Within-Model vs Between-Prompt Variability in Large Language Models for Creative Tasks

Jennifer Haase[1,2] **Jana Gonnermann-Müller**[2,3] **Paul H. P. Hanel**[4]
Nicolas Leins[3,5] **Thomas Kosch**[1,2] **Jan Mendling**[1,2] **Sebastian Pokutta**[3,5]

[1]Humboldt-Universität zu Berlin [2]Weizenbaum Institute for the Networked Society
[3]Zuse Institute Berlin [4]University of Essex [5]TU Berlin

Abstract

When a large language model (LLM) generates a creative response, how much of the outcome is determined by the prompt, the model, the task, or pure sampling randomness? We address this question using a variance decomposition across 89,806 generations from 10 LLMs, 6 prompt strategies, and 30 Alternative Uses Task items, one of the most widely used creativity tests. For output quality (originality), the results invert common assumptions: the AUT item being evaluated and the prompt strategy together account for 73.0% of variance, while model choice (between-model differences) accounts for 7.6% and within-LLM stochasticity accounts for 16.6%, more than twice the model contribution. Controlling for the AUT item sharpens the picture further: prompt strategy then accounts for 60.3% of within-item variance versus 17.3% for model choice. For output quantity (fluency), the pattern reverses: model choice accounts for 55.1% of variance versus 29.8% for prompt strategy. We further show that prompts shape output *distributions*, not point estimates: "discriminative" prompts (e.g., Persona) broaden the quality distribution and yield higher peaks in capable models, while "constraining" prompts (e.g., Format) narrow it. Together, these results argue that single-sample LLM evaluations are unreliable for open-ended tasks and that evaluation designs must account for the full variance structure of generative systems.

1 Introduction

Prompt engineering assumes that natural-language instructions steer Large Language Models (LLMs) toward predictable, optimized outputs. This *prompt-as-design* paradigm treats prompt formulation as the primary driver of model behavior, implicitly assuming that LLMs function as near-deterministic mappings from inputs to outputs. Consequently, most LLM evaluation work selects a prompt x, runs the model once, and interprets the resulting output y as the model's behavior under that prompt [4, 27, 34]. Variability induced by sampling from the underlying conditional distribution is rarely measured and typically ignored.

This paradigm stands in contrast to the probabilistic nature of LLMs: Formally, LLMs implement a conditional distribution $p_\theta(y \mid x)$ over outputs. Decoding procedures such as temperature scaling or nucleus sampling yield realizations of a random variable rather than a fixed response. Even for a temperature set to zero, practical inference stacks exhibit nondeterminism due to GPU parallelism and batching effects [22]. When this source of variance is ignored in experiments, it is effectively absorbed into other variance components such as model choice and prompt effects, potentially inflating the apparent magnitude of prompt-induced differences. Without quantifying sampling variance, it remains unknown whether observed effects reflect genuine prompt sensitivity, model choice, or unmeasured within-LLM stochasticity.

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

The central question of our paper is: *How is variance in LLM outputs partitioned across different factors: task, prompt, model choice, and within-LLM variance?* We test this question in a setting where output variability is not noise, but a defining feature: creative idea generation. We select creative tasks as our experimental testbed for two pragmatic reasons. First, they provide a high-entropy output space where $p_\theta(y \mid x)$ is multi-modal; unlike mathematical or factual tasks with peaked distributions, open-ended generation requires the model to traverse a large semantic solution space, making it a sensitive stress-test for disentangling prompt-induced effects from sampling variability. Second, we employ the Alternative Uses Task (AUT; 21), an established measure of divergent (i.e., free-associative) thinking, with a validated, fully automated evaluation pipeline [20]. This allows for the large-scale repeated sampling necessary to partition variance components with a scale that is prohibitive for human-based evaluation. Our findings show a consistent hierarchy: for originality, *what you ask about* explains 40% of variance, *how you ask* explains 33%, stochastic sampling explains 17%, and *which model you ask* explains less than 8%—which qualifies the common assumption that model selection is the primary design decision. Practically, teams building brainstorming or tagline tools routinely switch models while holding the prompt fixed, yet prompt design and repeated sampling are at least as important and often cheaper to change (Section 5). Our key contributions are:

- **Task content dominates originality.** Across 30 AUT items, what object the model reasons about explains 40.46% of originality variance, more than any other factor, including prompt strategy (32.55%) and model choice (7.58%). When decomposing variance within each item separately, the prompt strategy rises to 60.3% vs. 17.3% for the model, indicating that prompt effects are not driven by item-level differences. For fluency, the pattern inverts: model choice dominates (55.14%).
- **Stochasticity exceeds model differences.** Within-LLM variance (16.64%) exceeds model choice by more than $2\times$ for originality, meaning single-sample evaluations risk conflating noise with genuine effects.
- **Prompts are discriminative, not deterministic.** "Persona" prompts elicit high-variance, high-quality responses from capable models, while format constraints suppress diversity, indicating that prompts shape output *distributions*, not point estimates.

2 Related work

2.1 Prompt engineering and benchmarking

Research in prompt engineering treats instructions as optimizable parameters for controlling model behavior, from manual heuristic refinement [17] to automated optimization [1, 16]. Robustness benchmarks perturb prompts via paraphrasing or adversarial noise [4, 27, 34], reporting large performance swings, yet typically fix decoding (e.g., temperature $= 0$) and draw a *single* sample per variant, collapsing the output distribution to one realization per condition. More complex frameworks inherit this determinism: DIPPER [13] constructs inference-time ensembles via diverse prompts but explicitly treats prompt variation—not sampling stochasticity—as the meaningful source of behavioral diversity. Together, these patterns reinforce a prompt-centric view in which sampling stochasticity is assumed negligible and left unmeasured.

2.2 Stochasticity, creativity, and inference variability

LLMs are probabilistic by definition, yet stochasticity is frequently neglected or treated as a challenge to be minimized. Techniques like Self-Consistency [6] or test-time compute scaling [29] aggregate multiple samples to improve reliability, effectively merging the distribution $p_\theta(y \mid x)$ into a single robust prediction. However, these approaches treat sampling diversity as a resource to be *collapsed* by an aggregation operator, rather than as an object of measurement; they do not quantify the magnitude of sampling-induced variance nor compare it to that induced by prompt changes. Recent work on "LLM drift" [22] explicitly documents that setting temperature to 0 does not guarantee fully deterministic results due to GPU parallelism, batching behavior, and non-associative floating-point operations, yet rarely quantifies this variance relative to prompt effects.

Studies on LLM creativity suggest that model configuration substantially affects output diversity: Highly aligned models may trade output-variety for consistency, while base models produce more varied outputs [18, 31]. Evidence of substantial intra-model variability has been documented in

creativity evaluations: Haase et al. [10] conducted 100 repeated trials per model under identical prompts and observed outputs ranging from below-average to highly original, with performance distributions spanning the full range from the bottom quartile to the top quartile. Yet with a single sample per prompt, it is unclear whether a difference between prompts reflects the prompt or sampling randomness. The relative contribution of within-LLM versus between-prompt variance remains unknown.

Because a variance decomposition depends on the output space of the task, the choice of creativity measure matters. AUT is a divergent-thinking task [7, 28]: it asks for many novel responses to an open prompt, unlike association tasks (Divergent Association Task, Remote Associates Test), which probe semantic memory [2, 10], or product-oriented tasks (story/metaphor generation, design ideation), which are scored holistically. Applied to LLMs, it tests the ability to retrieve and recombine uncommon but relevant semantic associations, one facet of creativity rather than a proxy for narrative or design creativity. It is the most widely used LLM creativity benchmark with a validated automated scorer [10, 21]. Our ordering should transfer to other many-answer divergent tasks, but not necessarily to product-oriented or constrained tasks.

2.3 Methodological gap: the sampling fallacy

The experimental design of most prompting studies mirrors a known bias in experimental psychology, termed the "fixed-effect fallacy" [14], in which stochastic components (e.g., stimuli or samples) are incorrectly treated as fixed factors. In the LLM context, taking a single generation as representative of a prompt's distribution commits a similar error. If $p_\theta(y \mid x)$ exhibits substantial spread, any single realization may deviate considerably from the central tendency.

The core problem [14, 15] is a mismatch between the data-generating process and the analytic model. When multiple random factors contribute, collapsing one factor attributes its variance to others. Standard ANOVA procedures assume independence of observations; when within-person correlation is ignored, error variance is misestimated, which can bias effect sizes and inflate Type I error rates. The same misspecification also lowers power: an underestimated error term makes genuine effects harder to detect (Type II error). Single-sample evaluation conflates prompt effects with sampling variability, potentially leading to an overestimation of prompt effects.

3 Methodology

We apply a repeated-sampling design to analyze the full breadth of potential variability by generating 50–100 independent outputs per condition and decomposing total variance into prompt, model, item, and within-LLM components via sum-of-squares partitioning. Because the design is nearly balanced and all factors are fully crossed, this approach yields the same proportional estimates as a random-effects variance-component model [26] while remaining transparent and easy to replicate. Bootstrap confidence intervals provide valid uncertainty estimates without distributional assumptions.

3.1 Stimuli: prompt design

We designed a baseline prompt (P1) following the standard AUT instructions [7], which asks for alternative uses of an everyday object. To ensure generalizability, we employed 30 distinct AUT items spanning a broad range of everyday objects (e.g., ball, brick, chair, knife, paperclip, shoe, tire; see Appendix A.6 for the full list). We tested five strategic prompting variations alongside the baseline: providing a one-shot example (P2; 5), heuristic/domain exploration cues (P3; 33), anticipatory avoidance instructions (P4; 30), zero-shot chain-of-thought (P5; 5), and a creative persona assignment (P6; 19). For one item (plastic bottle), we additionally tested four minor prompt variations: paraphrasing (P7), formatting constraints (P8; 12), information reordering (P9), and typographical noise injection (P10; all following 25). All prompts concluded with identical JSON formatting instructions. Full prompt texts are provided in Appendix Tables 2 and 3.

3.2 Inference configuration

In deployed settings, users do not interact with model weights in isolation; instead, they interact with a model-as-service that bundles architecture, training, alignment, default decoding parameters,

and system prompts into a single interface. Our design reflects this reality: proprietary LLMs were accessed via their official provider APIs, while smaller open-weight models (Llama 3.3, Mistral Nemo, Gemma 3, and GPT OSS) ran locally on a partitioned NVIDIA H100 NVL (MIG 7g 94GB) using Ollama v0.21.0. Data collection took place between November 2025 and April 2026. Exact model version identifiers for each provider API are reported in Table 1 (Appendix).

We used the default hyperparameter settings provided by each provider (i.e., no manual overrides to temperature, top-p, or maximum output length). This choice reflects three considerations: (1) standardizing parameters creates false equivalence since temperature effects vary across models [23]; (2) defaults represent vendor-optimized configurations; (3) using defaults maximizes ecological validity. Variance attributed to the model factor reflects the combined effect of architecture, alignment, and provider-specific defaults (see Limitations). Appendix A.12 separates open- and closed-weight models, and Appendix A.11 reports a fixed-seed replication for the locally hosted models.

3.3 Procedure and data collection

We evaluated 10 LLMs spanning six providers: six reasoning models—Claude Sonnet 4.5 (Anthropic), DeepSeek V3.2 (671B; DeepSeek), Gemini 3.1 Pro (Google), GPT 5.1 and GPT 5.2 (OpenAI), and GPT OSS 120B (OpenAI)—and four standard models—Grok 4.1 (xAI), Gemma 3 27B (Google), Llama 3.3 70B (Meta), and Mistral Nemo 12B (Mistral). Each response is one *idea-generation* episode: the model lists alternative uses for a single object. We use *idea generation* and *AUT response generation* interchangeably.

For each model $\times$ prompt $\times$ AUT item combination, we generated 50–100 independent outputs. Each run constituted a separate API call with no shared conversation history, no system prompts, and only a few-shot context beyond the stimulus itself, ensuring statistical independence across runs. The primary design covers 10 models $\times$ 6 prompts (P1–P6) $\times$ 30 items $\times$ 50 runs, yielding 1,800 cells and $N = 89{,}806$ valid scored responses after excluding malformed JSON, empty responses, and API timeout errors. In total, 194 of the 90,000 intended generations (0.22%) were excluded. Cell sizes remain nearly balanced (at least 48 valid runs in all but two cells), so the impact on the decomposition is negligible. Cells with more than 50 valid runs were randomly subsampled to 50 to ensure a balanced design; a convergence analysis (Appendix A.8) confirms that variance estimates are stable from $N \approx 20$ onward, so no precision is lost.

The complete data-collection process included creative-response generation by the LLMs and subsequent scoring with the AUT service, which itself relies on LLM-based evaluation. By logging execution time per file, we estimated a total runtime of approximately 15 days. This can be reduced substantially through more aggressive parallelization. The primary bottleneck was locally hosted Ollama models running on a single GPU, which constrained the number of models that could be loaded concurrently. On average, one full run (generation plus AUT scoring) took 36 hours (minimum: 20 hours; maximum: 3 days).

One item (plastic bottle) was additionally evaluated with four minor prompt variations—paraphrasing (P7), formatting constraints (P8; 12), information reordering (P9), and typographical noise injection (P10; 25)—to assess sensitivity to surface-level prompt changes rather than strategic prompting variations ($N = 2{,}493$; 10 models $\times$ 5 prompts $\times$ 50 runs; cf. Appendix A.2). Because item choice accounts for $\sim$40% of the originality variance in the main analysis, we fix the item here to isolate the prompt effect without item-level confounds. All responses were required to adhere to a prescribed JSON schema specifying a single key (the object name) mapped to an array of string ideas.

3.4 Evaluation & analysis

Scoring. We evaluated outputs with the Open Creativity Scoring system (OCSAI; 20). OCSAI embeds each idea and computes its semantic distance from a reference set of common responses for the same item, mapping this distance to an *originality* score on a 1–5 scale. Because the comparison is against item-specific human responses rather than token frequencies, scores reflect semantic novelty rather than surface-form rarity. Scoring is *item-conditional*: an idea scored 4.0 for "knife" reflects the same within-item rarity as a 4.0 for "shoe," even though items differ in their base rates. Reference distributions are built from human AUT responses and validated against human raters across multiple datasets ($r \approx .81$, 21), including follow-up analysis on LLM-generated AUT outputs from

our group [9]. The scorer is publicly available, enabling replication. OCSAI scores originality per idea; we compute response-level originality as the mean over valid ideas. The scorer does not by itself check physical feasibility, a limitation we quantify in Appendix A.10.

Fluency was defined as the count of valid ideas generated per response. To ensure validity, we applied an automated parsing and filtering pipeline: (a) extraction from the required JSON structure, (b) removal of empty strings and formatting artifacts, and (c) deduplication of exact string matches within each response. After filtering, the dataset contains 89,806 valid scored responses.

Variance Decomposition. We partition total variance with a linear model containing four factors. For a creativity score Y_{mipr} (model m, item i, prompt p, run r):

$$Y_{mipr} = \mu + \alpha_m + \gamma_i + \beta_p + \delta_{mp} + \epsilon_{mipr} \tag{1}$$

where μ is the fitted grand-mean intercept (not a variance component), α_m is the effect of model m, γ_i is the effect of AUT item i, β_p is the effect of prompt p, δ_{mp} is the model×prompt interaction (we avoid $(\alpha\beta)_{mp}$ to preclude reading it as a product of main effects), and ϵ_{mipr} is the within-cell residual. We center the observations at $\hat{\mu}$ and partition the remaining sum of squares. Each reported component is that factor's sum of squares divided by the total sum of squares of the centered observations, so the shares in a given fit sum to 100% [3]. The item term γ_i appears only in the unconditional fit (Figure 1a). The item-conditioned fit (Figure 1b) refits the model separately within each item, so item variance is removed by conditioning rather than by renormalization. We obtain 95% confidence intervals by stratified bootstrap resampling (5,000 iterations): runs are resampled with replacement within each model×prompt×item cell, the decomposition is recomputed, and percentile intervals are taken [8]. Appendix A.15 gives a schematic of the design-to-decomposition mapping.

4 Results

4.1 Variance decomposition

Our variance decomposition (Figure 1) shows a clear divergence between *originality* and *fluency* and identifies the AUT item as the largest single factor for originality. All main effects (item, prompt, model) and the model×prompt interaction were significant for both metrics (all $p < .001$).[1]

For **Originality** (Figure 1a; descriptive data in Table 4), the dominant source of variance is the AUT item, i.e., *what* object the model is asked about (40.46%). Prompt strategy is the second-largest factor (32.55%), followed by within-LLM stochasticity (16.64%). Model choice explains only 7.58%, with an additional 2.77% from model×prompt interactions. Together, task content and prompt strategy explain over 72% of originality variance, while model choice accounts for less than 8%. An extended 7-factor decomposition (Table 6) further partitions the within-LLM residual into item×prompt (4.87%), item×model (2.30%), and pure stochasticity (9.48%), the last of which still exceeds model choice alone. The model×prompt interaction indicates that some prompts are differentially effective across models (Figures 4a and 6). Bootstrap 95% CIs for all components are narrow (< 0.5 pp; Table 6).

The dominance of item effects reflects substantial differences in what objects afford creative reinterpretation. Mean originality ranges from 4.17 (backpack, shoe) to 2.53 (knife), a span of 1.64 points on a 5-point scale (see Appendix A.6 for full ranking). Items with high functional versatility and physical manipulability (backpack, shoe, chair) consistently elicit more original responses, while items with a dominant canonical function (knife, rope) or simple geometry (brick, box) constrain the semantic search space. This item-level variance is not reducible to prompt or model effects: it persists across all 10 models and 6 prompts, suggesting that the "affordance structure" of the task stimulus is a major driver of creative output that single-item designs cannot capture.

Item-Conditioned Decomposition. The 40.46% item variance is largely a measurement property of the AUT: some objects simply afford richer creative reinterpretation than others (e.g., backpack vs. knife). This between-item heterogeneity is not of primary interest for understanding what drives LLM creative behavior. Our core question is: *given a specific task, is creative output shaped more*

[1] Originality: $F_{\text{prompt}}(5,\ 89{,}717) = 35{,}149$, $F_{\text{item}}(29,\ 89{,}717) = 7{,}533$, $F_{\text{model}}(9,\ 89{,}717) = 4{,}546$, $F_{\text{m}\times\text{p}}(45,\ 89{,}717) = 336$. Fluency: $F_{\text{prompt}} = 3{,}685$, $F_{\text{item}} = 63$, $F_{\text{model}} = 18{,}393$, $F_{\text{m}\times\text{p}} = 549$. All $p < .001$.

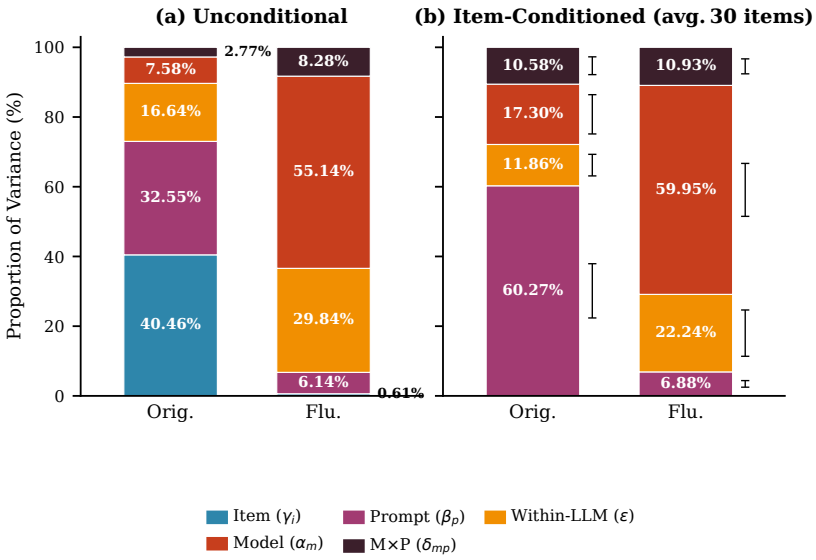


Figure 1: Variance decomposition for originality and fluency ($N = 89{,}806$, 30 AUT items, 10 models, 6 prompts, 50 runs/cell). **(a)** Unconditional fit of Eq. 1: AUT-item effects (γ_i) dominate originality (40.46%), while model choice (α_m) dominates fluency (55.14%). **(b)** AUT-item-conditioned fit (mean ± 1 SD across the 30 per-item fits): prompt strategy (β_p) becomes the dominant factor for originality (60.3%), exceeding model choice (17.3%) by 3.5:1.

by the prompt, the model, or sampling randomness? To isolate these controllable factors, we repeat the variance decomposition *within* each of the 30 items separately (dropping the item factor) and average the resulting proportions (Figure 1b). For originality, prompt strategy becomes the dominant factor at 60.3% (SD $= 7.8$), followed by model at 17.3% (SD $= 5.6$), within-LLM stochasticity at 11.9% (SD $= 3.1$), and model$\times$prompt interactions at 10.6% (SD $= 2.6$). Prompt dominance holds for all 30 items (min 41.0% for sock; max 72.5% for tile). Model's share rises from 7.6% (unconditional) to 17.3% (conditional). This is not a renormalization artifact: deleting the item sum of squares and rescaling the remaining components would leave within-LLM variance above model variance (prompt 54.7%, within-LLM 28.0%, model 12.7%). The rise arises because (i) the per-item refits absorb the global item$\times$model (2.30%) and item$\times$prompt (4.87%) interactions into the within-item model and prompt components, and (ii) averaging per-item ratios differs from taking a ratio of pooled sums of squares. The within-item residual (11.9%) is correspondingly closer to pure stochasticity. Appendix A.14 gives a step-by-step numerical illustration. This pattern also explains why single-item designs tend to overestimate model effects: with only one item, item-level variance is undetectable and is allocated to the remaining components. The multi-item design separates these sources and shows that, once item effects are accounted for, prompt strategy consistently dominates model choice.

By contrast, for **Fluency** (Figure 1a; descriptive data in Table 5), model choice dominates (55.14%), followed by within-LLM variance (29.84%), the model$\times$prompt interaction (8.28%), prompts (6.14%), and a negligible AUT item effect (0.61%). Prompts steer *quality* but have little impact on *quantity*; output volume is largely an intrinsic property of the model's decoding behavior and alignment.

4.2 Mechanisms of model variance

Although model choice explains only 7.58% of originality variance (vs. 55.14% for fluency), the between-model differences are systematic, reflecting distinct alignment strategies (Figure 5). We observe several distinct model behaviors:

Reasoning-Oriented Architectures (High Quality, Low Fluency). Models like Gemini 3.1 Pro prioritize depth over breadth. Gemini 3.1 Pro achieves the highest mean originality (3.85) but gen-

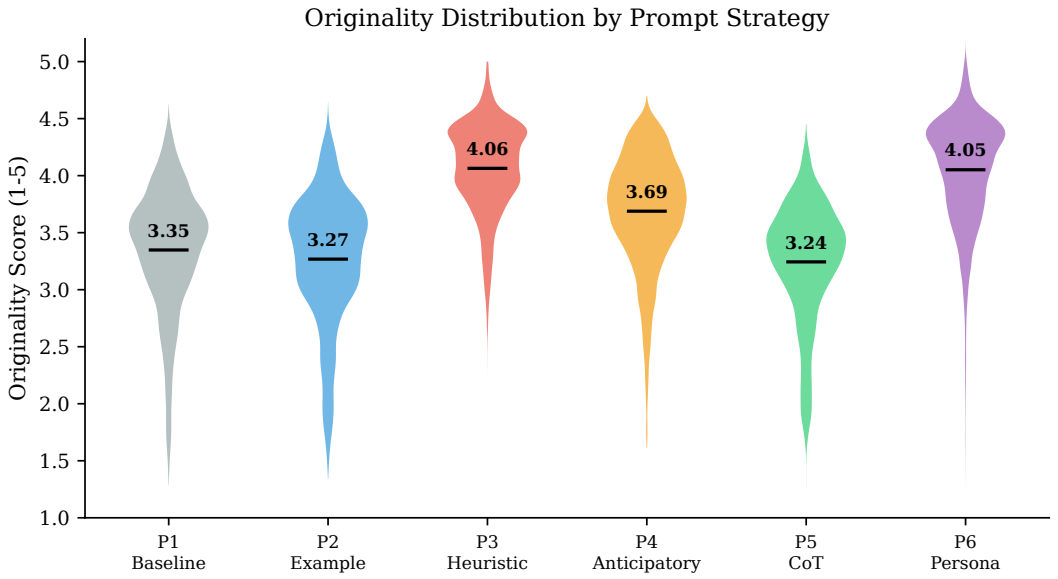


Figure 2: Violin plots displaying originality scores as a function of prompt strategy across all 30 AUT items and 10 models. **Prompt key**: **P1**: direct baseline. **P2**: one-shot example. **P3**: heuristic/domain. **P4**: anticipatory. **P5**: chain-of-thought. **P6**: creative persona.

erates the fewest ideas (avg. 8.7). This pattern suggests that deliberative reasoning steps prune lower-quality candidates in favor of focused, creative output. However, this effect appears specific to models architecturally designed for reasoning; the simple prompt asking for *chain-of-thought* reasoning (P5) did not improve originality across all models (see Figure 2).

High-Ceiling Models. Mean performance can obscure "high-ceiling" capabilities. Grok 4.1 ranks 3rd in overall mean originality (3.82) but is the most volatile model (Rank Variance = 8.70), ranking 1st on P1, P3, and P6 but falling to 7th on P2 and P5. As this peak performance occurs only under specific prompts, average scores mask these peaks. Gemini 3.1 Pro (3.85) and GPT 5.1 (3.83) also showed high-ceiling capability (Figure 9).

High-Fluency Architectures (High Fluency, Moderate Quality). Models like GPT 5.1 and GPT 5.2 prioritize volume, generating > 100 ideas per prompt on average (GPT 5.2: 110.3; GPT 5.1: 103.2), compared to a grand mean of 38.3. While their peak quality is competitive (GPT 5.1 ranks 2nd overall at 3.83), average originality is diluted by the volume of output. This quality-quantity trade-off is visible in Figure 5 (Appendix): models cluster into a "reasoning-oriented" region (high originality, low fluency) and a "high-fluency" region (high volume, moderate quality).

Ranking Consistency. Models differ in their reliability across prompts (see Figure 3). Gemini 3.1 Pro is highly stable (Rank Variance = 0.57), consistently ranking in the top 3 regardless of the prompt. GPT 5.1 is similarly stable (Rank Variance = 0.80). In contrast, Grok 4.1 is highly volatile (Rank Variance = 8.70), ranking 1st on the Persona and Heuristic prompts but falling to 7th on Example and CoT. Gemma 3 27B also shows high volatility (Rank Variance = 4.80). This indicates that some models are "generalist" creators while others are "specialists" that require specific triggers (e.g., Persona prompt).

4.3 Mechanisms of prompt control

We observe two opposing mechanisms behind the 32.55% of originality variance attributable to prompts (Figure 1a, Table 4): increasing quality in specific models versus constraining the output space. Across all prompts, we find that strategic prompts like the Heuristic/Domain prompt (P3, mean = 4.06), the Persona prompt (P6, mean = 4.05), and the Anticipatory prompt (P4, mean = 3.69) elicited higher originality than the baseline (P1, mean = 3.35), but also higher between-

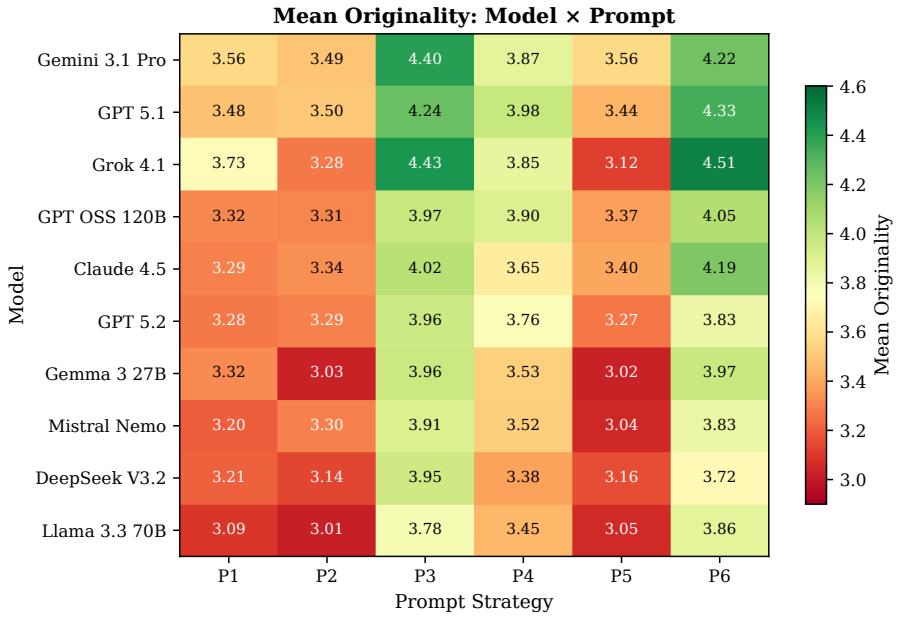


Figure 3: Heatmap of mean originality scores (model × prompt), aggregated across 30 AUT items. Prompt key as in Figure 2.

model variance. Instructions with one example (P2, mean = 3.27) or the prompt to think step-by-step (P5, mean = 3.24) did not improve output quality; P5 actually yielded the lowest mean originality. In a supplementary analysis on a single item (plastic bottle) with minor prompt variations (P7–P10), Paraphrasing (P7) and Information Reordering (P9) showed small positive effects on originality ($d = 0.63$ and 0.53, respectively), while Format Constraint (P8) and Typo injection (P10) had negligible effects ($d = 0.10$ and -0.02; Section A.7).[2]

Discriminative vs. Constraining Prompts. These two mechanisms are illustrated by contrasting P6 (Persona) and P8 (Format). The Persona prompt ("You are the most creative person...") acts as a discriminative stress test: it yields the highest between-model variance (0.065 vs. baseline P1: 0.036), with Grok 4.1 responding most strongly (Cohen's $d = 1.69$, mean 4.51), followed by GPT 5.1 ($d = 1.48$) and Claude Sonnet 4.5 ($d = 1.33$), while GPT 5.2 ($d = 0.60$) and DeepSeek V3.2 ($d = 0.67$) improve only modestly—explaining the significant model×prompt interaction (Figure 3). Conversely, the Format Constraint (P8: "no titles or colons"), tested on plastic bottle ($N = 50$/cell), systematically degrades fluency for high-output models without affecting originality (Figure 10): GPT 5.1 loses 65.1 ideas (from 190.0 to 124.9) and GPT 5.2 loses 32.5 (from 177.4 to 145.0), while lower-fluency models (Grok 4.1, Llama 3.3 70B, Mistral Nemo) show negligible changes. Together, these cases show that prompts do not shift a single mean but reshape the output distribution—widening it (P6) or compressing it (P8)—with effects that depend on model architecture.

5 Discussion

The field of prompt engineering has largely operated on the premise that prompts act as close-to-deterministic programs [24, 32]: find the right instruction, and the model will reliably execute it. Our variance decomposition (Section 4.1) complicates this assumption. For originality, task content and prompt strategy jointly explain over 72% of variance, while model choice contributes less than 8%—and within-LLM stochasticity alone exceeds model differences by more than 2×. Once the task is controlled for, the imbalance grows: prompt strategy accounts for 60% of within-item variance vs. 17% for model choice, meaning that the prompt-over-model advantage holds regardless

[2]All Cohen's d values for P7–P10 are computed against the pooled within-condition SD of response-level originality scores (not model-level means), making them comparable to standard effect-size benchmarks.

of which object is being evaluated. This suggests that prompting is better understood as steering a probabilistic distribution whose shape depends on task content than as programming a deterministic function. While we demonstrate this in creativity, the underlying mechanism—that within-model variance can exceed between-model differences—likely applies to any open-ended generative task where $p_\theta(y \mid x)$ is multi-modal, from story generation to code synthesis.

Implications for Practice. *(1) Benchmarking.* Report variance components alongside point estimates, and prefer discriminative prompts: between-model variance is 0.036 for P1 vs. 0.065 for P6. Avoid single-shot evaluation. A pilot estimate of within-cell variance sets the runs needed, about 20 to 160 per model$\times$prompt condition for ± 0.1 points at 95% confidence (Appendix A.13). *(2) Prompt engineering.* Prompt effects are positive across all ten models but vary in magnitude: the P6 gain over P1 ranges from $\Delta = +0.51$ (DeepSeek V3.2) to $\Delta = +0.90$ (Claude Sonnet 4.5), a $1.76\times$ ratio, so prompts should be tuned per model. *(3) Deployment.* Within-LLM variance is also a resource: Grok 4.1 places 7.7% of its Persona responses in the top 1% of originality scores (Figure 9b), so repeated sampling with selection beats single-shot inference.

Scope and generality. Our empirical ordering is established for AUT-style divergent ideation. The methodology, a fully crossed design plus variance decomposition, applies to any task with a repeated-sampling quality metric. The ordering itself should not be assumed to transfer: for constrained tasks such as mathematics, coding, or factual QA, the output space is nearly single-modal and we expect a stronger model component. Within creativity, AUT primarily taps semantic retrieval and recombination. Narrative creativity, metaphor generation, and problem formulation may decompose differently. Cross-item stability across the 30 AUT stimuli (Kendall's $W = 0.79$ for originality ranks in a three-model check) supports generalization across AUT items, not beyond the AUT domain. A caveat: switching models is often easier than redesigning prompts, and matching a model to a prompt yields gains up to $\Delta = +0.90$ (Claude Sonnet 4.5, P6 vs. P1). The lesson is to tune prompts per model rather than treat either lever as dominant.

Probabilistic Prompting: Shaping Distributions, Not Outputs. Our findings show that a prompt defines a *distribution* of likely outputs, not a single result. For discriminative prompts like the "Creative Persona" (P6), this distribution shifts upward in quality but also increases in variance. This means that "better" prompts often yield *less* predictable behavior. This creates a trade-off for developers: the prompts that elicit the highest originality (P6) are also the least stable, while the format-constrained prompt (P8) suppresses idea diversity. Effective prompting systems must therefore treat outputs as samples, with multiple candidates ($N > 1$) generated to capture the distribution's shape and enable statistical inference.

The Need for Discriminative Benchmarking. Current benchmarks often employ generic instructions that fail to distinguish between model capabilities. We found that baseline prompts (P1) yielded low between-model variance (0.036), making them poor tools for comparison. In contrast, the "Persona" prompt (P6) acted as a stress test (variance 0.065), showing substantial differences in how models respond to identity-based framing. We cannot attribute Grok's and Gemini's stronger responses to specific training procedures, but their differential sensitivity to persona framing suggests that some models are more responsive to identity-based instructions than to direct task commands. Future creativity benchmarks should therefore prioritize *discriminative prompts*, that is, prompts that differentiate between model capabilities through varied framing.

Resolving the Sampling Fallacy. Our results directly address the methodological concerns raised in Section 2. The "fixed-effect fallacy" [14] predicts that treating stochastic factors as fixed inflates other variance components. Our data confirm this quantitatively: within-LLM variance accounts for 16.64% of originality variance in the 5-factor model (and 9.48% as pure stochasticity in the 7-factor decomposition), exceeding the 7.58% attributed to model choice. Any single-sample evaluation that ignores these residual risks misattributes sampling noise to prompt or model effects. The concern that Self-Consistency [6] and test-time compute scaling [29] collapse $p_\theta(y \mid x)$ without measuring its shape is borne out by our finding that "better" prompts (P6) widen, rather than narrow, the output distribution—making aggregation-based approaches particularly lossy for creative tasks. Similarly, the observation by Paleyes et al. [22] that temperature-zero inference is not fully deterministic is consistent with the substantial within-LLM variance we observe even under provider-default decoding. Finally, extending the single-item, single-model design of Haase et al. [10] to 10 models and 30

items reveals that the variability they documented is not an artifact of one model or one task: it is a structural feature of the generative process, with item content and prompt strategy jointly dwarfing model identity as sources of variation.

Beyond $N = 1$: Implications for Evaluation. If within-LLM variance explains 17–30% of total variance, many reported prompt or model comparisons may reflect sampling noise rather than genuine effects. Current evaluation practices that rely on single generations effectively treat LLM outputs as deterministic point estimates, an assumption our results show to be unreliable. The risk is not uniform: for stable models like Gemini 3.1 Pro (Rank Variance $= 0.57$), small samples may suffice for coarse ranking, but for volatile models like Grok 4.1 (Rank Variance $= 8.70$), a single sample risks rank-reversal. This applies beyond creativity: any benchmark on open-ended generation—summarization, code generation, long-form QA—faces the same confound when within-model variance is unquantified. We recommend that future benchmarks report variance components alongside means, enabling valid cross-model inference.

Limitations. *Task scope.* Our study evaluates 10 LLMs across 30 AUT items with 6 core prompting strategies and 50 repeated samples per condition ($N = 89{,}806$). While the AUT is a well-validated creativity measure [2, 21, 28] and our multi-item design substantially improves generalizability over single-item studies, future work should validate these findings across other creative domains (e.g., story generation, design tasks). Furthermore, we did not directly prompt for fluency, so this measure assesses default output volume rather than the capacity to generate more ideas on demand; response length and fluency correlate at $r = 0.98$, so fluency differences are largely verbosity differences.

Dimensions of creativity. OCSAI supports originality and fluency. Flexibility and elaboration are also standard AUT dimensions. Recent work scores flexibility with LLMs [11], but a validated elaboration scorer is not available, and adding either dimension would require re-scoring the corpus. We therefore do not claim that our decomposition exhausts all creativity dimensions.

Semantic validity. The scorer rewards semantic distance and does not enforce physical feasibility. Our audit (Appendix A.10) covers the plastic-bottle item only and relies on LLM raters with low agreement. It bounds but does not eliminate the concern that infeasible ideas inflate scores. Human adjudication of a larger sample remains future work.

Semantic diversity. We report exact-duplicate rates and qualitative examples (Appendix A.13) but do not cluster ideas into semantic modes, which is future work.

Confounded variance components. Our "model" factor bundles architecture, training data, alignment, and default parameters; "within-LLM variance" includes unmodeled higher-order interactions. An extended 7-factor decomposition and the item-conditioned analysis address these confounds in detail (Appendix A.9).

Statistical model. Our variance decomposition uses sum-of-squares partitioning with bootstrap confidence intervals rather than a formal mixed-effects model. For a nearly balanced, fully crossed design, the two approaches yield equivalent variance proportions; we chose the former for transparency. For fluency (count data), the same linear decomposition is an approximation; however, the qualitative ranking of components is robust across sensitivity checks, and our primary claims concern originality.

6 Conclusion

We decompose variance in LLM creative output at scale ($N = 89{,}806$; 30 items, 10 models, 6 prompts) and find that common assumptions about what drives LLM behavior are inverted: for originality, *what* you ask about (40%) matters more than *how* you ask (33%), which matters far more than *which model* you ask (8%), and within-LLM stochasticity (17%) alone exceeds model differences, making single-sample evaluations unreliable. Within individual items, prompt strategy dominates even more strongly (60% vs. 17% model), meaning that investing in prompt design yields larger returns than switching models. Practically, this argues for variance-aware benchmarking, model-specific prompt tuning, and repeated sampling with selection. These findings call for a shift from treating LLM outputs as deterministic point estimates toward principled sampling designs that quantify the full variance structure of generative systems.

Acknowledgments and Disclosure of Funding

Research reported in this paper was partially supported through the Research Campus Modal funded by the German Federal Ministry of Research, Technology and Space (BMFTR) (fund numbers 05M14ZAM, 05M20ZBM) and the German Research Foundation (DFG) through the DFG Cluster of Excellence MATH+ (EXC-2046/1, project ID 390685689, project AA3-15), by the German Federal Ministry of Research, Technology and Space (BMFTR), grant number 16DII143 (Weizenbaum Institute), by the German Research Foundation (DFG), CRC 1404: "FONDA: Foundations of Workflows for Large-Scale Scientific Data Analysis" (Project-ID 414984028) and by the Zuse Institute Berlin via RISE@ZIB services for hosting the LLM models.

Competing interests. The authors declare no competing financial interests.

References

[1] Lakshya A. Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J. Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alexandros G. Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. GEPA: Reflective Prompt Evolution Can Outperform Reinforcement Learning. In *Proceedings of the Fourteenth International Conference on Learning Representations (ICLR)*, 2026. doi: 10.48550/arXiv.2507.19457.

[2] Roger E. Beaty, Dan R. Johnson, Daniel C. Zeitlen, and Boris Forthmann. Semantic Distance and the Alternate Uses Task: Recommendations for Reliable Automated Assessment of Originality. *Creativity Research Journal*, 34(3):245–260, July 2022. ISSN 1040-0419, 1532-6934. doi: 10.1080/10400419.2022.2025720.

[3] Paul D. Bliese. Within-group agreement, non-independence, and reliability: Implications for data aggregation and analysis. In Katherine J. Klein and Steve W. J. Kozlowski, editors, *Multilevel Theory, Research, and Methods in Organizations: Foundations, Extensions, and New Directions*, pages 349–381. Jossey-Bass, San Francisco, CA, 2000.

[4] Bowen Cao, Deng Cai, Zhisong Zhang, Yuexian Zou, and Wai Lam. On the Worst Prompt Performance of Large Language Models. *Advances in Neural Information Processing Systems*, 37:69022–69042, December 2024. doi: 10.52202/079017-2205.

[5] Banghao Chen, Zhaofeng Zhang, Nicolas Langrené, and Shengxin Zhu. Unleashing the potential of prompt engineering for large language models. *Patterns*, 6(6):101260, June 2025. ISSN 2666-3899. doi: 10.1016/j.patter.2025.101260.

[6] Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. Universal Self-Consistency for Large Language Model Generation, November 2023. doi: 10.48550/arXiv.2311.17311.

[7] P. Christensen, J. Guilford, R. Merrifield, and R. Wilson. *Alternate Uses Test*. Beverly Hills, CA: Sheridan Psychological Service, 1960.

[8] Bradley Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*, volume 57. Chapman & Hall, New York, 1993. ISBN 978-0-412-04231-7.

[9] Jennifer Haase and Paul H. P. Hanel. Artificial muses: Generative artificial intelligence chatbots have risen to human-level creativity. *Journal of Creativity*, 33(3):1–7, 2023. ISSN 2713-3745. doi: 10.1016/j.yjoc.2023.100066.

[10] Jennifer Haase, Paul H. P. Hanel, and Sebastian Pokutta. Has the creativity of large-language models peaked? An analysis of inter- and intra-LLM variability. *Journal of Creativity*, 35(3): 100113, December 2025. ISSN 2713-3745. doi: 10.1016/j.yjoc.2025.100113.

[11] Eran Hadas and Arnon Hershkovitz. Using large language models to evaluate alternative uses task flexibility score. *Thinking Skills and Creativity*, 52:101549, June 2024. ISSN 1871-1871. doi: 10.1016/j.tsc.2024.101549.

[12] Jia He, Mukund Rungta, David Koleczek, Arshdeep Sekhon, Franklin X. Wang, and Sadid Hasan. Does Prompt Formatting Have Any Impact on LLM Performance?, November 2024. doi: 10.48550/arXiv.2411.10541.

[13] Wenyang Hu, Gregory Kang Ruey Lau, Liu Diwen, Chen Jizhuo, See-Kiong Ng, and Bryan Kian Hsiang Low. Dipper: Diversity in Prompts for Producing Large Language Model Ensembles in Reasoning Tasks. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 35546–35560, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1801.

[14] Charles M. Judd, Jacob Westfall, and David A. Kenny. Treating stimuli as a random factor in social psychology: A new and comprehensive solution to a pervasive but largely ignored problem. *Journal of Personality and Social Psychology*, 103(1):54–69, 2012. ISSN 1939-1315. doi: 10.1037/a0028347.

[15] Charles M. Judd, Jacob Westfall, and David A. Kenny. Experiments with More Than One Random Factor: Designs, Analytic Models, and Statistical Power. *Annual Review of Psychology*, 68(1):601–625, January 2017. ISSN 0066-4308, 1545-2085. doi: 10.1146/annurev-psych-122414-033702.

[16] Xiaoqiang Lin, Zhaoxuan Wu, Zhongxiang Dai, Wenyang Hu, Yao Shu, See-Kiong Ng, Patrick Jaillet, and Bryan Kian Hsiang Low. Use Your INSTINCT: INSTruction optimization for LLMs usIng Neural bandits Coupled with Transformers. In *Proceedings of the Forty-First International Conference on Machine Learning (ICML)*, 2024. doi: 10.48550/arXiv.2310.02905.

[17] Vivian Liu and Lydia B. Chilton. Design Guidelines for Prompt Engineering Text-to-Image Generative Models. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, pages 1–23, New York, NY, USA, April 2022. Association for Computing Machinery. ISBN 978-1-4503-9157-3. doi: 10.1145/3491102.3501825.

[18] Behnam Mohammadi. Creativity Has Left the Chat: The Price of Debiasing Language Models, 2024. doi: 10.2139/ssrn.4858364.

[19] Carlos Olea, Holly Tucker, Jessica Phelan, Cameron Pattison, Shen Zhang, Maxwell Lieb, Doug Schmidt, and Jules White. Evaluating Persona Prompting for Question Answering Tasks. In *Security, Privacy and Trust Management*, pages 63–81. Academy & Industry Research Collaboration Center, June 2024. ISBN 978-1-923107-28-1. doi: 10.5121/csit.2024.141106.

[20] P. Organisciak, D. Dumas, S. Acar, and P. L. de Chantal. Open Creativity Scoring [computer software], 2025.

[21] Peter Organisciak, Selcuk Acar, Denis Dumas, and Kelly Berthiaume. Beyond semantic distance: Automated scoring of divergent thinking greatly improves with large language models. *Thinking Skills and Creativity*, 49:101356, September 2023. ISSN 1871-1871. doi: 10.1016/j.tsc.2023.101356.

[22] Andrei Paleyes, Radzim Sendyka, Diana Robinson, Christian Cabrera, and Neil D. Lawrence. Prompt Variability Effects On LLM Code Generation, June 2025. doi: 10.48550/arXiv.2506.10204.

[23] Max Peeperkorn, Tom Kouwenhoven, Dan Brown, and Anna Jordanous. Is Temperature the Creativity Parameter of Large Language Models? In *Proceedings of the 15th International Conference on Computational Creativity (ICCC)*, 2024. doi: 10.48550/arXiv.2405.00492.

[24] Laria Reynolds and Kyle McDonell. Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI EA '21, pages 1–7, New York, NY, USA, May 2021. Association for Computing Machinery. ISBN 978-1-4503-8095-9. doi: 10.1145/3411763.3451760.

[25] Abel Salinas and Fred Morstatter. The Butterfly Effect of Altering Prompts: How Small Changes and Jailbreaks Affect Large Language Model Performance, April 2024. doi: 10.48550/arXiv.2401.03729.

[26] Shayle R. Searle. *Linear Models*. John Wiley & Sons, March 1997. ISBN 978-0-471-18499-7.

[27] Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. Large Language Models Can Be Easily Distracted by Irrelevant Context. In *Proceedings of the 40th International Conference on Machine Learning*, pages 31210–31227. PMLR, July 2023.

[28] Paul J. Silvia, Beate P. Winterstein, John T. Willse, Christopher M. Barona, Joshua T. Cram, Karl I. Hess, Jenna L. Martinez, and Crystal A. Richard. Assessing creativity with divergent thinking tasks: Exploring the reliability and validity of new subjective scoring methods. *Psychology of Aesthetics, Creativity, and the Arts*, 2(2):68–85, 2008.

[29] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM Test-Time Compute Optimally Can be More Effective than Scaling Parameters for Reasoning. *International Conference on Learning Representations*, 2025:10131–10165, May 2025.

[30] Haoyu Wang, Tao Li, Zhiwei Deng, Dan Roth, and Yang Li. Devil's Advocate: Anticipatory Reflection for LLM Agents, June 2024. doi: 10.48550/arXiv.2405.16334.

[31] Peter West and Christopher Potts. Base Models Beat Aligned Models at Randomness and Creativity. In *Proceedings of the Conference on Language Modeling (COLM)*, 2025. doi: 10.48550/arXiv.2505.00047.

[32] Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C. Schmidt. A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT, February 2023. doi: 10.48550/arXiv.2302.11382.

[33] Hanzhang Zhou, Junlang Qian, Zijian Feng, Lu Hui, Zixiao Zhu, and Kezhi Mao. LLMs Learn Task Heuristics from Demonstrations: A Heuristic-Driven Prompting Strategy for Document-Level Event Argument Extraction. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11972–11990, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.647.

[34] Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Yue Zhang, Neil Gong, and Xing Xie. PromptRobust: Towards Evaluating the Robustness of Large Language Models on Adversarial Prompts. In *Proceedings of the 1st ACM Workshop on Large AI Systems and Models with Privacy and Safety Analysis*, LAMPS '24, pages 57–68, New York, NY, USA, November 2024. Association for Computing Machinery. ISBN 979-8-4007-1209-8. doi: 10.1145/3689217.3690621.

A Appendix

A.1 Model version identifiers

Table 1: Model version identifiers and access configuration

Model	Model size	Access	Reasoning
Claude Sonnet 4.5	n/a	Anthropic API	Yes
DeepSeek V3.2	671B total, 37B active	DeepSeek API	Yes
Gemini 3.1 Pro	n/a	Google AI API	Yes
GPT 5.1	n/a	OpenAI API	Yes
GPT 5.2	n/a	OpenAI API	Yes
GPT OSS 120B	117B total, 5.1B active	Ollama (local)	Yes
Grok 4.1	n/a	xAI API	No
Llama 3.3 70B GGUF	70.6B	Ollama (local)	No
Gemma 3 27B	27B	Ollama (local)	No
Mistral Nemo Instruct 2407 GGUF	12B	Ollama (local)	No

Note: All proprietary models were accessed via their respective provider APIs with default parameters (no manual overrides to temperature, top-p, or maximum output length). Small open-weight models were served via Ollama v0.21.0 on a partitioned NVIDIA H100 NVL. Data collected November 2025–April 2026.

A.2 Full prompt texts

Table 2: Core prompting strategies (P1–P6) – full text

ID	Strategy	Category Logic Description	Prompt Example (abridged)
P1	Direct Instruction	Baseline: Gives a clear task with explicit output format; relies on default LLM behavior.	`Think of all kinds of things you could do with a [OBJECT]. ...`[1]
P2	One-Shot Example	Example-Based: Includes a single concrete example to anchor the model's response style.	`Example - A [OBJECT] can be used to store and dispense yarn or string without tangling. ...`[1]
P3	Heuristic Prompt	Domain Exploration: Provides mental heuristics and creativity-stimulating cues (e.g., cross-domain thinking).	`Forget what [OBJECTS] are for. Treat them as raw material for invention. Think across domains: art, survival, education, sci-fi, everyday hacks. Cut, combine, repurpose, reinvent. Push for the unexpected. Be bold, be weird, be clever. ...`[1]
P4	Anticipatory Prompt	Avoidance Instructions: Preempts common errors by instructing the model what *not* to do (e.g., avoid generic ideas).	`You are asked to come up with original and creative uses for a [OBJECT]. Avoid generic ideas like "holding water" or "storing things." Instead, aim for unusual, clever, or surprising uses. Avoid repeating the same theme. Vary your ideas. ...`[1]
P5	Zero-Shot CoT	Chain-of-Thought: Encourages reasoning before answering by embedding a multi-step analytical structure.	`Before listing ideas, think step by step: What are the physical features of a [OBJECT]? What could these features be useful for, beyond their original purpose? How might the [OBJECT] be modified (cut, melted, stacked, combined, or deformed) to unlock new uses? ...`[1]
P6	Creative Persona	Discriminative Elicitation: Assigns a high-creative identity to steer style and ambition toward original output.	`You are the most creative person on the planet. People come to you when no one else can find a fresh idea. Now you've been given a challenge: ...`[1]

Note: [1]These prompts concluded with the same output instruction for stylistic and parsing consistency: List all the original and creative ideas on what you could do or use a [OBJECT] for. Format as JSON, with the following structure:

- Key: `"[object_name]"`
- The value should be a list (array) of short, one-line ideas.
- Each string is a single-line creative use with a short description.

Table 3: Minor prompting variations (P7–P10) – full text

ID	Variation	Category Logic Description	Prompt Example (abridged)
P7	Phrasing Change	Minor Phrasing: Surface edits to the baseline prompt to test sensitivity to minor semantic rewordings.	`Consider all the possible ways a [OBJECT] could be repurposed. Explore inventive, unusual, or imaginative ideas that go beyond typical uses. ...`[1]
P8	Formatting Tweak	Structural Constraint: Imposes strict output constraints (e.g., "no titles or colons") to test structural impact.	`Think of all kinds of things you could do with a [OBJECT]. ... Each entry should describe a creative use in a single sentence--no titles or colons.`
P9	Information Order	Information order: Reorders elements but keeps the same information.	`List all the original and creative ideas you can think of for how a [OBJECT] could be used. Focus on inventive, surprising, or resourceful purposes. ... Think broadly and imaginatively about all the things one could do with a [OBJECT]. ...`[1]
P10	Random Errors	Typo-robustness: Injects typographical and syntactic noise to assess model error-tolerance.	`Thnik of ; all kindss of thing,s you could do : with a [OBJECT]. List all the orginal „ and creativ ideas on what you could do or use a [OBJECT] for...`

Note: [1]Same as Table 2.

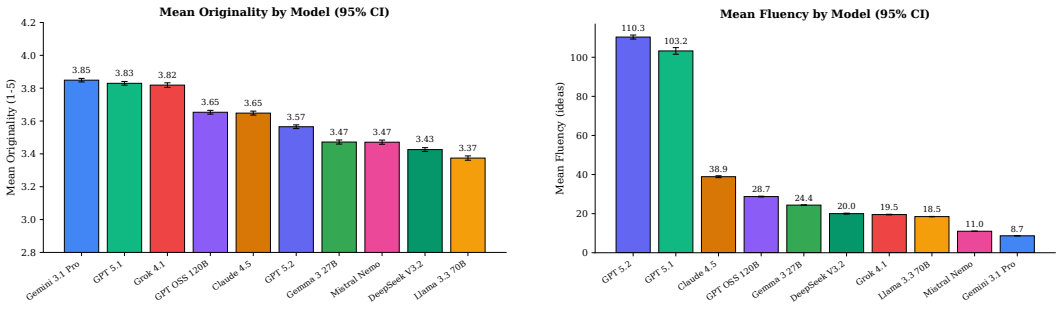
A.3 Originality and fluency distributions

Table 4: Mean originality scores by LLM and prompt (P1–P6), aggregated across 30 AUT items ($N = 89{,}806$)

	Model	P1 (Baseline)	P2 (Example)	P3 (Heuristic)	P4 (Anticipatory)	P5 (CoT)	P6 (Persona)	Mean
1	Claude Sonnet 4.5	3.29	3.34	4.02	3.65	3.40	4.19	3.65
2	DeepSeek V3.2	3.21	3.14	3.95	3.38	3.16	3.72	3.43
3	Gemini 3.1 Pro	3.56	3.49	4.40	3.87	3.56	4.22	3.85
4	Gemma 3 27B	3.32	3.03	3.96	3.53	3.02	3.97	3.47
5	GPT 5.1	3.48	3.50	4.24	3.98	3.44	4.33	3.83
6	GPT 5.2	3.28	3.29	3.96	3.76	3.27	3.83	3.57
7	GPT OSS 120B	3.32	3.31	3.97	3.90	3.37	4.05	3.65
8	Grok 4.1	3.73	3.28	4.43	3.85	3.12	4.51	3.82
9	Llama 3.3 70B	3.09	3.01	3.78	3.45	3.05	3.86	3.37
10	Mistral Nemo	3.20	3.30	3.91	3.52	3.04	3.83	3.47
	Prompt Mean	3.35	3.27	4.06	3.69	3.24	4.05	

Table 5: Mean fluency (number of ideas) by LLM and prompt (P1–P6), aggregated across 30 AUT items ($N = 89{,}806$)

	Model	P1 (Baseline)	P2 (Example)	P3 (Heuristic)	P4 (Anticipatory)	P5 (CoT)	P6 (Persona)	Mean
1	Claude Sonnet 4.5	36.2	41.0	49.6	21.2	70.2	15.1	38.9
2	DeepSeek V3.2	20.6	18.8	21.6	15.6	30.9	12.7	20.0
3	Gemini 3.1 Pro	9.9	9.5	9.8	7.3	8.8	6.8	8.7
4	Gemma 3 27B	23.7	22.5	30.9	20.1	33.4	15.7	24.4
5	GPT 5.1	142.7	114.4	112.9	64.3	138.4	46.7	103.2
6	GPT 5.2	134.7	123.5	148.6	81.8	133.3	39.9	110.3
7	GPT OSS 120B	35.8	30.9	34.0	21.9	36.5	13.4	28.7
8	Grok 4.1	20.0	22.4	19.8	15.4	29.6	10.1	19.5
9	Llama 3.3 70B	19.8	18.8	23.4	16.9	20.5	11.6	18.5
10	Mistral Nemo	11.5	9.8	12.2	10.6	13.1	9.0	11.0
	Prompt Mean	45.5	41.1	46.3	27.5	51.5	18.1	



(a) Mean originality by model (95% CI).

(b) Mean fluency by model (95% CI).

Figure 4: Model-level performance summaries across 30 AUT items (P1–P6).

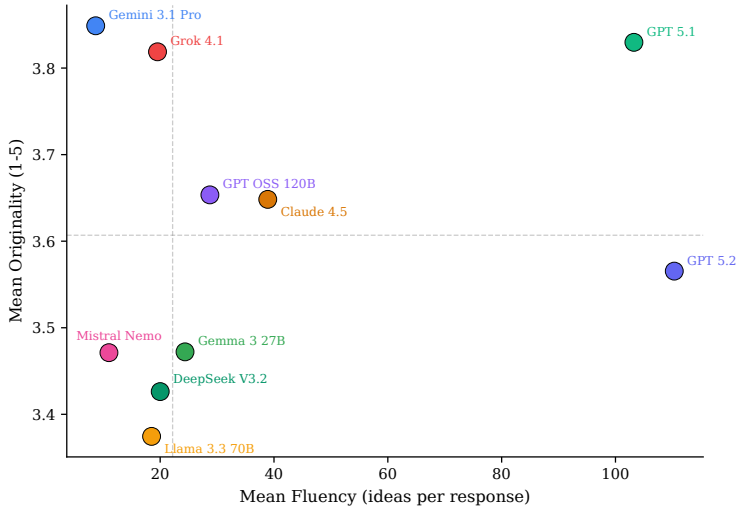


Figure 5: Quality vs. quantity across 10 models (30 AUT items, P1–P6). Each point is one model's mean originality (y) and fluency (x).

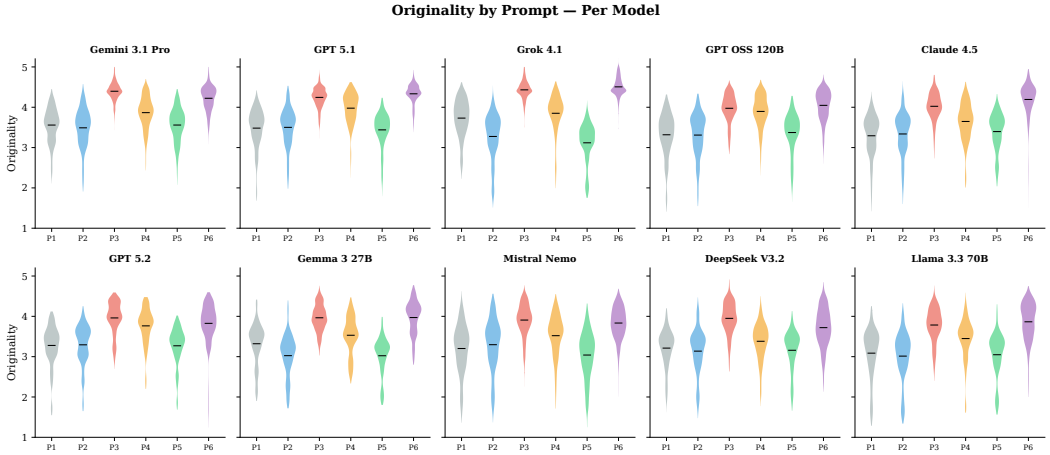


Figure 6: Violin plots displaying originality scores as a function of prompt separately for each model (30 AUT items). Prompt key as in Figure 2.

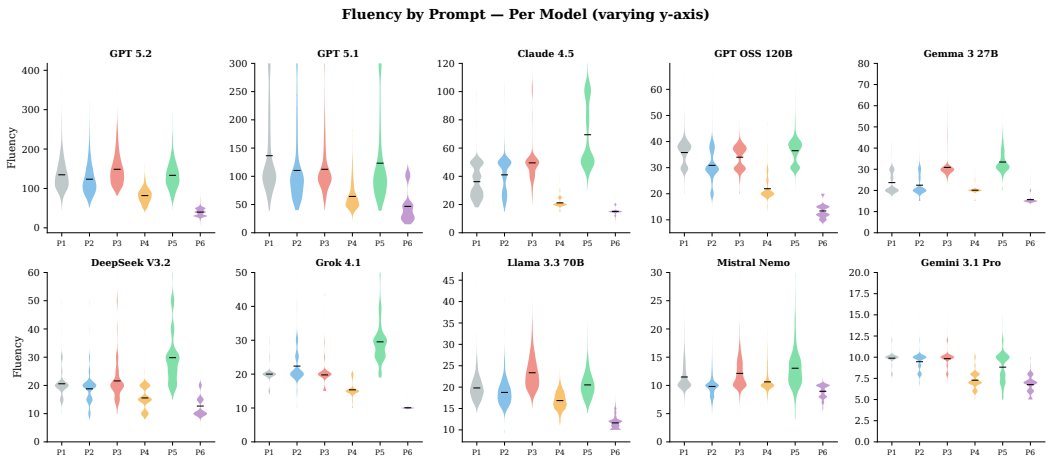


Figure 7: Violin plots displaying fluency (number of ideas) as a function of prompt separately for each model (varying y-axis range to show distribution detail; 30 AUT items). Prompt key as in Figure 2.

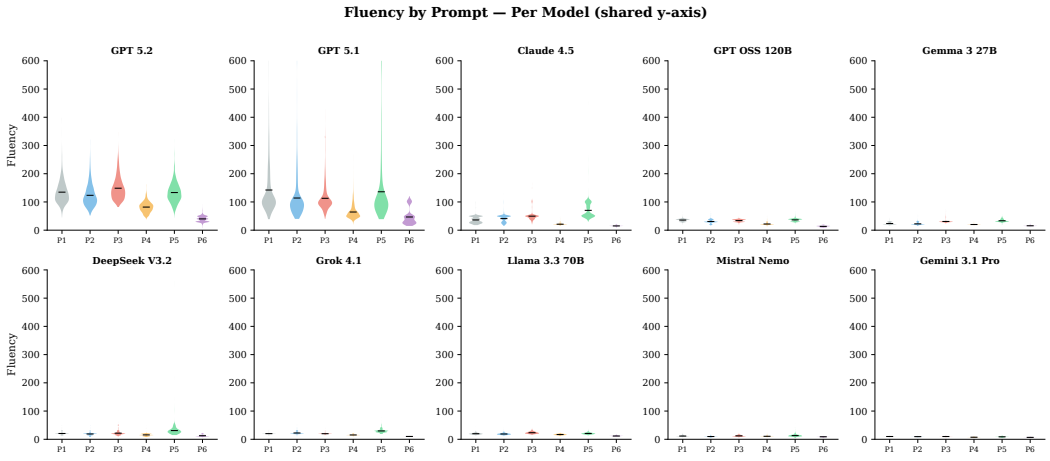


Figure 8: Violin plots displaying fluency (number of ideas) as a function of prompt separately for each model (shared y-axis to compare scales across models; 30 AUT items). Prompt key as in Figure 2.

A.4 Specific prompt analyses

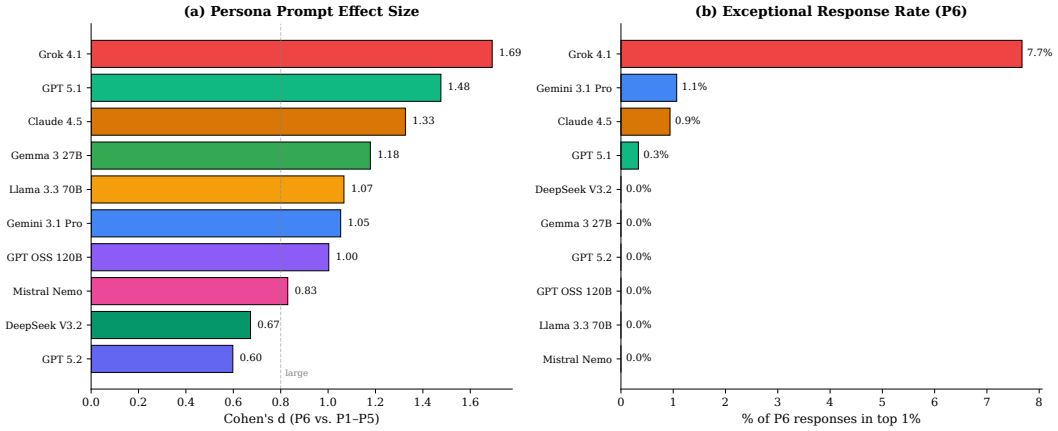


Figure 9: Distribution of exceptional responses (top 1% originality) and performance on Prompt 6 across 30 AUT items. (a) Cohen's d effect sizes for the Persona prompt (P6) relative to all other prompts. (b) Percentage of P6 responses in the top 1% of originality scores.

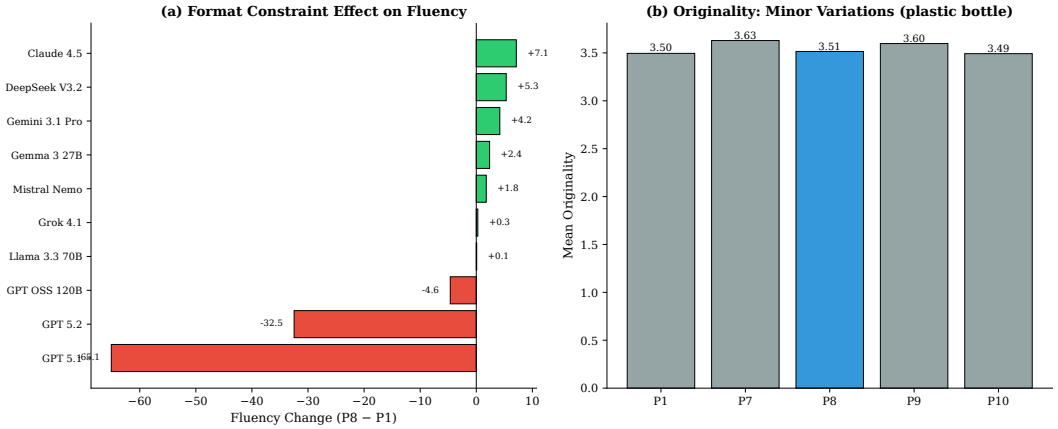


Figure 10: The "structural constraint" effect (plastic bottle; 10 models, $N = 50$/cell). (a) Fluency change from baseline (P1) to format constraint (P8) per model. (b) Mean originality across minor prompt variations (P1, P7–P10), showing P8 does not degrade quality despite reducing output volume.

A.5 Extended variance decomposition

Table 6 presents a 7-factor decomposition that further partitions the within-LLM residual from the primary 5-factor model. Two additional two-way interactions—item×model and item×prompt—are extracted, revealing that the originality residual (16.64%) comprises item×prompt interactions (4.87%), item×model interactions (2.30%), and pure within-cell stochasticity (9.48%). For fluency, the decomposition is more straightforward: the residual (29.84%) contains only modest item×model (1.83%) and item×prompt (0.64%) interactions, with pure stochasticity at 27.37%.

Table 6: Extended 7-factor variance decomposition ($N = 89{,}806$; 50 runs/cell)

Component	Originality % Var.	Fluency % Var.
AUT Item	40.46	0.61
Prompt	32.55	6.14
Model	7.58	55.14
Model × Prompt	2.77	8.28
Item × Model	2.30	1.83
Item × Prompt	4.87	0.64
Pure Residual	9.48	27.37
Within-LLM (5-factor)	*16.64*	*29.84*

Note: The last row (italicized) shows the 5-factor residual for comparison; it equals the sum of the three rows above it. The item×prompt interaction (4.87% for originality) indicates that certain prompts are more effective for certain objects, e.g., the Persona prompt may boost originality more for open-ended items (backpack, shoe) than for functionally constrained items (knife, rope). All runs capped at 50 per cell to ensure balanced design. Bootstrap 95% CIs are < 0.5 pp for all components.

A.6 AUT items

The 30 AUT items used in this study, ordered by mean originality (P1–P6): backpack (4.17), shoe (4.17), chair (4.04), table (3.99), ladder (3.95), tire (3.95), pants (3.94), book (3.90), hat (3.87), shovel (3.85), lightbulb (3.82), ball (3.76), sponge (3.74), sock (3.71), pencil (3.68), plastic bottle (3.66), candle (3.66), fork (3.61), towel (3.56), plate (3.56), bucket (3.50), toothbrush (3.49), spoon (3.46), blanket (3.43), tile (3.35), paperclip (3.18), box (2.98), brick (2.90), rope (2.90), knife (2.53). The item-level range of 1.64 points (backpack 4.17 vs. knife 2.53) underscores why task content accounts for 40.46% of originality variance.

A.7 Minor prompt variations (P7–P10)

To assess sensitivity to surface-level prompt changes, we evaluated four minor prompt variations (P7–P10) on a single item (plastic bottle) with all 10 models and $N = 50$ runs per cell ($N = 2{,}493$ total). We fix the item because the main analysis shows item choice dominates ∼40% of originality variance; using a single item isolates the prompt effect without item-level confounds while retaining all 10 models in a balanced design. Table 7 summarizes the results.

Table 7: Minor prompt variations vs. baseline (plastic bottle; 10 models, $N = 50$/cell)

Prompt	Variation	Mean Orig.	Δ vs. P1	Cohen's d	Interpretation
P1	Baseline	3.50	—	—	Reference
P7	Paraphrase	3.63	+0.13	0.63	Moderate positive
P8	Format constraint	3.51	+0.02	0.10	Negligible
P9	Information reorder	3.60	+0.10	0.53	Small positive
P10	Typo injection	3.49	−0.00	−0.02	Negligible

Note: Cohen's d computed vs. P1 pooled SD. P7 and P9 use slightly more creativity-signaling language (e.g., "repurposed," "inventive") than the neutral P1 baseline, which may explain the positive effects. P8 reduces fluency substantially for high-output models (GPT 5.1: −65.1 ideas, GPT 5.2: −32.5 ideas) without meaningfully affecting originality. P10 (typos) has negligible effects on both metrics, suggesting robustness to surface-level noise.

P8 Format Constraint: Fluency Impact. The format constraint (P8: "no titles or colons") differentially affects high-fluency models. GPT 5.1 drops from 190.0 to 124.9 ideas per response ($\Delta = -65.1$), and GPT 5.2 drops from 177.4 to 145.0 ($\Delta = -32.5$). In contrast, models with lower baseline fluency (Gemini 3.1 Pro, Grok 4.1, Mistral Nemo) show negligible fluency changes, suggesting that format constraints primarily affect models that rely on structured output templates for volume generation (see Figure 10).

A.8 Sample-size convergence analysis

To verify that $N = 50$ runs per cell provide stable variance estimates, we performed a subsampling analysis on plastic bottle—the one item for which all 10 models completed ≥ 96 runs per cell (P1–P6). For each subsample size $n \in \{5, 10, \ldots, 90\}$, we drew n runs without replacement from each model×prompt cell, recomputed the variance decomposition (prompt, model, model×prompt, within-LLM), and repeated this procedure 5,000 times. Figure 11 shows that point estimates are stable from approximately $N = 20$ onward; 95% bootstrap CIs at $N = 50$ are within ± 1.5 pp of the full-sample estimate for all components. This confirms that $N = 50$ runs per cell, as used in our primary design, provide sufficient precision for the variance proportions reported in the main text.

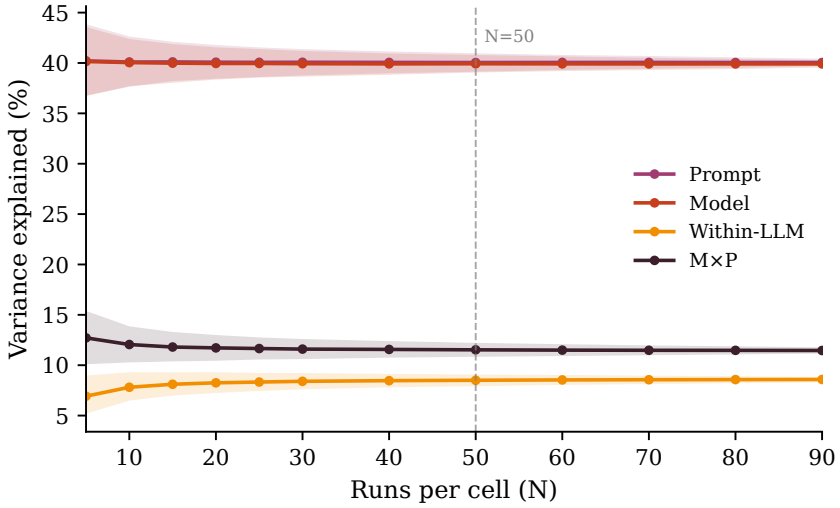


Figure 11: Convergence of variance decomposition estimates as a function of runs per cell (plastic bottle, 10 models, P1–P6). Shaded bands show 95% bootstrap CIs (5,000 iterations). Point estimates stabilize by $N \approx 20$; CIs are narrow by $N = 50$ (dashed line).

A.9 Confounded variance components

Our "model" factor bundles architecture, training data, RLHF/alignment, default parameters, system prompts, and serving infrastructure. We retain "model" as it reflects common usage, although variance cannot be attributed to architecture alone. Our primary "within-LLM variance" (16.64% for originality) is technically the residual from a 5-factor model and includes unmodeled item×prompt and item×model interactions. An extended 7-factor decomposition (Table 6) shows that pure within-cell stochasticity is 9.48%, with item×prompt (4.87%) and item×model (2.30%) interactions accounting for the remainder. Even this reduced residual still exceeds model choice (7.58%). Our item-conditioned analysis (Figure 1b) partially addresses this confound by removing item variance from the decomposition; the resulting within-LLM residual (11.9%) is closer to the pure stochasticity estimate. Disentangling architectural effects requires controlled experiments with matched parameters, which is left for future work.

A.10 Semantic validity audit

OCSAI rewards semantic distance from common responses and does not itself enforce physical feasibility. To bound this concern we audited the plastic-bottle item, which is also the item used

for the P7–P10 and convergence analyses. We drew a stratified sample of 990 ideas (11 model configurations $\times$ 10 prompts $\times$ 9 ideas per cell). Each idea was rated on a three-point scale (1 = nonsense, 2 = unusual but possible, 3 = feasible) by an LLM judge (Claude) and an independent rule-based classifier. Disagreements were adjudicated item by item, defaulting to the stricter label. Two further LLM raters (GPT 5.5 and Gemini 3.6 Flash) rated the same sample as a sensitivity check. Table 8 reports the resulting prevalence.

Table 8: Semantic validity audit of 990 plastic-bottle ideas. Percentages in parentheses.

Rater	Nonsense	Edge case	Feasible
Adjudicated (final)	5 (0.51%)	71 (7.17%)	914 (92.32%)
Claude	2 (0.20%)	24 (2.42%)	964 (97.38%)
GPT 5.5	9 (0.91%)	155 (15.66%)	826 (83.43%)
Gemini 3.6 Flash	0 (0.00%)	1 (0.10%)	989 (99.90%)

Note: Mean OCSAI originality: nonsense 4.18, edge 3.92, feasible 3.62. Pairwise Cohen's κ among the three LLM raters ranged from -0.002 to 0.18.

Three observations follow. First, nonsense is rare (0.51%), but edge cases are not (7.17%). Second, both are concentrated in speculative models (Gemini, Grok) and ambitious prompts (P3, P6). Third, flagged ideas score *higher* on OCSAI than feasible ideas, so removing them would lower means. At the adjudicated prevalence, the estimated grand-mean bias is at most 0.024 points ($5/990 \times 0.56 + 71/990 \times 0.30$). Rater agreement was low, so we treat the audit as a robustness check, not as independent human validation.

Cell-wise sensitivity. Because flagged ideas are concentrated in specific models and prompts, a grand-mean bias bound is not sufficient. Restricting to the nine audited models that are part of the main design and P1–P6 (486 ideas), 2 ideas are nonsense (0.41%) and 42 edge (8.64%). We estimated each cell's bias as $\pi_c \times (\bar{s}_{\text{flagged}} - \bar{s}_{\text{feasible}})$ and subtracted it from the cell mean. The plastic-bottle decomposition (nine audited models, P1–P6, 50 runs/cell) changes by at most 1.7 pp: prompt $49.05 \rightarrow 47.37\%$, model $27.04 \rightarrow 27.46\%$, interaction $13.86 \rightarrow 14.99\%$, residual $10.05 \rightarrow 10.18\%$. The ordering is unchanged.

A.11 Seed-controlled reproducibility

For the four locally hosted models (Llama 3.3, Mistral Nemo, Gemma 3, GPT OSS) we repeated the plastic-bottle P1 condition with a fixed seed (Ollama v0.21.0, GPU inference). All 50 runs per model produced byte-identical outputs, and OCSAI assigned identical scores ($\sigma^2 = 0$ across 200 responses), confirming that the local serving and scoring pipeline is deterministic when the seed is fixed. A paraphrase set with the same semantic content but slightly reworded text yielded $\sigma^2 = 0.00087$ (SD ≈ 0.03) at the response-mean level, whereas the random-seed condition yielded $\sigma^2 = 0.00593$, more than $6\times$ the paraphrase sensitivity. The within-LLM variance in the main experiment therefore reflects genuine differences in generated content, not scoring artifacts or serving nondeterminism. For API-based models we cannot control seeds, so additional serving-layer variance cannot be excluded. The overlap in within-cell variance between open- and closed-weight models (Table 9) suggests it is not the dominant driver.

A.12 Open- vs. closed-weight models

Table 9 separates the models by access mode. The mean within-cell variance of response-level originality is 0.031 for open-weight and 0.019 for closed-weight models. The range within each category is larger than the between-category difference (0.008–0.062 for open-weight, 0.007–0.036 for closed-weight), and the most and least variable models are Mistral Nemo (open) and GPT 5.2 (closed), respectively. This argues against a simple "API equals more variance" explanation.

A.13 Qualitative examples and sampling precision

Table 10 shows low-, median-, and high-originality responses from the same condition (Grok 4.1, P1, plastic bottle). The high-originality response is substantively different (recycling into construction material), not a rewording of the low or median response.

Table 9: Mean within-cell variance of response-level originality by access mode (P1–P6, 180 item×prompt cells per model).

Open-weight		Var.	Closed-weight		Var.
GPT OSS 120B	local	0.023	Claude Sonnet 4.5	API	0.018
Gemma 3 27B	local	0.008	DeepSeek V3.2	API	0.036
Mistral Nemo 12B	local	0.062	Gemini 3.1 Pro	API	0.025
Llama 3.3 70B	local	0.031	GPT 5.1	API	0.014
			GPT 5.2	API	0.007
			Grok 4.1	API	0.016
Pooled		0.031	Pooled		0.019

Table 10: Qualitative examples from one condition (Grok 4.1, P1, plastic bottle).

Originality	Generated idea
2.70 (low)	Perfume sprayer: Clean and fill with homemade scents, mist from the cap.
3.50 (median)	Self-watering planter: Poke holes in cap, fill with water, bury neck-down in soil for slow drip irrigation.
4.50 (high)	Crush and layer in concrete to form lightweight, textured garden stepping stones.

Sampling precision. For a model×prompt condition averaged over the 30 items, the number of runs needed to estimate the condition mean to ± 0.1 points at 95% confidence (normal approximation, $N = (1.96\sigma/0.1)^2$) ranges from 18 (Gemini 3.1 Pro, P3) to 159 (Mistral Nemo, P2), with a median of 84 across the 60 conditions. These requirements are condition-specific, so a pilot estimate of σ is more useful than a fixed N.

Exact-duplicate rates. Across all P1–P6 plastic-bottle ideas (282,168), 93.5% are text-unique. Per-model distinct-idea rates range from 43.8% (Gemma 3 27B) to 99.8% (Gemini 3.1 Pro, GPT 5.1). Llama 3.3 (81.0%), Grok 4.1 (84.1%), and Claude Sonnet 4.5 (94.2%) fall in between. This is a strict lower bound on semantic overlap because synonyms and paraphrases are not counted. Embedding-based clustering into semantic "idea modes" is left for future work.

A.14 Numerical illustration of the item-conditioned decomposition

Table 11 isolates the mechanism behind Figure 1a→b on a small real subset: two items (plastic bottle, knife), two models (Claude Sonnet 4.5, Grok 4.1), two prompts (P1, P6), and 50 runs per cell ($N = 391$). Deleting the item sum of squares and renormalizing leaves the residual above the model component (27.49% vs. 22.14%). Refitting within each item and averaging the per-item shares absorbs the item×model and item×prompt interactions into the model and prompt components, yielding model 26.22% and residual 4.66%. The two mechanisms are therefore (i) absorption of interactions into the within-item components and (ii) averaging per-item ratios rather than taking a ratio of pooled sums.

Table 11: Numerical illustration of the item-conditioned decomposition (2 items × 2 models × 2 prompts × 50 runs, $N = 391$).

Component	Unconditional	Item SS deleted, renormalized	Per-item fits, averaged
Item	34.44	n/a	n/a
Prompt	30.90	47.13	64.61
Model	14.51	22.14	26.22
Model×Prompt	2.12	3.24	4.50
Residual	18.02	27.49	4.66

A.15 Schematic of the variance-partitioning procedure

Figure 12 summarizes the mapping from the experimental design to the decomposition model and the resulting variance shares.

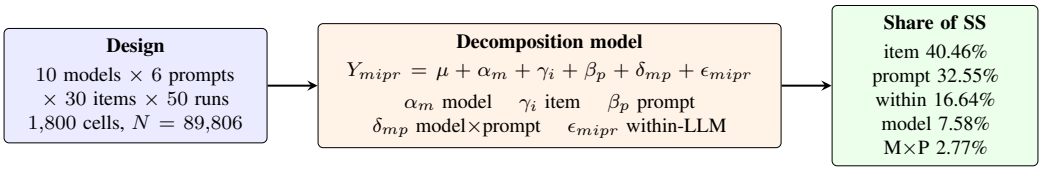


Figure 12: Schematic of the variance-partitioning procedure: the fully crossed design maps onto the terms of Eq. 1, whose sum-of-squares shares are reported in Figure 1.

A.16 Data and code availability

The response-level dataset (89,806 scored responses), prompt templates, analysis scripts (variance decomposition and bootstrap confidence intervals), figure-generation code, semantic-validity audit labels, and fixed-seed replication data are available at `https://github.com/JenniferHaase/neurips-aut-variance`. The repository includes a README, a data dictionary, and instructions for reproducing the main decomposition and the appendix analyses. Raw idea-level text and provider logs are available from the authors on request.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract states three claims (item dominance, stochasticity $>$ model differences, distributional prompt effects) with quantitative bounds; the introduction (Section 1) repeats them with exact percentages. Scope is explicitly limited to creative divergent-thinking tasks (AUT) and does not claim generalization beyond open-ended generation.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The "Limitations" paragraphs in Section 5 discuss six specific limitations: (1) task scope (single creativity measure, no fluency-targeted prompts), (2) unmeasured creativity dimensions (flexibility, elaboration), (3) semantic validity (audit restricted to one item and LLM raters, Appendix A.10), (4) unclustered semantic diversity, (5) confounded variance components ("model" bundles architecture, alignment, and defaults, detailed in Appendix A.9), and (6) the linear sum-of-squares decomposition as an approximation for count data (fluency).

   Guidelines:

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

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

   Justification: This is an empirical study; no theoretical results or proofs are presented.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Section 3 specifies all 10 model identifiers and providers (Table 1), full prompt texts (Appendix A.2), hyperparameter choices (provider defaults; Section 3.2), sample sizes (N=50 runs/cell), and the publicly available scoring system (Open Creativity Scoring; Section 3.4).

   Guidelines:

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

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: The full dataset (89,806 scored responses as CSV with model, prompt, item, run, originality, and fluency columns), analysis scripts (variance decomposition, bootstrap CIs), figure generation code, semantic-validity audit data, and seed-controlled replication data are available at `https://github.com/JenniferHaase/neurips-aut-variance` (Appendix A.16).

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: No model training was performed. Section 3.2 reports that all models used provider-default hyperparameters (temperature, top-p, max tokens) with the rationale for this choice. Section 3.3 specifies the fully crossed design (10 models $\times$ 6 prompts $\times$ 30 items $\times$ 50 runs) and data-filtering pipeline.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We report 95% bootstrap confidence intervals (5,000 iterations, percentile method) for all variance proportions; these capture sampling variability of the variance decomposition under resampling within cells, assuming exchangeability of runs within each model×prompt×item cell (Section 4.1, Table 6). F-tests with degrees of freedom and p-values are reported for all main effects and interactions.

   Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Section 3.3 reports hardware (NVIDIA H100 NVL MIG 7g 94GB for local models via Ollama v0.21.0; official APIs for proprietary models), total wall-clock time (∼15 days; per-run average 36h, range 20h–3d), and estimated API cost (∼$750 across OpenAI, Anthropic, Google, xAI, DeepSeek). Table 1 lists all model identifiers and access methods.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

Answer: [Yes]

Justification: The study evaluates publicly available LLMs on a standard creativity task (AUT). No human subjects, private or sensitive data, dual-use capabilities, or harmful applications are involved; all API usage complied with provider terms of service (Sections 3.3 and 5).

Guidelines:

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

10. **Broader impacts**

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

Answer: [Yes]

Justification: Positive impact: the finding that single-sample evaluations are unreliable (within-LLM variance $>$ model differences) can improve benchmarking practices across the field. Negative impact: we do not foresee direct negative societal consequences; the work is methodological and does not introduce new capabilities, datasets, or deployment recommendations that could be misused (Section 5).

Guidelines:

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

11. **Safeguards**

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

Answer: [N/A]

Justification: No pretrained models are released. The dataset consists solely of creative-text outputs (AUT responses) from commercially available LLMs, which poses no misuse risk beyond what the models themselves already provide (Appendix A.16).

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All 10 LLMs are credited with provider names (Table 1) and accessed under their respective API terms of service. The Open Creativity Scoring system (OCSAI, 2025 release; `https://openscoring.du.edu/`) is cited [20] and used under its MIT license. Our released code is MIT-licensed and the released data are CC BY 4.0 (Section 3.4, Appendix A.16). No license restrictions were violated.

    Guidelines:

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

13. **New assets**

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

    Answer: [Yes]

    Justification: The dataset (89,806 scored responses) includes per-response metadata (model, prompt, item, run, originality, fluency). The README and data dictionary document collection methodology, intended uses, and limitations. Code and data are publicly available (Appendix A.16) under an open license.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: No human subjects or crowdsourcing were involved.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: No human subjects were involved in this study.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: LLMs are the primary object of study, not an auxiliary tool. All 10 models are listed by name, provider, size, and access method in Section 3.3 and Table 1. LLM-generated outputs are evaluated. LLMs further supported the generation of the Figures in the paper.

    Guidelines:

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