrity. There is one win condition: finding the key on the desk and using it on the door. There should never be something blocking this path. There are only three objects. Players can't brute-force through the door. Players can have freeform interactions with any of the three objects in surprising ways (e.g., throwing the chair at the desk). Prose should be terse and slightly deadpan. A branch is wrong if it introduces a fourth object, creates an alternate escape route, or softlocks the player away from the desk.

`lifesim`.

This author values quiet observation over dramatic narration. Prose should be sparse and evocative—one precise image rather than three generic ones. Choices should feel equally valid; no path is "the good ending." Consequences should compound naturally through the variable system, but the author dislikes heavy-handed moralizing. Absurd or destructive choices should be acknowledged briefly and redirected, not punished. The world is an ordinary American small town—no magic, no fantasy. A branch is wrong if it editorializes ("you regret choosing sports over art") or resolves ambiguity that should stay open.

G. Prompts

All prompts are reproduced verbatim from the notebooks under `analysis/`. Curly braces denote interpolated fields.

Generation (rating-based eval and loop sim).

```
Generate a response for interactive fiction.
Scene:
{scene}
RULES TO FOLLOW:
- {rule 1}
- {rule 2}
...
Player input:  ``{user_input}''
Output 1-3 lines of narrative response only.  No metadata or syntax.
```

Author edit simulation.

```
You are simulating an author who has these preferences:
{preferences_text}
An AI generated this response:
Scene:  {scene}
Player input:  ``{user_input}''
Generated:  {generated}
Rewrite this response to better match your preferences.  Keep it 1-3 lines.  If it
already matches well, return it unchanged.
Your preferred version:
```

Naive rule extraction (preference-document-guided).

```
Compare these two responses and extract a rule.
GENERATED (AI): {generated}
EDITED (Author):  {edited}
```

```
AUTHOR'S PREFERENCES: {preferences}
EXISTING RULES: {existing_rules}
If the edit reveals a preference NOT already captured, propose ONE new rule.  If
the responses are similar or the rule already exists, respond with:  NO_NEW_RULE
Format:  RULE: [your rule here] or NO_NEW_RULE
```

Category-aware rule extraction (Orchid taxonomy, preference-document-guided).

```
Compare these two responses and decide what rules the edit reveals, using the
Orchid (Wu 2025) taxonomy for IDN design:
WORLD --- world-settings & characters:  which entities/objects/facts exist, plot
rails, setting constraints, canonical behavior of characters or forces.
NARRATOR --- narrator behavior:  POV, prose style, tone, diction, editorializing,
atmosphere.
PLAYER --- player interactions:  how player inputs are interpreted/acknowledged/redirected,
what kinds of actions are encouraged or discouraged, consequences.
GENERATED (AI): {generated}
EDITED (Author):  {edited}
AUTHOR'S PREFERENCES: {preferences}
EXISTING RULES: {existing}
For EACH category, ask:  does the diff between GENERATED and EDITED reveal a
preference rule in this category that is NOT already covered by EXISTING RULES?
If yes, propose ONE rule.  If not, write NONE. Do not invent rules unsupported by
the diff.
Output EXACTLY three lines:
WORLD: <rule or NONE>
NARRATOR: <rule or NONE>
PLAYER: <rule or NONE>
```

Rating (per dimension, headline eval). The rater receives the generation, the scene, and a different reference per dimension: the seed script for STYLE, the parsed simulated preferences for PREFERENCE, and the seed script alone (preference-blind) for CONSISTENCY. Rubric phrasings:

- **STYLE**: "how well this response matches the AUTHOR'S VOICE shown in this reference script. 1 = totally different voice, 5 = indistinguishable from author."
- **PREFERENCE**: "how well this response follows the AUTHOR'S PREFERENCES. 1 = violates preferences, 5 = perfectly follows preferences."
- **CONSISTENCY**: "how well this response fits the world & intent established by this seed script. Judge as if you were the author asking: 'is this consistent with what I've already written?' 1 = clashes with the established work, 5 = feels like part of the same authored piece."

The rater is preference-blind on CONSISTENCY (sees the seed but never the preferences doc), while the PREFERENCE rater sees the simulated author preferences. The rater is asked to answer with exactly `SCORE=N` and the score is parsed by regex (defaulting to 3 on parse failure).

Style-rule extraction (used to derive the *style* condition).

```
Read this interactive fiction script and extract 3-5 imperative STYLE rules about
the author's VOICE.
STRICT CONSTRAINT: Only rules about tone, sentence rhythm, word choice, diction,
POV, punctuation, formality.  DO NOT include rules about:  plot, setting, what
can/can't happen, puzzle mechanics, character behavior, the author's preferences
about game logic.
Script:  {base_script}
Output ONLY a numbered list of 3-5 style rules, one per line.  Each should start
with an imperative verb.
```

H. Cross-vendor cell-level lifts

The cell-level table backing the cross-vendor generator robustness check. PREFERENCE lift = mean(intent) – mean(none), per (project, scene, rater) cell, GPT-5 generator, $n = 6$ generations per cell per condition.

Project	Scene	Rater A (Opus)	Rater B (GPT-5)
`escaperoom`	train	+0.33	+0.00[†]
`escaperoom`	held-out	+0.83	+1.33
`fireplace`	train	+3.83	+4.00
`fireplace`	held-out	+3.67	+4.00
`lifesim`	train	+1.33	+1.67
`lifesim`	held-out	+1.67	+1.67

[†] Both *intent* and *none* saturate at 5.0 under rater B for this cell, yielding a tie rather than a positive lift. All other cells are strictly positive.

I. Rater-leak diagnostic

Fraction of 4-grams in each generation that appear verbatim in the project's `secret-preferences.md`, computed from `intent_vs_style_clean_results.json` via `analysis/leak_diagnostics.ipynb`. The conditional distribution is heavily zero-inflated, so we report the mean alongside per-trial maxima and the count of trials with any overlap. If the PREFERENCE rater were primarily rewarding lexical echo, *intent* would show a heavy right tail and a large fraction of trials with detectable overlap; the empirical right tail is short and rare, with no control trial leaking at all.

Condition	Mean	Max	Trials w/ any overlap	n
none	0.000	0.000	0	72
style	0.000	0.000	0	36
intent	0.0015	0.059	3	72
wrong_intent	0.0007	0.026	1	36

The 95th percentile is 0 in all four conditions: at least 95% of generations under every condition have zero 4-gram overlap with the preferences document. The signal is therefore not lexical—the PREFERENCE rater rewards semantic alignment, not keyword match.

J. Chance baseline for recovery %

Reported recovery is sensitive to total recovered rules: an extractor producing N rules has a uniform-classification chance ceiling of $(N/3)/|\text{target}_C|$ per category. Reading actual recovery as lift over this floor disentangles structural learning from rule-count inflation.

Extractor	Cat.	Actual	Chance	Lift
pref-doc-guided naive (N=12)	WORLD	8%	33%	−25 pp
	NARRATOR	64%	29%	+35 pp
	PLAYER	22%	44%	−22 pp
pref-doc-guided cat-aware (N=28)	WORLD	83%	78%	+5 pp
	NARRATOR	86%	67%	+19 pp
	PLAYER	67%	100%[*]	−33 pp
leak-free naive (N=12)	WORLD	17%	33%	−16 pp
	NARRATOR	71%	29%	+42 pp
	PLAYER	0%	44%	−44 pp
leak-free cat-aware (N=25)	WORLD	83%	69%	+14 pp
	NARRATOR	79%	60%	+19 pp
	PLAYER	44%	93%[*]	−49 pp

[*] Capped: $N/3$ exceeds the 9 target PLAYER rules. Numbers from `analysis/leak_diagnostics_results.json`.

Note that leak-free category-aware has the strongest WORLD lift over chance (+14 pp) of any configuration: removing preference-document access reduces total rule count, which mechanically lowers the chance floor, exposing genuine structural learning that the ceiling number obscured.

K. Leak-free extractor prompts and recovery

The leak-free condition mirrors the preference-document-guided recovery setup except the extractor sees only (y_t, y'_t) and the existing rules; `secret-preferences.md` is never shown to the extractor. Both naive and category-aware variants are run on the same 3 projects $\times$ 6 inputs.

Naive (leak-free).

```
Compare these two responses and extract a rule.
GENERATED (AI): {generated}
EDITED (Author): {edited}
EXISTING RULES: {existing_rules}
If the edit reveals a preference NOT already captured, propose ONE new rule. If
the responses are similar or the rule already exists, respond with: NO_NEW_RULE
Format: RULE: [your rule here] or NO_NEW_RULE
```

Category-aware (leak-free).

```
Compare these two responses and decide what rules the edit reveals, using the
Orchid (Wu 2025) taxonomy for IDN design:
WORLD --- world-settings & characters: which entities/objects/facts exist, plot
rails, setting constraints, canonical behavior of characters or forces.
NARRATOR --- narrator behavior: POV, prose style, tone, diction, editorializing,
atmosphere.
PLAYER --- player interactions: how player inputs are interpreted/acknowledged/redirected,
what kinds of actions are encouraged or discouraged, consequences.
GENERATED (AI): {generated}
EDITED (Author): {edited}
EXISTING RULES: {existing}
For EACH category, ask: does the diff between GENERATED and EDITED reveal a
preference rule in this category that is NOT already covered by EXISTING RULES?
If yes, propose ONE rule. If not, write NONE. Do not invent rules unsupported by
the diff.
Output EXACTLY three lines:
WORLD: <rule or NONE>
NARRATOR: <rule or NONE>
PLAYER: <rule or NONE>
```

Leak-free recovery. Per-project category counts under the two leak-free extractors. Naive recovers no PLAYER rules in any project; category-aware recovers WORLD in all three, matching the preference-document-guided ceiling on WORLD coverage.

	Naive (LF)			Cat-aware (LF)		
Project	W	N	P	W	N	P
`escaperoom`	0	4	0	1	6	0
`fireplace`	2	1	0	4	2	3
`lifesim`	0	5	0	5	3	1
Total	2	10	0	10	11	4
Recovery %	17	71	0	83	79	44

Compared with the preference-document-guided ceiling (Appendix B), leak-free naive loses PLAYER recovery entirely (22% $\rightarrow$ 0%) but slightly improves WORLD (8% $\rightarrow$ 17%) and NARRATOR (64% $\rightarrow$ 71%); leak-free category-aware

matches the ceiling on WORLD (83% = 83%), drops 7 pp on NARRATOR (86% → 79%) and 23 pp on PLAYER (67% → 44%). The structural recovery story—category-aware extraction restores world-constraint coverage that naive misses—survives the removal of preference-document access; the PLAYER gap, where that access inflated the ceiling most, does not. Numbers are from `leakfree_loop_results.json`.

L. Retrieval pilot: does context-conditional retrieval help at this rule-set size?

PRELUDE-style retrieval selects the k nearest historical preferences per generation context rather than concatenating the full rule set ((Gao et al., 2024)). Bonsai uses global concatenation, isolating the contribution of extraction. To check whether that simplification costs anything at our current scale, we ran a small two-phase pilot using the leak-free category-aware rule sets (25 rules total across the three projects: `escaperoom` 7, `fireplace` 9, `lifesim` 9) and the same nine held-out inputs as the RQ1 grid (3 inputs × 3 projects on each project's held-out scene). All three projects' rules are pooled into one pool, simulating the cross-project pollution a deployed Bonsai instance would face if one author worked on multiple projects or if rules from a shared studio leaked across projects. Embeddings: `text-embedding-3-small`; queries: `<scene>: <input>`; generator: Claude Sonnet 4.6; rater: Claude Haiku 4.5. Numbers are from `retrieval_pilot_results.json`.

Phase 1: retrieval precision. For each held-out input, retrieve top-k rules from the pooled 25 by cosine similarity. Precision@k = fraction of those k from the input's own project; chance baseline is the project's share of the pool.

	escaperoom (7/25)		**fireplace (9/25)**		**lifesim (9/25)**		
k	prec	lift	prec	lift	prec	lift	**lift (all)**
1	1.00	+0.72	1.00	+0.64	0.00	−0.36	+0.33
3	0.67	+0.39	0.67	+0.31	0.56	+0.20	+0.30
5	0.60	+0.32	0.40	+0.04	0.47	+0.11	+0.16
10	0.40	+0.12	0.33	−0.03	0.53	+0.17	+0.09

Aggregate precision lifts above chance for $k \leq 5$, with the strongest signal at small k. Per project, `lifesim`'s top-1 is 0/3 (every `lifesim` query's nearest neighbor in the pool is a non-`lifesim` rule), recovering by $k = 3$. Project membership is therefore a recoverable signal in the embedding space, but heterogeneously so.

Phase 2: comparative generation. Phase 1's precision-lift cleared a pre-registered +0.10 gate, so we ran 9 inputs × 4 conditions: *none* (no rules), *all* (all 25 polluted rules—Bonsai's current setting under cross-project pollution), *topk* (k=3 retrieved, the Phase 1 sweet spot), *own* (the input's own-project rules only, the retrieval ceiling). Each generation rated PREFERENCE (1–5) against that project's `secret-preferences.md`.

Project	*none*	*all*	*topk*	*own*
`escaperoom`	2.33	4.67	3.67	5.00
`fireplace`	1.33	2.67	**4.00**	5.00
`lifesim`	3.33	4.67	3.00	4.67
Mean	2.33	4.00	3.56	4.89

Four deltas summarize the result. *topk vs. none*: +1.22 (retrieval still extracts useful signal). *topk vs. all*: −0.44 (at k=3, naive retrieval underperforms polluted global concat). *own vs. all*: +0.89 (perfect retrieval would help, but only modestly at this scale). *own vs. topk*: +1.33 (the gap from naive retrieval to the ceiling is large: most of the available retrieval value is left on the table by querying on `<scene>: <input>` alone).

Why precision was a misleading proxy. Phase 1 reported 63% precision at k=3 (lift +0.30); Phase 2 shows that three correctly-targeted rules do not out-condition twenty-five mostly-noisy ones at our current scale. Even cross-project rules push generation toward concrete, edited prose in aggregate; the in-project signal at k=3 is dominated by the bulk-conditioning effect of the full pool. Project-membership precision is the wrong success metric at small rule-set sizes; the right one would be a downstream quality metric, which is what Phase 2 measures.

Per-project asymmetry: voice-bleed as the failure mode. `fireplace` is the only project where retrieval beats global concat ($+1.33$): its distinctive loud-absurdist voice was uniquely *hurt* by cross-project pollution (*all* mean 2.67, the lowest of any *all* cell), so pruning to top-k helps. `lifesim` is the inverse (-1.67): its contemporary, understated voice embeds close to the other projects' rules, so retrieval picks up wrong-project rules and applies them. The smoking-gun trial is `lifesim`, "drop out of school," *topk* condition:

> *Generated:* "Your bike is gone—sold off to cover last month's rent while you were still deciding. The afternoon smells like cut grass and exhaust, and somewhere down the block a screen door bangs shut."

The bike is from `fireplace`; the prose register (sensory, atmospheric, late-paragraph closing image) is `fireplace`'s, not `lifesim`'s. The retrieved rules pulled the generation across project lines into the wrong author's voice. Two other `lifesim` *topk* trials show the same pattern ("start a band" in `fireplace`'s wry meta-narration, "get into trouble" in `escaperoom`'s action register).

What this means. At 25 rules, the bottleneck is extraction quality, not conditioning topology: global concatenation is empirically competitive, naive embedding retrieval over rule text underperforms it, and the ceiling for any retrieval strategy querying on `<scene>: <input>` alone is $+0.89$ PREFERENCE. Retrieval becomes load-bearing once rule sets grow large enough that prompt budget rather than rule signal is the binding constraint. The per-project asymmetry, particularly the `lifesim` voice-bleed, identifies query enrichment (scene context, edit history, per-edit confidence, Orchid-category routing) as the concrete next step rather than off-the-shelf embedding retrieval. Pilot scope: $n{=}9$ inputs, 1 trial per condition; treat the absolute deltas as directional rather than precise.