

Generating Quality Word-Association Puzzles

Ashish Khadka
Tagliatela College of Engineering
University of New Haven
West Haven, CT, United States
akhad4@unh.newhaven.edu

Mohamad Nassar
0000-0001-8857-4436
Tagliatela College of Engineering
University of New Haven
West Haven, CT, United States
mnassar@newhaven.edu

Muhammad Aminul Islam
0000-0002-6221-3090
Tagliatela College of Engineering
University of New Haven
West Haven, CT, United States
mislam@newhaven.edu

Sanaa Kaddoura
0000-0002-4384-4364
Computing and Applied Technology
Zayed University
Abu Dhabi, United Arab Emirates
sanaa.kaddoura@zu.ac.ae

Abstract—Word-association puzzles require players to identify a hidden relationship or a connection between a group of keywords. Generating such puzzles is possible with proprietary/frontier LLM models, probably due to their reasoning capacities, but we do not have many insights on how quality puzzles could be generated with limited resources. Open-source instruction-tuned language models show limited capability as they only generate poor quality and repetitive puzzles. In this paper, we explore how an open-source language model (i.e., Llama-3.1-8B-Instruct, Mistral-7B-Instruct-v0.3, and Qwen2.5-7B-Instruct) could be leveraged to generate high-quality and diverse puzzles. We demonstrate a method based on Supervised Fine-Tuning (SFT) from a teacher model (i.e., Claude) for generating quality word association puzzles. We validate the puzzles and assess their diversity using OpenAI’s GPT-5.2 Thinking as a judge. Our findings show a low-cost and efficient method for distilling a lateral thinking skill from a large reasoning model.

Index Terms—LLM Reasoning, Lateral Thinking, Synthetic Data, Puzzle Generation, Supervised Fine-Tuning (SFT)

I. INTRODUCTION

LARGE Language Models (LLMs) have become remarkably capable at tasks that require lateral thinking [1] and the emergence of ‘thinking’ models has pushed performance on standardized reasoning benchmarks to near-saturation levels [2]. Lateral thinking refers to the ability to form creative or unexpected connections between things that may not be obvious at first glance [3]. Word-association puzzles such as LinkedIn Pinpoint [4] provide a natural and challenging testbed for evaluating this capability in LLMs. These puzzles require reasoning in breadth rather than in depth to solve them. A player must move beyond obvious semantic associations, and instead identify the underlying logic connecting words that may seem unrelated.

In this paper, we study the popular word game Pinpoint in which players must identify a hidden category connecting five words that are revealed one at a time as clues. The clues are deliberately designed to appear unrelated, making them difficult to solve using linear, deductive reasoning. In this work, we present a method for generating and evaluating Pinpoint

puzzles using light-weight LLMs. We then apply *Supervised Fine-Tuning* (SFT) to improve puzzle quality and validate our approach using a strong LLM-as-a-judge framework that compares *fine-tuned* (FT) outputs against base models, other open models, and a state-of-the-art model like Anthropic’s Claude Opus 4.6 [5]. We use a teacher-student pipeline that is based on a well-established tradition of knowledge distillation [6], in which a smaller student model is trained to replicate the output distribution of a larger teacher.

Our work makes the following contributions. First, we demonstrate that open-source light-weight *language models* (LMs) can serve as competent puzzle generators. This provides a new low-cost source of Pinpoint, which, at the time of writing, are limited to only 704 puzzles. Second, we publicly release a benchmark dataset of 9,824 Pinpoint-style puzzles¹, constructed from Wikipedia-derived seeds, for evaluating lateral thinking in LMs. Finally, we show that SFT on synthetic examples generated by Claude substantially improves the quality of puzzles produced by these smaller models, offering a scalable source for high-quality puzzles without relying on manual creation.

We deliberately focus on lightweight open-source models for three reasons. First, they enable reproducibility: any researcher with access to a single consumer GPU can replicate and extend our results without using proprietary APIs. Second, testing at this scale reveals failure modes like the clue-ordering failure in Mistral-Base model shown in Figure 6, which are handled correctly by frontier models. Third, our teacher-student pipeline requires the frontier model only for one-time data generation; once fine-tuned, the student models operate independently, making ongoing deployment practical without recurring API costs.

Ice	Ice
CLUE 2	Jet
CLUE 3	Booster
CLUE 4	Six
CLUE 5	Back

(a) A Pinpoint game at launch with first clue revealed.

(b) All clues and common category revealed after a correct guess after the third clue.

Fig. 1: Examples of a LinkedIn Pinpoint game. All 5 clues belong to a common category. Player needs to guess the category in as few clue reveals as possible.

II. BACKGROUND

A. Pinpoint Puzzle

Pinpoint is a daily word-association puzzle launched by LinkedIn on May 2024. A player begins with an initial clue or word (Fig. 1a), and additional clues are revealed one at a time after each guess. The clues follow a structured difficulty progression from more obscure to more obvious. As shown in Fig. 1b, initial clues such as “ice,” “jet,” and “booster” may suggest a misleading or partial association (e.g., “water”). However, as additional clues like “six” and “back” are revealed, the intended category (“pack”) becomes increasingly apparent. The player has up to five guesses to identify the common category and the game ends once the category is found or all guesses are exhausted. Player’s guess need not exactly match the correct category. In practice, we observe that morphologically similar variants or partial matches are often accepted as correct. To account for this during solving puzzles using LLMs, we use cosine-similarity based heuristic introduced in our prior work [1], where a guess is considered correct if it exceeds 75% similarity threshold with the correct answer. Pinpoint puzzle can be played on LinkedIn ².

B. Procedural Content Generation (PCG)

PCG refers to the creation of games using algorithms, including LLMs. It enables content generation at a linguistic and semantic level that was previously inaccessible to traditional algorithms and Machine Learning methods [7]. We frame Pinpoint puzzle generation as a PCG problem, using LLMs as the generator grounded in Wikipedia seeds to ensure diversity. We combine this with SFT to improve the quality of the generator, producing a pipeline that addresses both the diversity and quality of procedural puzzle generation.

III. RELATED WORK

The closest work to ours is [8], who investigate the use of GPT models to generate *New York Times* (NYT) Connections puzzles, a word-association game structurally similar

but strategically opposite to Pinpoint. In NYT Connections, players must categorize 16 words into four groups of four words based on a hidden theme. They frame puzzle generation as a PCG problem and propose a Tree-of-Thoughts prompting approach, validating quality through a human user study. Our work differs in three key ways: we focus on Pinpoint, we use SFT rather than prompting alone to improve generation quality, and we introduce a large-scale synthetic benchmark rather than relying on human evaluation alone. Another notable example of using LLM in PCG for games include MarioGPT [9], which fine-tunes GPT-2 to generate new levels in Super Mario Bros from textual prompts.

We also build on our own prior work [1] on the solving side, where we benchmark open-source models like LLaMa-3.1-8B, Qwen2.5-Coder-32B-Instruct, Mistral-7B-Instruct-v0.2 and DeepSeek-R1-Distill-Qwen-14B to solve Pinpoint and Connections. The results show that Pinpoint is comparatively easier than Connections, and that strong performance of solvers on Connections is highly predictive of strong performance on Pinpoint.

Our work draws on [6], using knowledge distillation with synthetic training examples from Claude Opus 4.6 to fine-tune student models. According to [10], black-box distillation via SFT has become the dominant technique to cheaply improve capabilities in open-source models. However, [11] places a fundamental ceiling on student performance relative to the teacher. We apply knowledge distillation for capability transfer in lateral-thinking puzzle generation, a task that requires “System 2 thinking” [12]. This type of thinking requires a deliberate, effortful reasoning rather than quick pattern-matching style of thinking.

Evaluating open-ended generation tasks remains a challenge in the field. [13] establish the LLM-as-judge paradigm, showing that strong LLM judges such as GPT-4 achieve over 80% agreement with human preferences making it comparable to agreement between human judges. This makes automated pairwise evaluation a scalable and cost-effective alternative to human annotation. We adopt GPT-5.2 Thinking as an independent judge drawn from a different model family than our teacher (OpenAI vs. Anthropic) to avoid self-preference bias. Recent work by [14] has further investigated the reliability of LLM-as-judge under varying conditions, including the effect of temperature on judge consistency. We used the GPT-5.2 Thinking with default generation parameters for ranking the puzzles.

IV. METHODOLOGY

Our overall framework is depicted in Figure 2. We present the details of our framework and methodology in this section.

A. Identifying Constraints for a High-Quality Puzzle

By analyzing existing puzzles and the in-game tutorial, we identify some basic structural constraints:

- each puzzle must contain exactly five clues;

¹<https://github.com/khadkaashish/pinpoint>

²<https://www.linkedin.com/games/pinpoint>

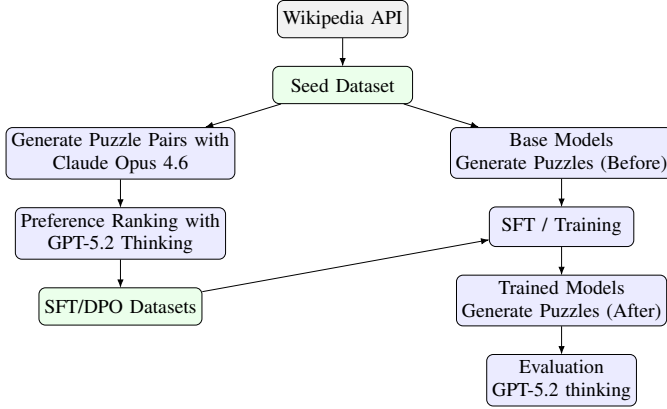


Fig. 2: Our framework for improving the generation of quality puzzles

- each clue must be limited to 1–3 words; and
- duplicate or near-duplicate clues are disallowed.

These constraints are not used for evaluation purposes, but are enforced through filtering steps in our puzzle generation pipeline. However, generating a high-quality Pinpoint puzzle, such that the puzzle is valid, engaging and enjoyable is a harder problem than solving it, as it requires reverse-engineering the reasoning process of an AI or human solver. It requires satisfying a set of additional rules that go beyond simple semantic and structural correctness:

No Leakage: None of the clues should contain the category or any close morphological variant. For example, if the correct category is “Painters”, clues such as “paint” or “paintings” would be considered invalid.

Coherence: Every clue must be a valid and unambiguous member of the category.

Playability: The category and clues must draw on generally accessible knowledge, avoiding overly niche, domain-specific, or culturally narrow references that would make the puzzle unsolvable for a general audience. For instance, categories such as “18th-century French poets” might be considered overly challenging.

Clue Ordering: Clues should progress from more obscure or misleading associations to more obvious ones. This keeps the game engaging and ensures a satisfying experience where the solver’s confidence increases as more clues are revealed.

These quality criteria were derived through heuristic analysis of more than 100 official LinkedIn Pinpoint puzzles. We systematically examined recurring structural and semantic patterns across puzzles to identify what distinguishes valid, engaging puzzles from poor ones. We observed that no puzzle contained clues that shared words with the correct answer i.e. no-leakage. Coherence is the most fundamental requirement as the entire premise of the game rests on all clues genuinely belonging to a single hidden category. For clue ordering, we observed that the final clue was consistently the most obvious association with the category, while earlier clues were more ambiguous or misleading. This confirmed that clues are

deliberately sequenced to make it easier to solve as they are revealed. Together, these four constraints passed in the prompts are the basis for both our synthetic data generation pipeline and our LLM-as-judge evaluation framework.

B. LLMs

We organize the LMs used in this work into three distinct roles: student models that are fine tuned for puzzle generation, a teacher model that produces high-quality synthetic training data, and a judge model that evaluates puzzle quality.

a) Student Models: We fine-tune three open-source models, all in the 7–8 billion parameter range: LLaMA 3.1-8B-Instruct [15], Mistral-7B-Instruct-v0.3 [16], and Qwen2.5-7B-Instruct [17]. These models were selected for three reasons. First, they span diverse training lineages by Meta, Mistral AI, and Alibaba. Second, all three are freely accessible on HuggingFace³ without access restrictions, ensuring reproducibility. Third, as models in the 7–8 billion parameter range, they are lightweight enough to fine-tune and produce results under the compute and budget constraints of our work.

b) Teacher Model: We use Anthropic’s Claude Opus 4.6 [5] as the teacher model responsible for generating the synthetic puzzle dataset used in SFT training. Claude Opus 4.6 was selected for its strong instruction-following ability and broad semantic knowledge. The teacher is used only for data generation and is not required at inference, keeping deployment costs low.

c) Judge Model: Puzzle quality is evaluated using Open AI’s —GPT-5.2 Thinking— [18] as an automated judge. We selected GPT-5.2 Thinking specifically for its extended reasoning capabilities, which are well-suited to the multi-criteria judgments required for our evaluation. Crucially, the judge is drawn from a different model family than the teacher, mitigating the self-preference bias that would arise from using the same model family for both generation and evaluation.

C. Wikipedia Seed Extraction and Dataset generation

We found that LMs quickly converge to a narrow set of repetitive, low-quality puzzle categories without external grounding or seeding. Models default to generic categories such as “Types of Colors” or “Words that start with A” or “Words that start with B”. This suggests that models tend to default to examples that are overrepresented in their training data. To mitigate this, we extract category seeds from the Wikipedia category graph and its associated entity relations [19], providing a diverse and well-curated semantic source for puzzle generation. A *seed* is a short category or theme extracted from Wikipedia that provides a semantic direction for generation, prevents duplicates and increases diversity of puzzles. We extract a total of 4,912 unique seeds. The number is not even because we generated more than that but then filtered out the ones not satisfying the four validity constraints aforementioned.

³<https://huggingface.co/models>

Some of the extracted seeds were too niche or too broad. To address this, we instructed the generation prompt to allow the model to reword the seed category, although the reworded category remained semantically related to the original seed. In practice, this produced meaningful improvements in playability. For instance, niche seed like “21st-century musicologists” was reworded to a more general “Music Genres”, yielding clues such as “Electronic”, “French style”, “Soul revival”, “Funky pop”, “Classic rock”, etc. which are associations that naturally guide a solver toward guessing words related with “Music”. Similarly, a general seed like “Alcohol Measurement” was reworded to “Bartending”, producing clues such as “Pouring speed”, “Shaking hard”, “Jigger full”, “Drink size”, and “Shot glass” which is a more enjoyable and playable puzzle for a general audience (not necessarily an alcoholic one!).

For each extracted seed, we generate two non-identical candidate puzzles using the generation prompt described in Appendix. The prompt template had only two fields that varied between API calls: the input *seed*, and the number of puzzles requested to be generated, *n_puzzles*. We experimented with larger values of *n_puzzles*, but empirical inspection showed that requesting more than two puzzles per seed often reduced quality and diversity, with later outputs becoming only minor variations of earlier ones. Hence, we set *n_puzzles*=2 per seed in a single Claude API call. Having two puzzles per seed is also beneficial to build a preference dataset for Direct Preference Optimization (DPO) training in future work. We constructed a dataset comprising $4912 \times 2 = 9,824$ Claude Opus 4.6-generated examples. Each puzzle was generated as a (seed, puzzle) pair, where the puzzle contains the group of ordered clues and the target category. In most cases, the target category matched the seed directly. However, Claude sometimes rephrased or generalized the category when a seed was overly niche or overly broad for a playable Pinpoint category. For SFT training, each example was reformatted as a (prompt, puzzle) pair, where the prompt is the same generation prompt used earlier and the puzzle is the Claude-generated puzzle. This dataset is used for fine tuning the three student models: LLaMA-3.1-8B-Instruct, Mistral-7B-Instruct-v0.3, and Qwen2.5-7B-Instruct. All outputs were validated for matching the four validity constraints by a human operator.

D. SFT

SFT directly trains the model to imitate high-quality examples, providing a stable and reliable alignment signal. We use the Transformers Reinforcement Learning (TRL) framework [20], with lightweight Parameter-Efficient Fine-Tuning adapters via *Low-Rank Adaptation* (LoRA) [21] applied on top of frozen base model weights. This plug-and-play approach keeps the base model parameters fixed, while the learned adapters capture task-specific behavior, including consistently producing the specified JSON format. LoRA was configured

with rank $r = 16$ and $\alpha = 32$, and applied to all attention and feed-forward projection layers of the models. All three models were trained for 2 epochs with a learning rate of 2×10^{-5} , a batch size of 1, and 16 gradient accumulation steps, yielding an effective batch size of 16 per update step. Training was performed on a single GPU (one device).

E. Puzzle evaluation

We evaluate generalization on 100 fresh seeds extracted separately from Wikipedia again, completely different from the training dataset. For each seed/model, one puzzle is generated before fine-tuning and one after fine-tuning, serving as the basis for evaluating whether SFT successfully improves puzzle quality in average. The judge model selects a winner from each pair using a forced-choice ranking prompt (shown in Appendix) with no ties permitted (“chosen” or “rejected”). To ensure a robust evaluation, we compare our FT models against three distinct categories:

- 1) **Student (Base) Model vs. FT model:** We compared each FT model against its own base version (e.g., Base Llama 3.1-8B-Instruct vs. Llama-FT)
- 2) **Cross-Model Comparison:** We conducted comparison between different FT models (Llama-FT vs. Mistral-FT vs. Qwen-FT) to identify which student model is most receptive to lateral-thinking alignment.
- 3) **Student vs Teacher:** Finally, we compared each FT model against Claude Opus 4.6 puzzles. This allows us to measure the performance gap between our lightweight models and a state-of-the-art frontier model.

V. EXPERIMENTAL RESULTS

A. Student (Base) Model vs. FT model

Figure 3 shows the preference scores of each FT model against its corresponding base version, as judged by GPT-5.2 Thinking. All three FT models were preferred by a substantial margin, demonstrating that SFT consistently improves puzzle quality across all model families. LLaMA-3.1-8B-Instruct showed the largest relative improvement, with the FT version preferred in 73% of trials compared to 27% for the base model. Mistral-FT followed closely with a 72% preference rate, while Qwen-FT achieved 65%. The consistency of these improvements across three architecturally distinct model families suggests that the gains are attributable to SFT alignment rather than any model-specific characteristic.

Before fine-tuning, the models frequently produced clues that were either too literal, lacking lateral breadth, too niche for a general audience to solve, or had answer leakage. Base models also struggled to calibrate clue difficulty, producing clues that were either too vague or too obvious, undermining the progressive reveal central to the Pinpoint format. Fine-tuning successfully addressed these failure modes, enabling the models to produce puzzles that more reliably satisfy the quality constraints defined in Section IV-A.

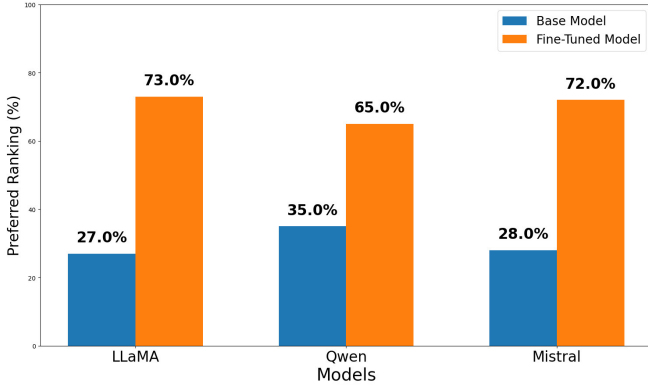


Fig. 3: Preference comparison for student (base) vs. their FT versions. Judged by GPT-5.2 Thinking model.

B. Cross-model Comparison

Figure 4 shows pairwise comparisons between FT models to identify which architecture is most suited to lateral-thinking puzzle generation. Qwen-FT emerged as the strongest overall performer, defeating LLaMA-FT in 66% of trials and Mistral-FT in 55% of trials. Mistral-FT ranked second, outperforming LLaMA-FT in 61% of direct comparisons. Notably, while LLaMA-3.1-8B-Instruct showed the greatest relative improvement over its base version, it ranked last in cross-model comparisons. This dissociation suggests that a model may respond strongly to fine-tuning while still falling short of competitors with a stronger baseline understanding of word and concept associations. The superior cross-model performance of Qwen-FT suggests that the Qwen2.5-7B-Instruct architecture may possess richer internal representations of the broad semantic and cultural knowledge required for high-quality puzzle generation.

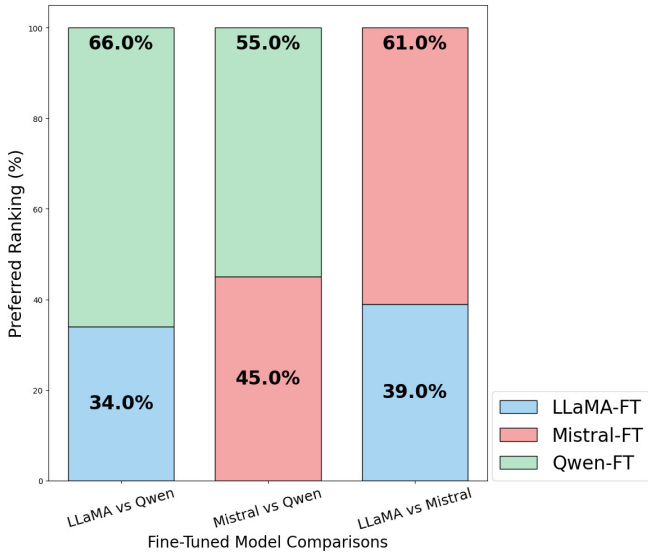


Fig. 4: Preference comparison between FT student models. Judged by GPT-5.2 Thinking model.

C. Student vs. Teacher

Figure 5 shows comparison of each base and fine-tuned model against Claude Opus 4.6, providing an upper-bound reference for puzzle quality. As expected, Claude Opus 4.6 substantially outperformed all student models in both base and fine-tuned configurations, confirming that a significant capability gap remains between lightweight open-source models and state-of-the-art frontier models on this task. However, fine-tuning produced meaningful gains even against this strong baseline. The base versions of Qwen and Mistral were effectively never preferred over Claude at only 2% of trials. After fine-tuning, both Qwen-FT and Mistral-FT reached a 16% preference rate against Claude, representing an eightfold increase. LLaMA also improved from 3% to 7%, though it remained the weakest performer against the teacher model.

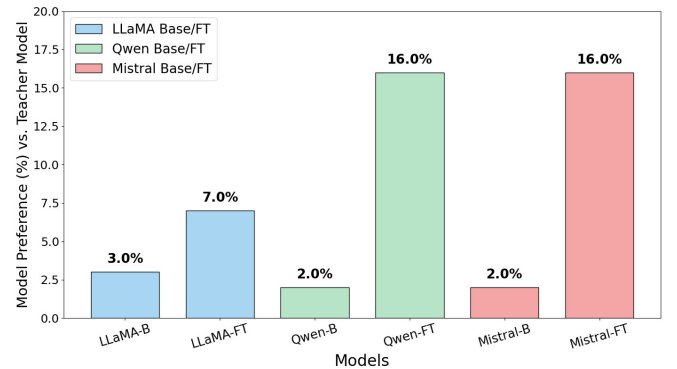


Fig. 5: Preference comparison for Base and FT student model vs. teacher model. Judged by GPT-5.2 Thinking model.

These results carry an important practical implication: while fine-tuned lightweight models do not match teacher-level quality, targeted SFT on synthetic teacher-generated data can substantially narrow the capability gap. The eightfold improvement in preference rate for Qwen-FT and Mistral-FT against Claude Opus 4.6 demonstrates that smaller models can be meaningfully specialized for creative puzzle generation tasks without access to massive computing power used by big AI companies.

D. Clue Ordering via Category–Clue Cosine Similarity

We also evaluate whether models follow the clue ordering constraint before and after SFT. We compute the average cosine similarity between each generated clue and its ground-truth category, grouped by clue position (clue1 through clue5). We use the sentence transformers library and the sentence-embedding model Lihuchen/pearl_base [22]. Figures 6 and 7 report these trajectories for the base and fine-tuned model variants respectively, alongside Claude Opus 4.6 as the teacher reference.

As shown in Figure 6, Claude Opus 4.6 shows a consistent upward trend across all five positions, confirming that its clues reliably progress from obscure to obvious as intended. Among the base models, only Llama-Base follows this trend. Mistral-Base begins at its highest similarity score at clue1 and declines,

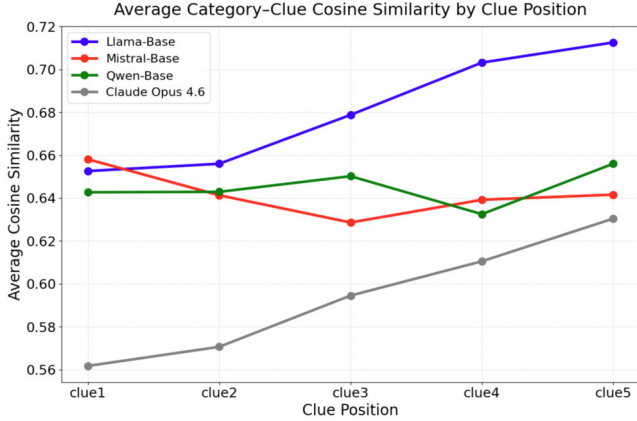


Fig. 6: Average cosine-similarity score between category and each clue position for student (base) models and teacher model.

indicating a reversed ordering in which the first clue is the most revealing. Qwen-Base follows the desired trend through clue3 but deteriorates in subsequent positions.

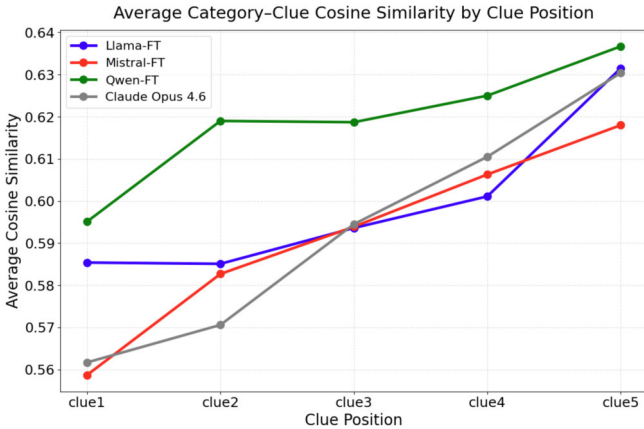


Fig. 7: Average cosine-similarity score between category and each clue position for FT models and teacher model.

Figure 7 demonstrates that fine-tuning substantially corrects these failure modes. All three fine-tuned models now show steadily increasing similarity scores across clue positions, closely matching the teacher’s pattern. Qwen-FT shows the strongest and most consistent improvement, ending with the highest similarity at clue5 among all FT models. Mistral-FT and Llama-FT also recover from their base-model failures, catching up to the teacher trajectory by clue3 and continuing to rise through clue5.

VI. DISCUSSION

A. Seed Quality and Category Rewording

Despite the rewording strategy described earlier, seed quality remained a recurring challenge. Seeds derived from highly specific Wikipedia categories such as “Government ministers

of the Bahamas”, frequently produced niche puzzles that were too boring and difficult to solve. At the other extreme, overly broad seeds such as “Sport-related lists” led to generic puzzles that lacked the lateral creativity central to the Pinpoint format. We were unable to fully resolve this limitation for every seed, as the model’s ability to reword a seed into a unique puzzle is itself bounded by its knowledge and creativity.

B. Limitations of SFT

We also acknowledge the inherent limitations of SFT, although it produced consistent and measurable improvements across all three student models. In SFT, a student learns to replicate the teacher’s output distribution, which places a fundamental ceiling on student performance [11], hence even the strongest fine-tuned model (Qwen-FT) was preferred over Claude Opus 4.6 in only 16% of trials. Training exclusively on synthetic data also risks narrowing the diversity of model outputs over successive generations [23]. One of the more interesting findings in our results is the dissociation between relative improvement and absolute performance across model families. LLaMA-3.1-8B-Instruct showed the greatest improvement from fine-tuning, yet ranked last in cross-model comparisons. For puzzle generation specifically, a model with richer prior knowledge of word associations and cultural references may have a structural advantage that fine-tuning alone cannot overcome.

C. Human evaluators vs. LLM-as-Judge

All experiments were conducted on a Dell server equipped with a single NVIDIA RTX 8000 GPU with 44 GB VRAM. These infrastructure constraints shaped several design decisions in this work. Fine-tuning was restricted to models in the 7–8 billion parameter range, as larger models exceeded available memory. Although human evaluation of a puzzle’s enjoyability and playability would be the most reliable and valid measure, it was not feasible at the scale required, given the cost and time involved in recruiting and compensating annotators. Therefore, our evaluation framework relies entirely on GPT-5.2 Thinking, which may exhibit its own biases in what constitutes a “good” puzzle. In particular, the Judge model evaluates puzzles along the criteria defined in our rubric, but cannot assess the subjective experience of solving a puzzle.

VII. CONCLUSION AND FUTURE WORK

In this work, we introduced a scalable pipeline for Pinpoint puzzle generation and for improving puzzle quality in smaller models without requiring expensive infrastructure used by big AI companies. Using Claude Opus 4.6 as a teacher model, we constructed a synthetic dataset of 9,824 high-quality puzzles and used it to fine-tune three lightweight open-source models via SFT. Evaluation by GPT-5.2 Thinking as an automated judge demonstrated consistent and substantial improvements across all three student models, with preference rates of 73%, 72%, and 65% for LLaMA-FT, Mistral-FT, and Qwen-FT respectively over their base versions.

LinkedIn Pinpoint is a deceptively challenging task that requires lateral thinking rather than linear, deductive reasoning. So, generating enjoyable and engaging puzzles using LMs requires an ability to reason about how other players will reason. We incorporate strict puzzle constraints (no leakage, coherence, playability, clue ordering, and other structural rules) in our prompts to guide puzzle generation and ranking.

However, several directions remain open for future studies. First, seed quality represents a bottleneck in our pipeline and future work should explore more systematic approaches to seed selection to reduce puzzles that are too niche or too generic. Second, our evaluation relied entirely on automated judgment. Human evaluation captures the subjective experience of solving a puzzle, including how different humans guess slower or faster based on their different knowledge and backgrounds. Our evaluation did not control for judge temperature, which has been shown to affect consistency in LLM-as-judge settings [14]. Future work should investigate the sensitivity of puzzle quality rankings to this parameter. Third, while SFT produced consistent improvements, it cannot directly reflect human preferences. A natural next step is to apply Reinforcement Learning from Human Feedback (RLHF) to directly optimize for puzzle quality as judged by human players.

Overall, our results demonstrate that LMs are capable puzzle generators, and that SFT on teacher-generated synthetic data is an effective and scalable strategy for improving puzzle quality in smaller models. We hope future work will build on our work through better seed curation, human evaluation, and RLHF.

APPENDIX

Puzzle Generation Prompt

You are a puzzle constructor for LinkedIn Pinpoint.
 INPUT SEED (may be too broad or too niche): "{seed}"
 TASK: Generate EXACTLY {n_puzzles} puzzles. Each puzzle must be a JSON object with:
 • category: string
 • clues: string[5]
 Rules:
 • Exactly 5 clues per puzzle.
 • Each clue must be 1--3 words (short tokens). NO full sentences. NO quotations. NO verse lines.
 • Clues must belong to the category.
 • Order clues from MOST OBSCURE (clue 1) to MOST OBVIOUS (clue 5), like Pinpoint.
 • NO LEAKAGE: none of the important words from the category may appear in any clue.
 • No duplicate or near-duplicate clues within a puzzle.
 • If the seed is too niche, make it general that people can have fun playing. You may reword the seed or choose something related to the seed as

the new category.

- Across puzzles, ensure they are not identical (at least one clue differs).
- Output format: Return ONLY valid minified JSON, exactly one JSON array of {n_puzzles} objects. No markdown. No prose. No extra keys.

Ranking Prompt

You are a strict puzzle-quality judge for LinkedIn Pinpoint-style puzzles. You will be given:
 • Seed (optional; may be empty)
 • Puzzle A: category + 5 clues
 • Puzzle B: category + 5 clues
 Goal: Decide which puzzle is BETTER.
 Rubric (highest priority first):
 1) No leakage: clues should NOT contain the category words (or close morphological variants).
 2) Correctness & coherence: Clues must belong to the category.
 3) Playability: avoid overly niche trivia; prefer generally-known knowledge. Puzzle that is fun to play.
 4) Ordering: clues should progress from more obscure to more obvious. You MUST pick a winner (no ties). If close, pick the one with better non-leakage and that is fun to play for a player.
 Output MUST be exactly one character: A or B. No other text.

GENERATIVE AI (GENAI) DISCLOSURE STATEMENT

During the preparation of this work the author(s) used Open AI's GPT-5.2 Thinking, Anthropic's Claude Sonnet 4.6 for editing and revising text, removing grammatical errors, and for help with generating code snippets used in experiments including SFT and evaluation. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

FUNDING

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

REFERENCES

- [1] A. Khadka, M. Nassar, and S. Khare, "Are language models good at lateral thinking?" in *2025 3rd International Conference on Foundation and Large Language Models (FLLM)*. IEEE, 2025, pp. 565–572.
- [2] I. Arcuschin, J. Janiak, R. Krzyzanowski, S. Rajamanoharan, N. Nanda, and A. Conmy, "Chain-of-thought reasoning in the wild is not always faithful," *arXiv preprint arXiv:2503.08679*, 2025.
- [3] E. De Bono, *Lateral Thinking: Creativity Step by Step*, ser. Harper colophon books. Harper & Row, 1970.

- [Online]. Available: <https://books.google.com/books?id=H-ROAAAAMAAJ>
- [4] LinkedIn Games, “Pinpoint,” 2024, <https://www.linkedin.com/games/pinpoint/>. Accessed: 2025-01-21.
 - [5] Anthropic, “Claude opus 4.6,” 2025, <https://www.anthropic.com/news/claude-opus-4-6>. Accessed: 2026-03-27.
 - [6] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” *arXiv preprint arXiv:1503.02531*, 2015.
 - [7] M. F. Maleki and R. Zhao, “Procedural content generation in games: A survey with insights on emerging llm integration,” in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 20, no. 1, 2024, pp. 167–178.
 - [8] T. Merino, S. Earle, R. Sudhakaran, S. Sudhakaran, and J. Togelius, “Making new connections: Llms as puzzle generators for the new york times’ connections word game,” in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 20, 2024, pp. 87–96.
 - [9] S. Sudhakaran, M. González-Duque, M. Freiberger, C. Glanois, E. Najarro, and S. Risi, “Mariogpt: Open-ended text2level generation through large language models,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 54 213–54 227, 2023.
 - [10] X. Xu, M. Li, C. Tao, T. Shen, R. Cheng, J. Li, C. Xu, D. Tao, and T. Zhou, “A survey on knowledge distillation of large language models,” *arXiv preprint arXiv:2402.13116*, 2024.
 - [11] A. Gudibande, E. Wallace, C. Snell, X. Geng, H. Liu, P. Abbeel, S. Levine, and D. Song, “The false promise of imitating proprietary llms,” *arXiv preprint arXiv:2305.15717*, 2023.
 - [12] T. Hagendorff, S. Fabi, and M. Kosinski, “Human-like intuitive behavior and reasoning biases emerged in large language models but disappeared in chatgpt,” *Nature Computational Science*, vol. 3, no. 10, pp. 833–838, 2023.
 - [13] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing *et al.*, “Judging llm-as-a-judge with mt-bench and chatbot arena,” *Advances in neural information processing systems*, vol. 36, pp. 46 595–46 623, 2023.
 - [14] L. Li, L. Sleem, Y. Xu, Y. Song, A. Jia, J. Francois, and R. State, “The necessity of setting temperature in llm-as-a-judge,” *arXiv preprint arXiv:2603.28304*, 2026.
 - [15] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan *et al.*, “The llama 3 herd of models,” *arXiv preprint arXiv:2407.21783*, 2024.
 - [16] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. Le Scao, T. Lavril, T. Wang, T. Lacroix, and W. El Sayed, “Mistral 7b,” *arXiv preprint arXiv:2310.06825*, 2023.
 - [17] B. Hui, J. Yang, Z. Cui, J. Yang, D. Liu, L. Zhang, T. Liu, J. Zhang, B. Yu, K. Lu *et al.*, “Qwen2.5-coder technical report,” *arXiv preprint arXiv:2409.12186*, 2024.
 - [18] A. Singh, A. Fry, A. Perelman, A. Tart, A. Ganesh, A. El-Kishky, A. McLaughlin, A. Low, A. Ostrow, A. Ananthram *et al.*, “Openai gpt-5 system card,” *arXiv preprint arXiv:2601.03267*, 2025.
 - [19] N. Aspert, V. Miz, B. Ricaud, and P. Vanderghyest, “A graph-structured dataset for wikipedia research,” in *Companion Proceedings of The 2019 World Wide Web Conference*, 2019, pp. 1188–1193.
 - [20] L. von Werra, Y. Belkada, L. Tunstall, E. Beeching, T. Thrush, N. Lambert, S. Huang, K. Rasul, and Q. Galouédec, “TRL: Transformers Reinforcement Learning,” 2020, <https://github.com/huggingface/trl>.
 - [21] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, “LoRA: Low-rank adaptation of large language models,” in *International Conference on Learning Representations*, 2022. [Online]. Available: <https://openreview.net/forum?id=nZeVKeeFYf9>
 - [22] L. Chen, G. Varoquaux, and F. M. Suchanek, “Learning high-quality and general-purpose phrase representations,” *arXiv preprint arXiv:2401.10407*, 2024.
 - [23] I. Shumailov, Z. Shumaylov, Y. Zhao, N. Papernot, R. Anderson, and Y. Gal, “Ai models collapse when trained on recursively generated data,” *Nature*, vol. 631, no. 8022, pp. 755–759, 2024.