WHERE YOU INJECT DIVERSITY MATTERS: A UNIFIED FRAMEWORK FOR DIVERSE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Open-ended generation tasks often require a set of meaningfully different outputs, yet large language models often produce similar generations. Existing test-time diversity methods operate at different stages of generation with varying effectiveness, but it remains unclear what design choices lead to meaningful diversity in the output. We introduce a framework that characterizes test-time diverse generation methods by the diversity source introduced during generation and provide a transmission score that quantifies how effectively variation in the source is realized in the final output. Guided by this framework, we instantiate two fully automated methods that first generate diverse intermediate specifications and then condition on them to produce final responses. Across five open-ended tasks and four backbone models, specification-level injection improves output diversity over test-time baselines while maintaining comparable quality. Our analysis shows that successful diversity injection depends on both the diversity of the sources and their transmission to the output, highlighting source design and source-to-output realization as two key levers for building more diverse generation systems.

1 INTRODUCTION

Large language models (LLMs) are increasingly used for open-ended generation tasks such as creative writing, brainstorming, argumentation, and ideation, where users often need a set of meaningfully different outputs rather than a single best answer. However, when prompted repeatedly with the same request, LLMs tend to converge on similar high-level patterns and recurring ideas, producing outputs that differ in wording but not in substance (Sorensen et al., 2024). This limits the practical value of generating multiple outputs and risks homogenizing the creative input available to users.

A growing body of test-time methods attempts to address this problem by increasing decoding randomness through temperature scaling or modified sampling strategies (Fan et al., 2018; Holtzman et al., 2020; Minh et al., 2025), introducing per-output random strings or noise tokens during generation (Misaki & Akiba, 2026; Agrawal & Goyal, 2026), or steering decoding with diversity-promoting modules (Ruan et al., 2025). These methods operate at different stages of the generation process and inject different forms of variation, raising a natural question: what kind of diversity source is more effective for producing diverse final outputs?

In this work, we propose a **Diversity Injection Framework**, a unified lens for studying test-time diverse generation. We formulate generation as $y_i = G(x, z_i)$, where x is the input prompt and z_i is an explicit diversity source that varies across outputs. Methods differ in whether z_i exists and what information it carries. We categorize methods into three levels: *no injection* (Level 0), where no per-output z_i exists; *surface-level injection* (Level 1), where z_i carries no semantic content about the output; and *specification-level injection* (Level 2), where z_i provides a detailed generative configuration. We introduce an entropy decomposition of output diversity into *injected diversity* (how diverse the sources z_i are), *information loss* (diversity in z not realized in the output), and *residual randomness* (variation from decoding). From this decomposition we derive a *transmission score* that quantifies what fraction of injected diversity reaches the final output.

Analysis through the framework reveals that existing fully automated diversity methods remain at Level 0 or Level 1, relying entirely on decoding randomness or injecting diversity sources that carry little semantic content. Level 2 injection has not yet been fully explored for promoting diverse

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

generation. This gap motivates us to ask: *can specification-level diversity injection improve output diversity across general open-ended tasks?*

We propose fully automated methods that instantiate Level 2 injection and are applicable to broad open-ended generation tasks. Given a prompt, the model first generates multiple diverse specifications, each describing a distinct generative configuration along dimensions such as perspective, tone, and thematic focus. The model then conditions on each specification to produce a final output. The entire process requires no human intervention and no task-specific engineering: the same pipeline applies to creative writing, argumentation, poetry, and brainstorming alike.

In summary, we make three contributions: (1) a unified framework that characterizes test-time diverse generation methods by their diversity source, and an entropy-based transmission score that quantifies how effectively injected diversity reaches the output; (2) automated, general-purpose specification-level methods that inject diversity through intermediate generative configurations; and (3) empirical evidence across 5 open-ended tasks and 4 backbone models showing that specification-level injection improves output diversity over lower-level test-time baselines while maintaining comparable quality.

2 Related Work

We discuss related work in three areas: test-time methods for diverse generation, planning-based generation, and diversity evaluation.

Decoding-based Diverse Generation. The most straightforward approach to increasing output diversity is to modify the sampling strategy at decoding time. Temperature scaling flattens the output distribution to encourage more varied token selections (Ackley et al., 1985). Top-k sampling (Fan et al., 2018), nucleus sampling (Holtzman et al., 2020), and min-p sampling (Minh et al., 2025) each restrict the candidate token pool using different truncation criteria. Contrastive search (Su & Collier, 2023) penalizes tokens that are too similar to previous context, and diverse beam search (Vijayakumar et al., 2018) encourages diversity across beam groups. Avoidance Decoding (Park et al., 2025) extends this contrastive idea to multi-output generation by treating prior branches as negative samples and penalizing candidate tokens using hidden-state and sentence-embedding similarity. G2 (Ruan et al., 2025) takes a different approach by steering decoding with dual guide modules that encourage novelty and suppress repetition. These methods all operate at the token level and do not introduce explicit per-output conditioning that structures the diversity in generation.

Prompt-based Diverse Generation. Several training-free methods promote diversity through prompt design. Zhang et al. (2024) improve diversity in commonsense generation tasks by diversifying in-context learning examples. Hayati et al. (2024) explore diverse perspective extraction through criteria-based prompting on subjective topics. Verbalized Sampling (Zhang et al., 2026) prompts the model to generate multiple candidate responses with associated probabilities in a single call. SSOT (Misaki & Akiba, 2026) prepends a randomly generated string to each generation as a diversity seed, and Agrawal & Goyal (2026) prepend random, unrelated concepts to prompts and measure the resulting changes in responses. Other approaches explore translation-based instructions, such as Multilingual Prompting, to elicit varied responses, though these generally yield limited diversity compared to our Level-2 specification methods(Wang et al., 2025). Unlike these methods, we make the diversity source explicit for each output before generation.

Planning-based Generation. A separate line of work uses outlines or plans to control text structure, especially in storytelling, though not for the purpose of promoting output diversity. DOC (Yang et al., 2023) uses detailed hierarchical outlines to improve coherence in long-form storytelling. Wang & Kreminski (2024) combine symbolic outline planning with LLM-based surface realization for story generation, building on earlier narrative planning work (Riedl & Young, 2010; Rivera et al., 2024). Luminate (Suh et al., 2024) generates stylistic and content dimensions for human-AI co-creative exploration, but requires interactive human steering. These methods usually target quality of the output. In contrast, we study automatically generated per-output specifications as a general mechanism for promoting diversity across open-ended generation tasks.

Diversity Evaluation Measuring diversity in LLM outputs remains an open challenge. Standard diversity metrics include lexical measures such as Distinct-N (Li et al., 2016) and Self-BLEU (Zhu et al., 2018), and embedding-based distances using pretrained sentence encoders (Reimers & Gurevych,

Table 1: Classification of existing diverse output generation methods by injection level.

Method	z_i	**Level**
Direct Generation	none	0
Verbalized Sampling (Zhang et al., 2026)	none	0
G2 (Ruan et al., 2025)	none	0
SSOT (Misaki & Akiba, 2026)	random string	1
Rand Concept (Agrawal & Goyal, 2026)	random concept	1

2019). Shypula et al. (2025) argue that diversity should be evaluated jointly with quality. Datasets have been developed recently to facilitate evaluation. Zhang et al. (2025) propose NoveltyBench, a benchmark specifically designed for evaluating response diversity. Sorensen et al. (2024) introduce Infinite Chats, a large-scale dataset of open-ended queries and use it to study inter- and intra-model homogeneity in modern LLMs. We use these metrics and benchmarks in our experiments.

3 Diversity Injection Framework

When an LLM generates multiple outputs for the same prompt, the outputs may differ from one another. These differences can come from two sources: randomness in the decoding process, or explicit guidance that steers each generation in a different direction. We formalize this distinction into a unified framework.

3.1 Formulation

We consider open-ended generation tasks for which multiple meaningfully different outputs can all be valid responses to the same input prompt x. Given an input prompt x, a method produces m outputs $\{y_1, \ldots, y_m\}$. We write the generation of each output as: $y_i = G(x, z_i)$ where z_i is a *diversity source*, the information beyond x that the model conditions on when generating y_i. For some methods, z_i is absent; for others, it may be a random string or a set of constraints. We categorize methods into three levels based on what information z_i carries.

Level 0: No Injection. No per-output diversity source exists. Every output is sampled from the same conditional distribution: $y_i \sim P(Y|x)$, and differences between outputs arise entirely from the stochasticity of the decoding process.

Level 1: Surface-Level Injection. Each output is conditioned on a z_i that carries no semantic content about the target output, such as a random seed string, a nonce token, or an arbitrary identifier. The diversity source perturbs the input but does not specify any property of the output.

Level 2: Specification-Level Injection. Each output is conditioned on a z_i that specifies the generative configuration along multiple dimensions, such as narrative perspective, tone, focal subject, stylistic register, and content keywords. The diversity source establishes *how* the model approaches generation and *which elements* it foregrounds, jointly steering style and content.

The three levels form a spectrum of increasing output diversity in the conditioning signal. A method belongs to Level 1 if there exists a per-output z_i that varies across outputs but carries no semantic content. It belongs to Level 2 if z_i specifies meaningful properties of the output along multiple dimensions.

3.2 Classification of Existing Methods

Table 1 classifies existing diverse generation methods by injection level.

Level 0 Methods. *Direct Generation* and its variants (temperature scaling, top-k sampling, nucleus sampling, and min-p sampling) repeatedly sample from the same prompt. They may reshape the token distribution but add no per-output conditioning, so diversity comes solely from sampling randomness. *Verbalized Sampling* (Zhang et al., 2026) elicits multiple candidates in a single call, but issues the

same prompt every time, leaving decoding stochasticity as the sole source of variation. *G2* (Ruan et al., 2025) steers decoding with diversity-promoting guide modules but likewise uses the same prompt for every output.

Level 1 Methods. *SSOT* (Misaki & Akiba, 2026) instructs the model to first generate a random string, then condition on it as a seed to produce the output. The random string serves as a z_i and differs across outputs. However, it carries no semantic content about the desired output's topic or organization. The same holds for Random Concept (Agrawal & Goyal, 2026), which prepends an unrelated concept as the seed.

The Level 2 Gap. To our knowledge, no existing test-time method provides per-output specifications to promote diversity in general open-ended generation. A related line of work on narrative planning conditions generation on outlines or plot structures (Yang et al., 2023; Suh et al., 2024), but these methods use specification mainly to improve the quality of a single generation and are often restricted to storytelling. We address this gap in Section 4 by introducing fully automated Level 2 methods that provide specifications to induce diverse outputs across general open-ended tasks.

3.3 Quantifying Diversity Transmission

The framework above tells us *where* diversity is injected. A natural next question is: how much of the injected diversity actually survives to the final output? A method might generate highly diverse plans (high diversity in z), but if the LLM ignores or flattens those plans during generation, the output diversity will be low. We now formalize a way to measure this.

Let Z and Y be random variables representing the diversity source and the output for a given prompt $X = x$. By the definition of conditional mutual information, the entropy of the output can be decomposed as:

$$H(Y \mid X) = I(Y; Z \mid X) + H(Y \mid X, Z) \tag{1}$$

where $I(Y; Z \mid X)$ is the **realized diversity**, the portion of output diversity that is attributable to variation in Z. In other words, this is the diversity that the injection mechanism successfully transmits to the output. $H(Y \mid X, Z)$ is the **residual diversity**, the variation that arises even when the diversity source z is held fixed, i.e., decoding randomness. The realized diversity can be further expanded as:

$$I(Y; Z \mid X) = H(Z \mid X) - H(Z \mid X, Y) \tag{2}$$

where $H(Z \mid X)$ is the **injected diversity**, measuring how diverse the sources themselves are, and $H(Z \mid X, Y)$ captures **information loss**, measuring the diversity present in Z that is not realized in the output. Based on this decomposition, we define the transmission score as follows.

Definition 1 (Transmission Score)**.** *For methods with an explicit diversity source Z and $H(Z \mid X) > 0$, we define the transmission score as*

$$T = \frac{I(Y; Z \mid X)}{H(Z \mid X)} \tag{3}$$

It measures the fraction of source diversity that is realized in the final output.

Since $I(Y; Z \mid X) \leq H(Z \mid X)$, the transmission score lies in $[0, 1]$. A value of $T = 0$ indicates that the output is conditionally independent of the source given the prompt, while $T = 1$ indicates that the source variation is fully recoverable from the output under the chosen representation.

The transmission score is defined only for methods with an explicit diversity source (Levels 1 and 2). For Level 0 methods, no per-output z_i exists, so the decomposition does not apply, and output diversity arises from decoding randomness. We examine how T and the decomposition terms vary across methods in Section 5.

3.4 Estimating Output Entropy and Transmission Score

Computing the transmission score T and the decomposition in Eq. 1-2 requires estimating $H(Y|X), H(Y|X, Z)$, and $H(Z|X)$. The latter two can be computed directly from the LLM's log probabilities given a fixed z or a fixed prompt. The challenge is estimating $H(Y|X)$, the marginal output entropy, which requires marginalizing over Z.

Algorithm 1 ESTIMATET(X, P, M, L): Estimate the Transmission Score

Require: Task X, model P, estimation size M, evaluation size L
Ensure: Transmission score $\hat{T}$

Sample estimation set $\{Z_j\}_{j=1}^{M} \sim P(Z|X)$
Sample evaluation set $\{Z_\ell\}_{\ell=1}^{L} \sim P(Z|X)$
Generate $Y_\ell \sim P(Y \mid Z_\ell, X)$ for each ℓ
for $\ell = 1, \ldots, L$ **do**
    $\hat{P}(Y_\ell \mid X) \leftarrow \frac{1}{M} \sum_{j=1}^{M} P(Y_\ell \mid Z_j, X)$
    $\hat{h}_\ell \leftarrow -\frac{1}{|Y_\ell|} \log_2 \hat{P}(Y_\ell \mid X)$
end for
$\hat{H}(Y \mid X) \leftarrow \frac{1}{L} \sum_\ell \hat{h}_\ell$
$\hat{H}(Y|X,Z) \leftarrow \frac{1}{L} \sum_\ell \left[-\frac{1}{|Y_\ell|} \log_2 P(Y_\ell|Z_\ell, X) \right]$
$\hat{H}(Z|X) \leftarrow \frac{1}{L} \sum_\ell \left[-\frac{1}{|Z_\ell|} \log_2 P(Z_\ell|X) \right]$
return $\hat{T} = \frac{\hat{H}(Y|X) - \hat{H}(Y|X,Z)}{\hat{H}(Z|X)}$

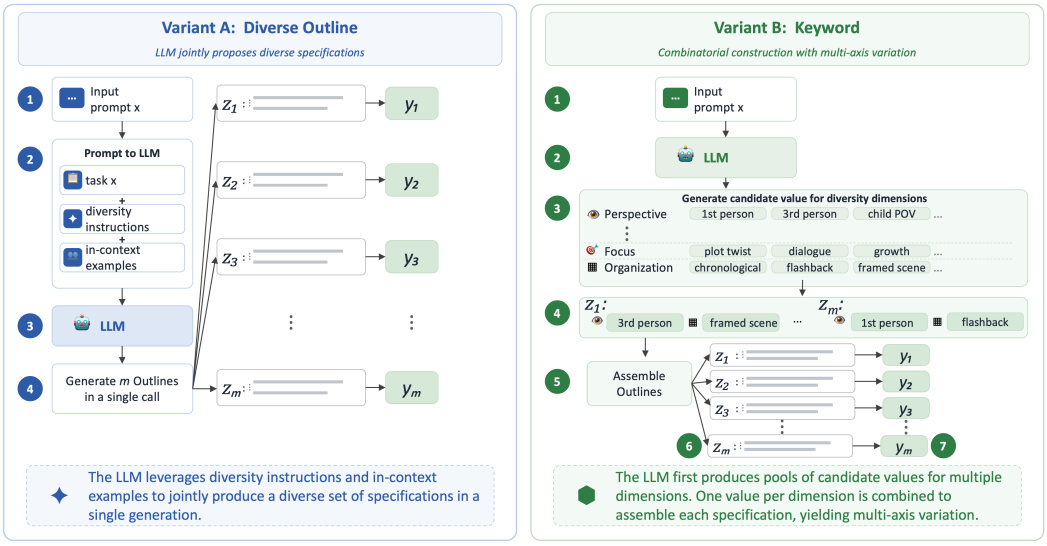


Figure 1: Two variants for generating diverse specifications (z_i's) to encourage LLM outputs. Diverse Outline (left) relies on the LLM's internal capacity to jointly produce a batch of diverse outlines. Keyword (right) explicitly constructs combinatorial outlines by sampling keyword combinations from LLM generated outline contents.

We estimate $H(Y|X)$ using Monte Carlo approximation on a holdout set. For a given task X, we draw two independent sets of outlines from the LLM: an **estimation set** $\{Z_j\}_{j=1}^{M}$ and an **evaluation set** $\{Z_\ell\}_{\ell=1}^{L}$. For each Z_ℓ, we generate one output $Y_\ell \sim P(Y \mid Z_\ell, X)$. We then approximate the marginal likelihood of each evaluation output by averaging over the estimation set: $\hat{P}(Y_\ell \mid X) = \frac{1}{M} \sum_{j=1}^{M} P(Y_\ell \mid Z_j, X)$. The output entropy is estimated by averaging the per-token log-probability across L evaluation members: $\hat{H}(Y \mid X) = \frac{1}{L} \sum_{\ell=1}^{L} \left[-\frac{1}{|Y_\ell|} \log_2 \hat{P}(Y_\ell \mid X) \right]$, where $|Y_\ell|$ is the number of tokens in Y_ℓ. Note that $\hat{P}(Y_\ell|X)$ is an unbiased estimator. Since each Z_j is drawn i.i.d. from $P(Z|X)$, $\mathbb{E}_{\{Z_j\}}[\hat{P}(Y_\ell|X)] = \mathbb{E}_{Z \sim P(Z|X)}[P(Y_\ell|Z,X)] = P(Y_\ell|X)$. However, the entropy estimate $\hat{H}(Y \mid X)$ has a positive bias of $O(1/M)$ due to Jensen's inequality applied to the concave log function, so $\hat{H}(Y|X)$ is a soft upper bound on the true entropy. The bias vanishes as M increases. Full derivations are provided in Appendix D. Given estimates of $H(Y|X)$, $H(Y|X,Z)$, and $H(Z|X)$, we can calculate the transmission score. Algorithm 1 summarizes the full procedure.

4 LEVEL-2 INSTANTIATIONS

We propose two fully automated methods that instantiate Level 2 injection for general open-ended generation tasks. Both methods follow a two-step pipeline. Given an input prompt x, we first generate a set of per-output specifications $\{z_1, \ldots, z_m\}$, and then condition on each specification to produce a final output $y_i = G(x, z_i)$. Each specification is represented as a short outline describing the intended generative configuration, including dimensions such as perspective, tone, thematic focus, and content organization. The two methods differ in how these specifications are constructed. Prompts are provided in Appendix C

Diverse Outline. Given task x, we prompt the LLM to generate m specifications in a single call, where each specification is a compact list of 4-6 keywords capturing tone, format, perspective, and key focus. The prompt instructs the model to maximize diversity across specifications (Figure 1 Left). This variant relies on the LLM's internal ability to propose diverse specifications.

Keyword. We prompt the LLM to generate a set of dimensions along which responses to the task can vary and a pool of candidate values for each (e.g., tone with values such as satirical or earnest). Each specification z_i is assembled by selecting one value per dimension. To ensure the selected specifications differ from one another, we use greedy Hamming-distance maximization, iteratively selecting the combination most distant from all previously chosen ones (Figure 1 right). This combinatorial construction encourages distinct outlines to differ across multiple axes.

5 EXPERIMENTS

Our evaluation studies the following questions: (1) Does injecting more informative diversity sources lead to more diverse outputs, without sacrificing quality (Sec 5.1)? (2) When a source is injected, does variation in the source contribute to output diversity, beyond what decoding randomness alone produces (Sec 5.2)? (3) How does output diversity scale with the number of outputs (Sec 5.3)?

We evaluate on five open-ended generation tasks spanning creative writing, conversation, poetry, argumentation, and general knowledge.

- **WritingPrompts** (Huang et al., 2024) contains Reddit writing prompts that ask for short fiction given a premise, spanning genres from science fiction to literary drama.
- **Infinite Chats** (Sorensen et al., 2024) consists of 26K open-ended user queries covering diverse everyday topics, requiring substantive, multi-paragraph responses.
- **NoveltyBench** (Zhang et al., 2025) is a benchmark designed to evaluate response diversity, covering creative writing, factual knowledge, and subjective opinion generation with prompts that admit multiple valid answers.
- **Poetry Foundation** (Poetry Foundation, 2023) provides poetry writing prompts with varied stylistic and thematic constraints.
- **IBM ArgKP** (Bar-Haim et al., 2020) contains debate topics for which the model must generate diverse argumentative key points.

Unless otherwise noted, all experiments use the default decoding settings of each backbone model. We provide the generated outputs in Appendix C.

5.1 SPECIFICATION-LEVEL INJECTION IMPROVES DIVERSITY WITH COMPARABLE QUALITY

We compare seven methods, including four Level 0 baselines (Direct Generation, Verbalized Sampling (Zhang et al., 2026), Multilingual Prompting (Wang et al., 2025) , and G2 (Ruan et al., 2025)), one Level 1 method (SSOT (Misaki & Akiba, 2026)), and two Level 2 methods proposed by this work (Keyword and Diverse Outline).

We use four backbone models: GPT-5-mini (Singh et al., 2025), Qwen3-4B (Yang et al., 2025), DeepSeek-V3.2 (Liu et al., 2025), and Mistral-Large-3 (Mistral AI, 2025). G2 requires access to model logits for its guide modules, which is only available for open-weight models. We therefore report G2 results with Qwen3-4B only. For open-weight models whose temperature can be set above one (Qwen3-4B and DeepSeek-V3.2), we additionally include a high-temperature baseline ($\tau = 1.5$). Detailed setup is elaborated in Appendix A.

Table 2: Diversity and quality on Infinite Chats and NoveltyBench. Methods are grouped by injection level. G2 and High Temperature require access to the model and are evaluated only on Qwen3-4B. Bold indicates the best result. ↑ = higher is better. Level 2 methods achieve higher diversity while maintaining comparable quality.

	Lv.	Method	Infinite Chats				NoveltyBench				
			LLM Div ↑	Emb Div ↑	Struct Div ↑	Quality ↑	LLM Div ↑	Emb Div ↑	Struct Div ↑	Distinct-N ↑	Quality ↑
GPT-5-mini	0	Direct	2.69±1.52	0.26±0.15	0.21±0.09	8.30±0.72	2.88±1.29	0.41±0.15	0.24±0.09	15.25±6.32	7.30±0.57
	0	Verbalized	2.84±1.18	0.23±0.11	0.20±0.07	**8.47±0.45**	3.10±1.21	0.36±0.11	0.22±0.07	9.76±4.09	7.30±0.62
	0	Multilingual	3.04±1.23	0.25±0.12	0.14±0.06	7.19±0.73	2.65±1.30	0.29±0.13	0.16±0.06	7.04±4.23	7.03±0.86
	1	SSOT	2.40±0.78	0.18±0.09	0.17±0.04	8.30±0.45	2.36±0.81	0.24±0.10	0.19±0.06	4.74±2.98	6.88±0.74
	2	Keyword (ours)	4.87±1.39	0.32±0.13	0.25±0.06	8.36±0.54	6.63±1.43	0.43±0.09	0.31±0.06	16.46±3.78	7.32±0.70
	2	Div Outline (ours)	**7.32±1.55**	**0.40±0.13**	**0.31±0.08**	8.28±0.61	**7.13±1.89**	**0.50±0.11**	**0.35±0.08**	**16.75±4.30**	**7.39±0.79**
Qwen3-4B	0	Direct	2.11±0.85	0.19±0.14	0.13±0.07	7.02±0.68	2.44±0.80	0.30±0.16	0.18±0.10	10.96±7.44	6.72±0.81
	0	Verbalized	2.31±0.91	0.25±0.15	0.13±0.07	6.55±1.04	2.77±0.80	0.37±0.14	0.18±0.09	11.01±5.09	6.26±1.29
	0	Direct (τ=1.5)	2.35±0.82	0.19±0.14	0.13±0.07	6.67±0.99	2.36±0.85	0.25±0.15	0.17±0.09	7.38±5.77	6.51±1.00
	0	G2	2.09±0.69	0.12±0.07	0.19±0.05	7.03±0.78	2.75±0.62	0.21±0.08	0.23±0.06	5.54±4.36	7.35±0.74
	0	Multilingual	3.32 ±1.19	0.26±0.12	0.14±0.08	6.66±0.77	3.58±1.13	0.32±0.11	0.17±0.08	11.02±4.71	6.40±0.86
	1	SSOT	2.21±0.66	0.15±0.09	0.15±0.06	6.70±0.85	2.13±0.74	0.20±0.12	0.17±0.08	6.03±4.78	6.36±1.05
	2	Keyword (ours)	**5.77±1.52**	0.33±0.09	0.22±0.06	**7.71±0.49**	**5.01±1.72**	0.38±0.09	0.25±0.08	13.21±5.58	**7.40±1.20**
	2	Div Outline (ours)	5.58±1.82	**0.35±0.12**	**0.24±0.08**	7.45±0.59	5.00±2.00	**0.41±0.11**	**0.27±0.09**	**13.85±5.79**	7.09±1.17

We evaluate our experimental results along five dimensions.

- **LLM-judged diversity**: for each pair of outputs from the same prompt, we ask GPT-5.3-Chat (Singh et al., 2025) to rate their pairwise diversity on a 1–10 scale and average over all pairs.
- **Lexical diversity**: we report Distinct-3 (proportion of unique trigrams) and Self-BLEU (mean BLEU of each output against all others.
- **Embedding diversity**: we compute the mean pairwise cosine distance ($1 - \cos$) between sentence embeddings of all outputs from the same prompt, using OpenAI's text-embedding-3-small (OpenAI, 2024).
- **Structural diversity**: we first prompt GPT-5.3-Chat to extract a short outline from each output, then compute embedding diversity over the extracted outlines rather than the full outputs. This captures variation in high-level organization rather than surface wording.
- **Quality**: we ask GPT-5.3-Chat to rate each output on a 0–10 scale for coherence, relevance, and overall quality, and report the average. To validate the reliability of the LLM-as-judge, we also conduct a human evaluation on 60 sampled response pairs (see Appendix B).

We sample 100 prompts per dataset for evaluation, except for ArgKP where we retain 41 prompts after filtering those containing harmful content. For each prompt, we generate 20 outputs and report the mean and std of each metric across prompts. For NoveltyBench, we additionally report its specialized Distinct-N metric, which counts the number of unique responses' equivalence classes.

Table 2 shows that both Keyword and Diverse Outline consistently achieve the highest diversity across all metrics on both benchmarks, while maintaining comparable quality to baselines. This suggests that introducing more informative specifications is an effective way to improve diversity in open-ended generation. Additional results across diversity metrics, datasets, and backbone models are reported in Appendix B.

5.2 Varying Injected Source Expands Output Diversity

To understand why Level-2 methods outperform, we examine how variation in the injected source contributes to output diversity. We apply the entropy decomposition introduced in Sec 3.3 to measure the transmission score T and realized diversity $I(Y; Z|X)$. This analysis requires an explicit per-output diversity source z_i. Level 0 methods have no such source, and both terms are undefined for them. We therefore compare only methods with explicit z_i. Table 3 reports these quantities alongside the output entropy $H(Y|X)$, estimated using the holdout procedure described in Sec 3.4.

Table 3: Transmission score T, realized diversity $I(Y;Z|X)$, and output entropy $H(Y|X)$ on Qwen3-4B. T and $I(Y;Z|X)$ are undefined for Level 0 methods. Level 2 methods achieve higher T and $I(Y;Z|X)$ than Level 1 method.

	Infinite Chats			Novelty Bench		
Method	$T\uparrow$	$I(Y;Z\|X)\uparrow$	$H(Y\|X)\uparrow$	$T\uparrow$	$I(Y;Z\|X)\uparrow$	$H(Y\|X)\uparrow$
Div Outline (ours)	0.552	1.74	2.98	0.507	1.43	2.88
Keyword (ours)	0.464	1.45	2.96	0.476	1.36	2.92
SSOT	0.003	0.00	0.80	0.006	0.00	0.73

As shown in Table 3, SSOT has near-zero transmission ($T \approx 0.003$) and near-zero realized diversity, with an output entropy $H(Y|X)$ of only 0.80, far below the Level 2 methods. Although SSOT introduces a per-output diversity source, its random strings carry little information about the desired output and are largely ignored during generation. In contrast, both Keyword and Diverse Outline achieve substantially higher transmission scores and realized diversity, with output entropy above 2.8. These results suggest that effective diversity injection requires both informative sources and reliable transmission from the source to the final output. More results are in Appendix B.

Figure 3 compares the output entropy of Direct Generation $H(Y'|X)$ against the residual entropy $H(Y|X,Z)$ of our methods when conditioned on a single fixed outline, capturing per-outline decoding randomness, and the total output entropy of our methods when the outline varies $H(Y|X)$. For both Keyword and Diverse Outline, the total entropy $H(Y|X)$ exceeds the entropy of Direct Generation, while the fixed-source entropy $H(Y|X,Z)$ falls below it. This ordering, $(H(Y|X,Z) < H(Y'|X) < H(Y|X)$, shows that conditioning on a single source narrows the per-source output distribution, yet varying the source across generations yields higher total diversity than Direct Generation. Thus, variation across injected sources contributes to the output diversity of specification-level methods.

We next provide a complementary analysis using embedding diversity. For each method, we vary the number of outlines $m \in \{2, 5, 10\}$, generating $k = 10, 4, 2$ outputs per outline respectively and measure two quantities: embedding diversity for the same outline (*within-outline diversity*) and for different outlines (*between-outline diversity*). If outline variation is the primary driver of diversity, between-outline diversity should exceed within-outline diversity. Figure 2 shows that between-outline diversity consistently exceeds within-outline diversity across all values of m, for both methods and datasets. Within-outline diversity remains roughly stable as m grows, indicating that repeated sampling under a fixed source contributes limited additional variation. Between-outline diversity increases with m, suggesting that adding more sources expands the range of generated outputs.

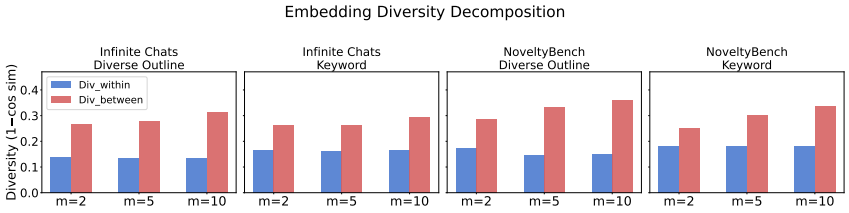


Figure 2: Within-outline diversity and between-outline diversity for Diverse Outline and Keyword, across $m \in \{2, 5, 10\}$ outlines on Infinite Chats and NoveltyBench. Between-outline diversity consistently exceeds within-outline diversity and grows with m, while within-outline diversity remains flat. More results are in Appendix B.

To directly test whether outline diversity drives output diversity, we perform a controlled intervention on the keyword method. As described in Sec 4, Keyword constructs each specification by selecting one value per dimension. By default, each specification has independent values for all dimensions. We reduce specification diversity by forcing specifications to share the same value on a subset of dimensions. As more dimensions are shared, specifications become more similar. As shown in Figure4, embedding diversity scales linearly with outline embedding diversity across both datasets, indicating that output variation is directly driven by the diversity of the injected specifications.

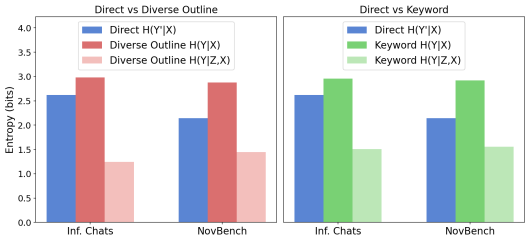


Figure 3: Entropy $H(Y'|X)$ of Direct Generation vs. outline-conditioned entropy $H(Y|X,Z)$ and total output entropy $H(Y|X)$ of Level 2 methods on NoveltyBench and Infinite Chats. Fixed-outline entropy falls below Direct Generation, yet total entropy exceeds it, showing that outline variation expands output diversity.

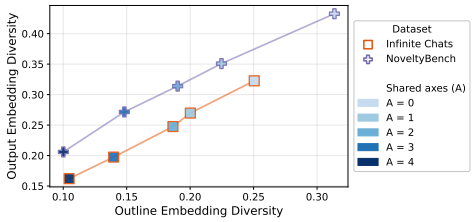


Figure 4: Effect of Outline Diversity on Output Diversity. We reduce outline diversity in the Keyword method by forcing outlines to share values on an increasing number of dimensions ($0 \rightarrow 4$). Output embedding diversity scales linearly with specification embedding diversity on both NoveltyBench and Infinite Chats.

5.3 Scaling with the Number of Generated Outputs

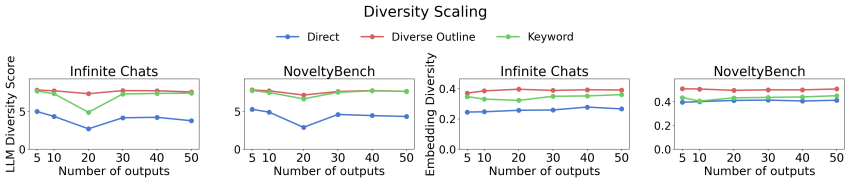


Figure 5: Diversity scaling as the number of generated outputs increases, using GPT-5-mini. Diverse Outline and Keyword maintain an advantage over Direct Generation across scales.

We investigate how output diversity scales with the number of generated outputs. Using GPT-5-mini as the backbone, we vary the number of outputs across $\{5, 10, 20, 30, 40, 50\}$ and compare Diverse Outline and Keyword against Direct Generation on Infinite Chats and NoveltyBench, reporting LLM-judged diversity and embedding diversity.

As shown in Figure 5, Diverse Outline and Keyword generally outperform Direct Generation across scales. In most cases, the gap remains visible as more outputs are generated. These results suggest that specification-level methods can make effective use of additional output slots by introducing diverse intermediate sources.

6 Conclusion

We introduced the Diversity Injection Framework, which unifies test-time diverse generation methods by the nature of their diversity source and quantifies diversity transmission via an entropy-based score. Guided by the framework, we proposed specification-level generation methods that inject diversity through detailed per-output configurations before generation begins. Diverse Outline lets the model propose all specifications in one pass and achieves high diversity in nearly all settings, while Keyword builds them from orthogonal axes, offering a more structured and interpretable construction. That both work through different source constructions shows that source design is the effective lever for diverse generation. Experiments across five open-ended tasks and four backbone models show that these methods substantially improve output diversity while maintaining comparable quality. We hope the framework and transmission score provide a foundation for diagnosing and improving diversity in future generation systems, and that specification-level injection offers a practical, training-free path toward richer, more varied LLM outputs.

AI USE STATEMENT

In this work, we used generative AI tools to edit the manuscript for readability and to polish the writing. The LLMs are also used for generating outputs and serving as judge models in our experiments as the objects of study. We have not used generative AI tools to generate synthetic data, develop the theoretical framework, formulate or prove mathematical claims, propose or refine hypotheses, design or evaluate the methodology or experiments, implement methods, interpret results, or clean or reformat datasets. Translation and qualitative data analysis are not applicable to this work. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski. A learning algorithm for boltzmann machines. *Cognitive Science*, 9(1):147–169, 1985. ISSN 0364-0213. doi: https://doi.org/10.1016/S0364-0213(85)80012-4. URL https://www.sciencedirect.com/science/article/pii/S0364021385800124.

Pulin Agrawal and Prasoon Goyal. Addressing llm diversity by infusing random concepts, 2026. URL https://arxiv.org/abs/2601.18053.

Roy Bar-Haim, Lilach Eden, Roni Friedman, Yoav Kantor, Dan Lahav, and Noam Slonim. From arguments to key points: Towards automatic argument summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4029–4039, 2020.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=4hturzLcKX.

Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 889–898, 2018.

Shirley Anugrah Hayati, Minhwa Lee, Dheeraj Rajagopal, and Dongyeop Kang. How far can we extract diverse perspectives from large language models? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 5336–5366, 2024.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rygGQyrFvH.

Xi Yu Huang, Krishnapriya Vishnubhotla, and Frank Rudzicz. The gpt-writingprompts dataset: A comparative analysis of character portrayal in short stories. *arXiv preprint arXiv:2406.16767*, 2024.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In Kevin Knight, Ani Nenkova, and Owen Rambow (eds.), *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 110–119, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1014. URL https://aclanthology.org/N16-1014/.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025.

Nguyen Nhat Minh, Andrew Baker, Clement Neo, Allen G Roush, Andreas Kirsch, and Ravid Shwartz-Ziv. Turning up the heat: Min-p sampling for creative and coherent LLM outputs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=FBkpCyujtS.

Kou Misaki and Takuya Akiba. String seed of thought: Prompting LLMs for distribution-faithful and diverse generation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=luXtbX1lVK.

Mistral AI. Mistral-large-3-675b-instruct-2512, 2025. URL https://huggingface.co/mistralai/Mistral-Large-3-675B-Instruct-2512.

OpenAI. New embedding models and API updates. https://openai.com/blog/new-embedding-models-and-api-updates, 2024.

Kyeongman Park, Nakyeong Yang, and Kyomin Jung. Avoidance decoding for diverse multi-branch story generation. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 7489–7505, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.381. URL https://aclanthology.org/2025.emnlp-main.381/.

Poetry Foundation. Poetryfoundation.org: Data summary, 2023. URL https://www.poetryfoundation.org. Public online archive.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 3982–3992, 2019.

Mark O Riedl and Robert Michael Young. Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research*, 39:217–268, 2010.

Rogelio E Cardona Rivera, Arnav Jhala, Julie Porteous, and R Michael Young. The story so far on narrative planning. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 34, pp. 489–499, 2024.

Zhiwen Ruan, Yixia Li, Yefeng Liu, Yun Chen, Weihua Luo, Peng Li, Yang Liu, and Guanhua Chen. G2: Guided generation for enhanced output diversity in llms. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 14127–14145, 2025.

Alexander Shypula, Shuo Li, Botong Zhang, Vishakh Padmakumar, Kayo Yin, and Osbert Bastani. Evaluating the diversity and quality of LLM generated content. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=O7bF6nlSOD.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Taylor Sorensen, Jared Moore, Jillian Fisher, Mitchell Gordon, Niloofar Mireshghallah, Christopher Michael Rytting, Andre Ye, Liwei Jiang, Ximing Lu, Nouha Dziri, Tim Althoff, and Yejin Choi. Position: a roadmap to pluralistic alignment. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Yixuan Su and Nigel Collier. Contrastive search is what you need for neural text generation. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=GbkWw3jwL9.

Sangho Suh, Meng Chen, Bryan Min, Toby Jia-Jun Li, and Haijun Xia. Luminate: Structured generation and exploration of design space with large language models for human-ai co-creation. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400703300. doi: 10.1145/3613904.3642400. URL https://doi.org/10.1145/3613904.3642400.

Ashwin Vijayakumar, Michael Cogswell, Ramprasaath Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. Diverse beam search for improved description of complex scenes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Phoebe J Wang and Max Kreminski. Guiding and diversifying llm-based story generation via answer set programming. *arXiv preprint arXiv:2406.00554*, 2024.

Qihan Wang, Shidong Pan, Tal Linzen, and Emily Black. Multilingual prompting for improving LLM generation diversity. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 6367–6389, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.324. URL https://aclanthology.org/2025.emnlp-main.324/.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Kevin Yang, Dan Klein, Nanyun Peng, and Yuandong Tian. Doc: Improving long story coherence with detailed outline control. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3378–3465, 2023.

Jiayi Zhang, Simon Yu, Derek Chong, Anthony Sicilia, Michael Tomz, Christopher D Manning, and Weiyan Shi. Verbalized sampling: How to mitigate mode collapse and unlock LLM diversity. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=zloIrd77G5.

Tianhui Zhang, Bei Peng, and Danushka Bollegala. Improving diversity of commonsense generation by large language models via in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 9226–9242, 2024.

Yiming Zhang, Harshita Diddee, Susan Holm, Hanchen Liu, Xinyue Liu, Vinay Samuel, Barry Wang, and Daphne Ippolito. Noveltybench: Evaluating creativity and diversity in language models. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=XZm1ekzERf.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=uccHPGDlao.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A benchmarking platform for text generation models. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pp. 1097–1100, 2018.

A Models and Evaluation Setup

We evaluate on four backbone models: **GPT-5-mini** (Azure API), **DeepSeek-V3.2** (Azure API), **Mistral-Large-3** (Azure API), and **Qwen3-4B** (served locally via vLLM). All models use default decoding settings ($\tau = 1.0$) unless stated otherwise. For DeepSeek-V3.2 and Qwen3-4B-Instruct-2507, we additionally include a high-temperature baseline (Direct at τ=1.5; Qwen additionally uses top-$p = 0.9$). Each experiment generates 20 outlines and one output per outline for 100 prompts per dataset.

For the entropy estimation in Table 3 and Figure 3, we reuse the same 20 specifications and outputs in Table 2. We randomly split the 20 specifications into an estimation set of 10 and an evaluation set of 10 to estimate $H(Y|X)$, $H(Y|X, Z)$ and $H(Z|X)$ following the procedure in Section 3.4.

For the diversity decomposition experiment in Figure 2, we fix the total number of outputs at 20 and vary the number of outlines $m \in \{2, 5, 10\}$, generating 10, 4, and 2 outputs per outline, respectively.

LLM-based diversity and quality scoring is performed by **GPT-5.3-Chat** as the judge model. Embedding diversity (both at the output level and the structural level) is computed using OpenAI's **text-embedding-3-small** model, measuring mean pairwise cosine distance $(1 - \text{cosine similarity})$ across the outputs.

B Additional Experiment Results

B.1 Additional Results for Sec 5.1.

We provide additional results across datasets, backbone models, and metrics in this section.

Table 4 reports diversity and quality on NoveltyBench and Infinite Chat on Deepseek-V3.2 and Mistral-Large-3. The same pattern as in Table 2 holds. Level 2 methods achieve the highest diversity across most diversity metrics while maintaining comparable quality.

Tables 5 and 6 extend the evaluation to three more datasets: IBM ArgKP (Bar-Haim et al., 2020), Poetry Foundation (Poetry Foundation, 2023), and WritingPrompts (Huang et al., 2024). The same evaluation protocol is applied. Level 2 methods consistently outperform Level 0 and Level 1 baselines on all three diversity metrics (LLM Div, Emb Div, Struct Div) across models and datasets, while maintaining competitive quality scores.

Table 7 and Table 8 report lexical diversity, including Distinct-3, which measures the proportion of unique trigrams across outputs, and Self-Bleu for all methods across all datasets and models. Level-2 methods generally improve or match lexical diversity across most settings. For DeepSeek-V3.2, high temperature produces high token-level randomness, resulting in lower Self-BLEU. However, this comes without the structural diversity gains that Level-2 methods provide (Tables 4-6).

We additionally validate the quality evaluator with a human study on 60 response pairs. Using outputs from the Qwen3-4B and GPT-5-mini experiments on NoveltyBench and Infinite Chats, we sampled 5 prompts from each model–dataset setting and 3 response pairs per prompt, with each pair comparing Diverse Outline against Direct Generation. Human judgments agreed with the LLM-based evaluator on 50 of the 60 pairs (83.3%). That is, the response preferred by the human annotator also received the higher quality score from the LLM evaluator. The agreement corresponds to Cohen's $\kappa = 0.653$.

B.2 Additional Results for Sec 5.2.

Table 9 extends the transmission-score analysis to WritingPrompts, Poetry Foundation, and IBM ArgKP, reporting T, $I(Y; Z \mid X)$, and $H(Y \mid X)$ for Diverse Outline, Keyword, and SSOT. Level-2 methods achieve substantially higher transmission scores and realized diversity than SSOT across all three datasets.

Additionally, Figure 6 shows the within-outline and between-outline embedding diversity decomposition for Diverse Outline and Keyword on these datasets. The same pattern as in Figure 2 holds: between-outline diversity consistently exceeds within-outline diversity and grows with the number of outlines, while within-outline diversity remains flat, indicating that source variation contributes to the diversity gains of Level 2 methods across task types.

Table 4: Diversity and quality on Infinite Chats and NoveltyBench using DeepSeek-V3.2 and Mistral-Large-3. Methods are grouped by injection level. Bold indicates the best result. ↑ = higher is better.

	Lv.	Method	Infinite Chats				NoveltyBench				
			LLM Div ↑	Emb Div ↑	Struct Div ↑	Quality ↑	LLM Div ↑	Emb Div ↑	Struct Div ↑	Distinct-N ↑	Quality ↑
DeepSeek-V3.2	0	Direct	3.37±1.59	0.29±0.16	0.21±0.07	**7.51±0.55**	3.39±1.44	0.42±0.16	0.25±0.08	17.87±4.00	7.39±0.49
	0	Direct (τ=1.5)	3.36±1.65	0.32±0.13	0.21±0.08	7.07±0.73	3.59±1.51	0.43±0.14	0.25±0.09	**19.28±2.64**	6.90±0.82
	0	Verbalized	3.03±1.30	0.31±0.15	0.19±0.06	6.88±1.00	3.22±1.14	0.39±0.15	0.22±0.06	11.55±4.11	6.79±0.72
	1	SSOT	2.86±0.96	0.26±0.16	0.19±0.06	6.97±1.09	3.25±0.77	0.38±0.13	0.22±0.07	11.18±4.40	6.53±1.28
	2	Keyword (ours)	5.88±1.44	0.40±0.13	0.25±0.06	7.50±0.81	6.77±1.85	0.47±0.10	0.29±0.07	17.41±3.75	**7.77±0.80**
	2	Div Outline (ours)	**7.07±1.76**	**0.44±0.13**	**0.27±0.07**	7.36±0.88	**7.13±2.12**	**0.50±0.11**	**0.32±0.08**	18.04±3.33	7.43±0.97
Mistral-L3	0	Direct	2.33±0.88	0.28±0.17	0.13±0.06	6.92±0.68	2.59±0.87	0.45±0.13	0.16±0.07	14.48±5.77	6.80±0.58
	0	Verbalized	2.63±0.94	0.26±0.15	0.12±0.05	6.91±0.85	3.12±1.02	0.37±0.13	0.15±0.07	9.69±4.17	6.81±0.62
	1	SSOT	2.78±0.72	0.21±0.12	0.12±0.05	7.26±0.73	2.70±0.76	0.30±0.12	0.14±0.06	9.77±4.28	6.74±1.07
	2	Keyword (ours)	6.51±1.32	0.35±0.11	0.19±0.05	**8.02±0.40**	7.33±1.18	0.42±0.08	0.23±0.06	15.13±5.62	**8.16±0.52**
	2	Div Outline (ours)	**7.65±1.67**	**0.40±0.12**	**0.23±0.07**	7.86±0.46	**7.84±1.72**	**0.46±0.10**	**0.26±0.08**	**15.74±5.46**	7.91±0.89

Table 5: Diversity and quality on IBM ArgKP and Poetry Foundation. Methods are grouped by injection level. DeepSeek-V3.2 and Qwen3-4B include a high-temperature baseline (Direct at τ=1.5). **Bold** = best per column within each model. ↑ = higher is better.

	Lv.	Method	IBM ArgKP				Poetry Foundation			
			LLM Div ↑	Emb Div ↑	Struct Div ↑	Quality ↑	LLM Div ↑	Emb Div ↑	Struct Div ↑	Quality ↑
GPT-5-mini	0	Direct	3.51±1.09	0.23±0.06	0.23±0.04	7.26±0.41	2.44±0.85	0.11±0.04	0.13±0.04	8.30±0.25
	0	Verbalized	2.95±0.76	0.19±0.05	0.22±0.05	7.30±0.43	2.99±0.89	0.11±0.04	0.15±0.03	**8.40±0.25**
	1	SSOT	2.51±0.47	0.15±0.07	0.18±0.04	6.93±0.54	3.15±0.67	0.10±0.02	0.13±0.03	8.32±0.21
	2	Keyword (ours)	6.17±0.92	0.28±0.04	0.31±0.05	7.32±0.20	7.98±1.25	0.22±0.06	0.24±0.04	7.94±0.85
	2	Div Outline (ours)	**7.97±1.07**	**0.33±0.07**	**0.34±0.05**	**7.37±0.30**	**8.83±1.29**	**0.23±0.06**	**0.27±0.04**	8.18±0.85
DeepSeek-V3.2	0	Direct	3.71±1.32	0.30±0.05	0.21±0.05	7.44±0.21	2.90±1.26	0.23±0.08	0.17±0.04	7.99±0.32
	0	Direct (τ=1.5)	3.58±1.27	0.30±0.08	0.23±0.05	7.26±0.33	2.80±1.02	0.25±0.08	0.15±0.04	7.73±0.37
	0	Verbalized	2.32±0.48	0.20±0.03	0.18±0.04	7.36±0.25	3.23±0.95	0.25±0.07	0.16±0.04	8.01±0.27
	1	SSOT	2.29±0.54	0.13±0.02	0.14±0.03	7.53±0.25	4.26±1.17	0.16±0.05	0.17±0.04	7.97±0.36
	2	Keyword (ours)	5.32±1.30	0.29±0.05	0.24±0.04	**7.73±0.23**	7.39±1.44	0.30±0.06	0.22±0.05	8.10±0.25
	2	Div Outline (ours)	**7.66±1.52**	**0.35±0.05**	**0.30±0.05**	7.71±0.19	**8.78±1.03**	**0.37±0.07**	**0.23±0.05**	**8.18±0.19**
Mistral-L3	0	Direct	2.60±0.93	0.24±0.09	0.11±0.06	6.66±0.37	1.92±0.38	0.14±0.05	0.08±0.03	7.61±0.34
	0	Verbalized	2.22±0.43	0.14±0.03	0.10±0.04	7.14±0.27	2.61±0.60	0.19±0.05	0.10±0.03	7.76±0.26
	1	SSOT	2.21±0.37	0.09±0.02	0.09±0.02	7.39±0.25	4.17±0.94	0.12±0.03	0.10±0.03	8.01±0.36
	2	Keyword (ours)	5.79±1.51	0.26±0.04	0.17±0.03	**7.94±0.22**	6.96±0.98	0.20±0.05	0.15±0.04	8.18±0.31
	2	Div Outline (ours)	**7.93±1.90**	**0.37±0.05**	**0.21±0.03**	7.81±0.28	**8.03±1.17**	**0.26±0.06**	**0.16±0.04**	**8.19±0.28**
Qwen3-4B	0	Direct	1.72±0.40	0.14±0.04	0.12±0.03	6.89±0.33	2.46±0.52	0.10±0.06	0.11±0.03	7.47±0.77
	0	Direct (τ=1.5)	2.02±0.82	0.14±0.07	0.12±0.03	6.78±0.52	2.83±0.57	0.10±0.05	0.12±0.04	7.13±0.87
	0	G2	1.59±0.41	0.07±0.02	0.16±0.05	7.03±0.25	2.46±0.48	0.08±0.03	0.19±0.05	7.19±0.96
	0	Verbalized	1.76±0.41	0.12±0.03	0.12±0.03	6.97±0.30	2.56±0.62	0.12±0.04	0.12±0.04	7.45±0.52
	1	SSOT	1.64±0.37	0.09±0.04	0.15±0.04	6.98±0.28	2.96±0.64	0.19±0.08	0.15±0.04	7.29±0.58
	2	Keyword (ours)	5.75±1.43	0.27±0.06	0.22±0.04	7.38±0.37	6.36±0.97	0.29±0.05	0.21±0.04	**8.11±0.17**
	2	Div Outline (ours)	**6.08±1.39**	**0.31±0.07**	**0.25±0.04**	**7.42±0.29**	**7.21±1.57**	**0.29±0.06**	**0.23±0.05**	8.04±0.23

Table 6: Diversity and quality on WritingPrompts. Methods are grouped by injection level. DeepSeek-V3.2 and Qwen3-4B include a high-temperature baseline (Direct at τ=1.5). **Bold** = best per column within each model. ↑ = higher is better.

			WritingPrompts			
	Lv.	**Method**	LLM Div ↑	Emb Div ↑	Struct Div ↑	Quality ↑
GPT-5-mini	0	Direct	4.96±2.38	0.28±0.10	0.22±0.08	8.59±0.21
	0	Verbalized	4.69±1.66	0.25±0.07	0.20±0.06	8.71±0.20
	1	SSOT	3.55±1.36	0.21±0.08	0.17±0.06	8.17±1.50
	2	Keyword (ours)	6.16±1.31	0.29±0.05	0.23±0.05	**8.86±0.14**
	2	Div Outline (ours)	**8.71±1.07**	**0.37±0.07**	**0.29±0.06**	8.73±0.18
DeepSeek-V3.2	0	Direct	5.31±1.98	0.34±0.11	0.21±0.06	8.21±0.37
	0	Direct (τ=1.5)	4.74±2.10	0.29±0.13	0.20±0.06	7.91±0.43
	0	Verbalized	4.73±1.57	0.29±0.10	0.19±0.05	8.19±0.37
	1	SSOT	3.74±1.07	0.24±0.09	0.17±0.04	8.01±1.00
	2	Keyword (ours)	6.93±1.27	0.34±0.06	0.23±0.05	**8.40±0.27**
	2	Div Outline (ours)	**8.07±1.94**	**0.38±0.10**	**0.26±0.06**	8.16±1.46
Mistral-L3	0	Direct	2.46±1.02	0.21±0.12	0.12±0.05	7.46±0.47
	0	Verbalized	3.26±1.36	0.23±0.09	0.13±0.04	7.74±0.73
	1	SSOT	3.76±1.50	0.20±0.08	0.13±0.04	7.56±1.83
	2	Keyword (ours)	6.15±1.45	0.29±0.06	0.18±0.04	**8.35±0.26**
	2	Div Outline (ours)	**7.86±2.05**	**0.34±0.07**	**0.20±0.05**	8.13±1.19
Qwen3-4B	0	Direct	1.89±0.72	0.15±0.09	0.11±0.05	7.10±0.86
	0	Direct (τ=1.5)	2.22±0.81	0.16±0.09	0.12±0.04	6.79±1.02
	0	Verbalized	2.06±0.98	0.14±0.07	0.11±0.04	6.88±0.73
	0	G2	2.00±0.69	0.09±0.03	0.22±0.05	6.81±1.01
	1	SSOT	2.12±0.83	0.14±0.07	0.15±0.05	6.56±0.89
	2	Keyword (ours)	5.74±1.57	0.30±0.05	0.20±0.05	**8.19±0.21**
	2	Div Outline (ours)	**6.51±1.80**	**0.34±0.06**	**0.22±0.06**	8.04±0.29

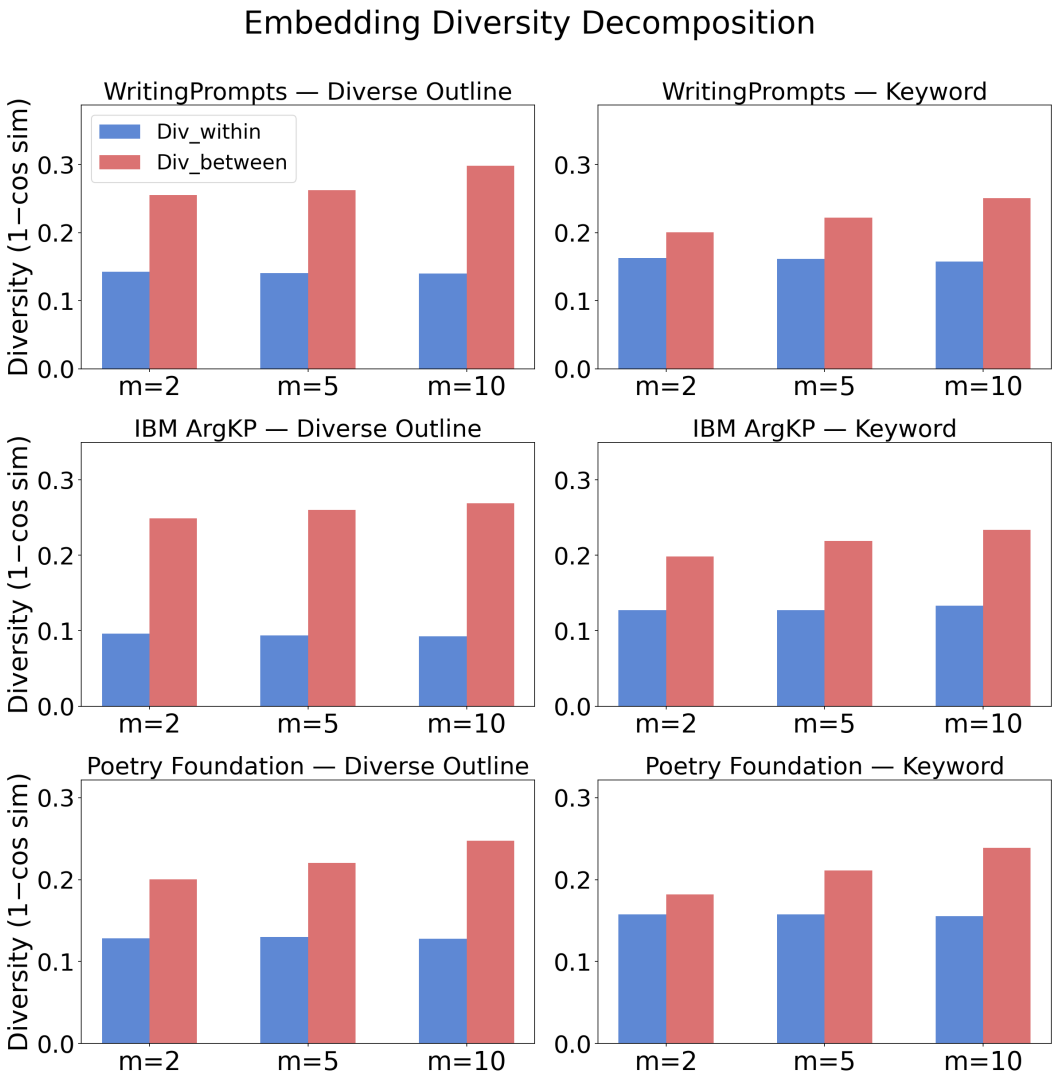


Figure 6: Within-outline diversity and between-outline diversity for Diverse Outline and Keyword, across $m \in \{2, 5, 10\}$ outlines on WritingPrompts, ArgKP and Poetry Foundation. Between-outline diversity consistently exceeds within-outline diversity and grows with m, while within-outline diversity remains flat.

Table 7: Lexical diversity (Distinct-3 and Self-BLEU) on Infinite Chats and NoveltyBench. Methods are grouped by injection level. DeepSeek-V3.2 and Qwen3-4B include a high-temperature baseline (Direct at τ=1.5). **Bold** = best per column within each model. ↑ = higher is better, ↓ = lower is better.

			Infinite Chats		NoveltyBench	
	Lv.	Method	Distinct-3 ↑	Self-BLEU ↓	Distinct-3 ↑	Self-BLEU ↓
GPT-5-mini	0	Direct	0.87±0.09	0.26±0.14	0.89±0.07	0.23±0.11
	0	Verbalized	0.88±0.07	0.26±0.11	0.87±0.06	0.28±0.10
	0	Multilingual	0.86 ±0.09	0.29±0.13	0.85 ±0.11	0.30 ±0.16
	1	SSOT	0.78±0.11	0.38±0.14	0.76±0.11	0.42±0.15
	2	Keyword (ours)	0.91±0.07	0.21±0.10	0.92±0.04	0.17±0.07
	2	Div Outline (ours)	**0.94±0.04**	**0.12±0.07**	**0.95±0.04**	**0.11±0.06**
DeepSeek-V3.2	0	Direct	0.91±0.05	0.19±0.09	0.90±0.06	0.21±0.09
	0	Direct (τ=1.5)	**0.98±0.04**	**0.06±0.05**	**0.97±0.10**	**0.07±0.05**
	0	Verbalized	0.85±0.08	0.31±0.11	0.82±0.12	0.34±0.14
	1	SSOT	0.77±0.12	0.40±0.15	0.78±0.14	0.37±0.13
	2	Keyword (ours)	0.92±0.06	0.17±0.09	0.94±0.04	0.14±0.06
	2	Div Outline (ours)	0.93±0.05	0.16±0.09	0.92±0.09	0.15±0.07
Mistral-L3	0	Direct	0.62±0.23	0.57±0.27	0.70±0.17	0.46±0.20
	0	Verbalized	0.71±0.12	0.49±0.13	0.70±0.13	0.50±0.14
	1	SSOT	0.71±0.11	0.48±0.13	0.66±0.15	0.52±0.16
	2	Keyword (ours)	0.89±0.05	0.22±0.08	**0.91±0.03**	0.18±0.05
	2	Div Outline (ours)	**0.91±0.05**	**0.19±0.08**	0.91±0.09	**0.17±0.06**
Qwen3-4B	0	Direct	0.79±0.12	0.37±0.15	0.78±0.13	0.37±0.14
	0	Direct (τ=1.5)	0.86±0.11	0.28±0.15	0.81±0.13	0.34±0.15
	0	Verbalized	0.78±0.10	0.40±0.13	0.77±0.18	0.36±0.13
	0	G2	0.71±0.11	0.47±0.13	0.80±0.06	0.37±0.08
	0	Multilingual	0.79 ±0.14	0.42±0.20	0.85 ±0.09	0.33 ±0.15
	1	SSOT	0.67±0.12	0.51±0.13	0.68±0.12	0.51±0.14
	2	Keyword (ours)	0.88±0.07	**0.23±0.08**	0.87±0.10	0.24±0.07
	2	Div Outline (ours)	**0.88±0.06**	0.23±0.09	**0.88±0.06**	**0.24±0.09**

B.3 Ablation Study: Source Content vs. Pipeline Structure

Our Level-2 methods use a two-step pipeline in which an intermediate specification is generated before the final output. To verify that the observed diversity improvements stem from the informative content of the injected specifications rather than the mere addition of a two-step generation pipeline, we conduct an ablation study. We compare our approach against a two-step control (2 Step-only) whose intermediate step does not introduce diversified specification content. As shown in Table 10, Diverse Outline substantially improves LLM-judged, embedding, and structural diversity over the two-step control while maintaining comparable quality. Table 11 shows the same pattern for lexical diversity, with higher Distinct-3 and substantially lower Self-BLEU. Because both methods use the same two-step generation structure, these results indicate that the additional generation step alone does not explain the diversity gains. The gains depend on the diversified content of the intermediate specifications.

B.4 Diversity in the Specification Generation

We further examine whether Diverse Outline consistently generates sufficiently diverse specifications and how specification diversity relates to the resulting output diversity in this section. For each prompt, we compute the embedding diversity of the generated specifications and of the corresponding

Table 8: Lexical diversity (Distinct-3 and Self-BLEU) on WritingPrompts, IBM ArgKP, and Poetry Foundation. Methods are grouped by injection level. DeepSeek-V3.2 and Qwen3-4B include a high-temperature baseline (Direct at τ=1.5). **Bold** = best per column within each model. ↑ = higher is better, ↓ = lower is better.

	Lv.	Method	WritingPrompts		IBM ArgKP		Poetry Foundation	
			Distinct-3 ↑	Self-BLEU ↓	Distinct-3 ↑	Self-BLEU ↓	Distinct-3 ↑	Self-BLEU ↓
GPT-5-mini	0	Direct	0.92±0.07	0.18±0.11	0.94±0.02	0.16±0.04	0.88±0.04	0.25±0.05
	0	Verbalized	0.92±0.05	0.18±0.09	0.91±0.02	0.23±0.04	**0.90±0.03**	0.22±0.05
	1	SSOT	0.84±0.16	0.27±0.11	0.81±0.03	0.38±0.04	0.88±0.03	0.25±0.05
	2	Keyword (ours)	0.94±0.04	0.14±0.06	0.91±0.01	0.22±0.02	0.85±0.09	0.16±0.05
	2	Div Outline (ours)	**0.95±0.03**	**0.10±0.05**	**0.96±0.01**	**0.11±0.03**	0.88±0.09	**0.14±0.05**
DeepSeek-V3.2	0	Direct	0.93±0.04	0.17±0.07	0.95±0.02	0.13±0.04	0.91±0.05	0.21±0.08
	0	Direct (τ=1.5)	**0.98±0.02**	**0.07±0.05**	**0.98±0.01**	**0.06±0.03**	**0.97±0.02**	**0.09±0.05**
	0	Verbalized	0.91±0.04	0.20±0.07	0.89±0.02	0.28±0.04	0.91±0.06	0.21±0.07
	1	SSOT	0.87±0.05	0.28±0.08	0.78±0.04	0.43±0.05	0.86±0.04	0.28±0.04
	2	Keyword (ours)	0.93±0.03	0.17±0.05	0.92±0.02	0.20±0.04	0.87±0.08	0.17±0.05
	2	Div Outline (ours)	0.91±0.16	0.13±0.05	0.94±0.01	0.16±0.03	0.90±0.04	0.13±0.05
Mistral-L3	0	Direct	0.51±0.19	0.70±0.22	0.69±0.20	0.50±0.25	0.35±0.11	0.89±0.12
	0	Verbalized	0.79±0.08	0.40±0.10	0.74±0.06	0.48±0.07	0.80±0.04	0.38±0.06
	1	SSOT	0.77±0.18	0.37±0.09	0.69±0.04	0.52±0.05	0.81±0.04	0.35±0.05
	2	Keyword (ours)	0.88±0.03	0.24±0.05	0.89±0.02	0.24±0.04	0.83±0.03	0.27±0.04
	2	Div Outline (ours)	**0.89±0.13**	**0.18±0.05**	**0.91±0.02**	**0.18±0.03**	**0.87±0.03**	**0.22±0.04**
Qwen3-4B	0	Direct	0.79±0.08	0.39±0.10	0.85±0.03	0.32±0.06	0.80±0.06	0.37±0.07
	0	Direct (τ=1.5)	0.83±0.10	0.33±0.13	0.87±0.10	0.26±0.16	**0.89±0.05**	0.23±0.07
	0	Verbalized	0.78±0.08	0.40±0.09	0.82±0.03	0.36±0.03	0.84±0.05	0.32±0.07
	0	G2	0.71±0.09	0.48±0.10	0.70±0.04	0.51±0.04	0.80±0.06	0.37±0.07
	1	SSOT	0.71±0.08	0.47±0.09	0.71±0.04	0.50±0.04	0.82±0.09	0.29±0.05
	2	Keyword (ours)	0.89±0.03	0.22±0.04	**0.89±0.03**	**0.23±0.05**	0.89±0.04	0.19±0.05
	2	Div Outline (ours)	**0.90±0.03**	**0.20±0.05**	0.89±0.02	0.23±0.04	0.89±0.04	**0.19±0.05**

outputs. Using SSOT as a reference point, we classify the specifications as diverse if their diversity reaches the structural diversity of SSOT in the corresponding setting, and classify the outputs as diverse if their diversity reaches the embedding diversity of SSOT.

Table 12 summarizes the results across Infinite Chats and NoveltyBench. Diverse Outline produces diverse specifications for most prompts: 97/100 and 95/100 prompts on Infinite Chats with GPT-5-mini and Qwen3-4B, respectively, and 97/100 and 90/100 prompts on NoveltyBench. Among prompts with diverse specifications, the resulting outputs are also diverse in most cases. These results indicate that Diverse Outline generally succeeds in producing distinct specifications.

C Experiment Prompts and Example Outputs

C.1 Diverse Outline: Prompts and Generation Details

As described in Section 4, Diverse Outline generates all m specifications in a single LLM call, then conditions on each to produce one output. We provide the prompt details for both steps below.

Outline generation. Each specification z_i takes the form of a compact keyword outline: a list of 4-6 keywords capturing tone, format, perspective, and key focus. The system prompt instructs the

Table 9: Transmission score T, realized diversity $I(Y;Z \mid X)$, and output entropy $H(Y \mid X)$ on Qwen3-4B for WritingPrompts, Poetry Foundation, and IBM ArgKP. T and $I(Y;Z \mid X)$ are undefined for Level 0 methods. Level 2 methods consistently achieve higher T and $I(Y;Z \mid X)$ than the Level 1 baseline.

Data	Method	$T \uparrow$	$I(Y;Z\|X) \uparrow$	$H(Y\|X) \uparrow$
Writing Prompts	Div Outline (ours)	**0.463**	1.46	3.08
	Keyword (ours)	0.388	1.12	2.85
	SSOT	0.003	0.00	0.64
Poetry Found.	Div Outline (ours)	**0.243**	0.94	3.86
	Keyword (ours)	0.165	0.65	3.94
	SSOT	0.000	0.00	1.22
IBM ArgKP	Div Outline (ours)	**0.565**	1.61	2.82
	Keyword (ours)	0.457	1.25	2.70
	SSOT	0.002	0.00	0.82

Table 10: Diversity and quality of Diverse Outline compared with a two-step control whose intermediate step does not introduce diversified specifications.

Dataset	Method	LLM Div	Emb Div	Struct Div	Quality
Infinite Chats	2 Step-only	2.91 $\pm$1.01	0.20 $\pm$0.12	0.257 $\pm$0.09	7.70 $\pm$0.70
Infinite Chats	Diverse Outline	7.32 $\pm$1.55	0.40 $\pm$0.13	0.31 $\pm$0.08	8.28 $\pm$0.61
NoveltyBench	2 Step-only	2.91 $\pm$1.04	0.25 $\pm$0.11	0.29 $\pm$0.08	7.46 $\pm$0.99
NoveltyBench	Diverse Outline	7.13 $\pm$1.89	0.50 $\pm$0.11	0.35 $\pm$0.08	7.39 $\pm$0.79

model to produce such outlines simultaneously, requiring that no two outlines share more than one keyword. Because the model sees its own previous outlines while generating each subsequent one, it naturally enforces diversity across the full set. The full prompt is shown in Figure 7.

System Prompt — Diverse Outline Generation

You are a diversity planning engine. Goal: produce {N} DISTINCT keyword outlines for the SAME request.

Rules:

1. Each outline is a compact list of 4–6 keywords/phrases capturing: tone, format, perspective, and key focus. Good examples: "instructional tone", "bullet-point format", "second-person", "focus on failure cases".
2. Maximize diversity: no two outlines should share more than 1 keyword.
3. Keywords must be task-specific — encode what makes each approach different for this particular request.

Output JSON only:

```
{
  "outlines": [
    { "id": 1,
    "keywords": ["keyword1", ..., "keyword4"] },
    ...
  ]
}
```

User Prompt

Task: {TASK}

Figure 7: Prompt used to generate m diverse keyword outlines in one batch call in Diverse Outline.

Table 11: Lexical diversity of Diverse Outline compared with the two-step control.

Dataset	Method	Distinct-3	Self-BLEU
Infinite Chats	2 Step-only	0.88 $\pm$0.10	0.25 $\pm$0.13
Infinite Chats	Diverse Outline	0.94 $\pm$0.04	0.12 $\pm$0.07
NoveltyBench	2 Step-only	0.87 $\pm$0.12	0.25 $\pm$0.12
NoveltyBench	Diverse Outline	0.95 $\pm$0.04	0.11 $\pm$0.06

Table 12: Prompt-level relationship between specification diversity and output diversity for Diverse Outline. Specification diversity is evaluated relative to the structural diversity of SSOT, and output diversity relative to the embedding diversity of SSOT in the corresponding setting.

Dataset	Model	Spec. Diverse		Spec. Not Diverse	
		Output Diverse	Output Not Diverse	Output Diverse	Output Not Diverse
Infinite Chats	GPT-5-mini	96	1	3	0
Infinite Chats	Qwen3-4B	90	5	5	0
NoveltyBench	GPT-5-mini	97	0	2	1
NoveltyBench	Qwen3-4B	87	3	8	2

Output generation. Each outline is then used to condition the generation of one output. The model is guided by the keyword list but not rigidly required to reproduce every keyword verbatim; the keywords shape the overall tone, format, and focus of the response. The generation prompt is shown in Figure 8.

System Prompt — Output Generation (Diverse Outline)

You are a helpful AI assistant. Your task is to respond to the given task, guided by the provided keywords.

Input:
- **Task:** the original request (follow ALL constraints in it, e.g. length, format, topic).
- **Outline:** a JSON object with axis:value pairs describing structural choices (e.g. narrative_pov, tone, structure, theme).

Instructions:
1. Strictly follow any constraints in the Task.
2. Let each keyword shape the tone, format, and focus of your response.
3. Keep your response to approximately 200 words.
4. Do not list or explain the keywords. Output ONLY the final response text.

Figure 8: Prompt used to generate outputs conditioned on a Diverse Outline specification.

C.2 Keyword: Axis Generation and Combination Strategy

As described in Section 4, Keyword first generates a set of dimensions and candidate values, then selects m combinations to construct specifications. We provide the prompt and selection details below.

Axis generation. For each prompt, we generate $A = 4$ dimensions (referred to as axes in the prompt) and $V = 8$ values each in a single call. Axes are required to be orthogonal: selecting a value on one axis should not constrain choices on other axes. The dimensions are generated once per prompt and shared across all outputs. The full prompt is shown in Figure 9.

Combination selection. Given A axes each with V values, there are $V^A = 8^4 = 4{,}096$ possible axis-value combinations. We select $m = 20$ combinations using a *greedy Hamming-distance maximization* strategy.

System Prompt — Axis Generation

You are a creative diversity planner. Generate {N_KEYS} independent structural dimensions (axes) that capture the most impactful creative choices for responses to the given task. For each axis, generate {N_VALUES} distinct values representing meaningfully different choices. Axes must be orthogonal — choosing a value for one axis should not constrain choices for other axes.

Output JSON only:

```
{
  "axes": [
    {
      "key": "narrative_pov",
      "label": "Narrative POV",
      "values": ["first-person unreliable",
                 "third-person omniscient", ...]
    }, ...
  ]
}
```

User Prompt

Task type: {TASK_TYPE} Task: {TASK}
Generate {N_KEYS} axes with {N_VALUES} values each.

Figure 9: Prompt used to generate diversity axes for Keyword. We use $A = 4$ axes and $V = 8$ values per axis.

Formally, represent each combination as a tuple $\mathbf{c} = (v_1, v_2, v_3, v_4) \in \{1, \ldots, V\}^A$. The Hamming distance between two combinations $\mathbf{c}$ and $\mathbf{c}'$ is the number of axes on which they differ:

$$d_H(\mathbf{c}, \mathbf{c}') = \sum_{a=1}^{A} \mathbf{1}[v_a \neq v'_a].$$

Two combinations with $d_H = A$ differ on every axis; two with $d_H = 0$ are identical.

The algorithm proceeds greedily:

1. Draw a random combination $\mathbf{c}_1$ uniformly from the full space as the seed.
2. For $i = 2, \ldots, m$: select the candidate $\mathbf{c}^*$ that maximizes its *minimum* Hamming distance to all already-selected combinations,

$$\mathbf{c}_i = \arg\max_{\mathbf{c} \notin \mathcal{S}} \min_{\mathbf{c}' \in \mathcal{S}} d_H(\mathbf{c}, \mathbf{c}'),$$

   where $\mathcal{S} = \{\mathbf{c}_1, \ldots, \mathbf{c}_{i-1}\}$ is the set of already-selected combinations. Ties are broken randomly.

This max-min criterion acts as a greedy farthest-point sampler in Hamming space, ensuring that every newly added combination is as different as possible from all existing ones on as many axes as possible. For $m = 20$ and $A = 4$, the resulting set typically achieves a minimum pairwise Hamming distance of ≥ 2, meaning no two selected outlines agree on more than $A - 2 = 2$ axes simultaneously.

Output generation. Each of the m selected combinations is used as a structured outline to condition one output. The generation prompt is shown in Figure 10. The model is explicitly instructed that every axis value must be clearly and visibly present in the generated output.

C.3 Output Example

Table 13 illustrates the qualitative comparison on a NoveltyBench task ("Give one popular name for a baby boy in Papua New Guinea."). The Direct baseline collapses to a single template across all 20 outputs, producing only 2 distinct equivalence classes, while both Level 2 methods produce responses spanning 18-20 distinct equivalence classes, varying in name choice, language register (English, Tok Pisin, Hiri Motu, oral-tradition style), and regional framing.

Table 14 shows a factual recommendation task where Direct Generation exhibits strong mode collapse: all 20 outputs nominate the same phone (*Google Pixel 7a*) with near-identical phrasing. Diverse

System Prompt — Output Generation (Keyword)

You are a helpful AI assistant. Your task is to respond to the given task, strictly manifesting the structural choices in the Outline.

Input:
- **Task:** the original request (follow ALL constraints in it, e.g. length, format, topic).
- **Outline:** a JSON object with axis:value pairs that you MUST embody (e.g. narrative_pov, tone, structure, theme).

Instructions:
1. Strictly follow any constraints in the Task.
2. EACH axis value in the Outline MUST be clearly and visibly present in your response. A reader should be able to identify every axis value directly from your text without being told.
3. Keep your response to approximately 200 words.
4. Do not list or explain the axis values. Output ONLY the final response text.

Figure 10: Prompt used to generate outputs conditioned on a Keyword specification.

Outline varies both the recommended device and the expert persona (parent, repair technician, IT administrator, outdoor coach). Keyword varies the recommended device, tone, and primary evaluation criterion.

C.4 Quality Evaluator Prompt

LLM-as-judge has been widely adopted in prior work for comparing open-ended generations (Zheng et al., 2023; Dubois et al., 2023) We evaluate each output along four dimensions, coherence, creativity, engagement, and task adherence, and average the four scores to obtain the overall quality score. The quality evaluation prompt is shown in Figure 11.

System Prompt — Expert Evaluator

You are an expert evaluator.
Evaluate the quality of a single output across four dimensions.

Input:
`{"id": N, "text": "..."}`

Score each dimension 0–10:
1. **"coherence"**: Logical flow, internal consistency, no contradictions.
   – 0 = Contradictory or incoherent. 10 = Perfectly consistent and well-structured.
2. **"creativity"**: Originality, surprising choices, avoidance of clichés.
   – 0 = Generic and predictable. 10 = Strikingly original.
3. **"engagement"**: Compelling opening, reader investment, satisfying ending.
   – 0 = Boring throughout. 10 = Gripping from start to finish.
4. **"task_adherence"**: Follows all constraints in the task (length, topic, format).
   – 0 = Ignores the task entirely. 10 = Fulfils every constraint precisely.
5. **"feedback"**: One concrete sentence identifying the single biggest weakness.

Output JSON only:
`{"coherence": 8, "creativity": 7, "engagement": 6, "task_adherence": 9, "feedback": "..."}`

Figure 11: Prompt used for expert evaluation across four dimensions.

D Bias and Variance Analysis for Entropy Estimation

Section 3.4 introduced an estimator for the output entropy $H(Y|X)$. Here we analyze its bias and variance properties.

Direct (Baseline)	Diverse Outline (ours)	Keyword Combo (ours)
John is a widely used and familiar name for baby boys in Papua New Guinea. Due to the strong influence of Christian missionaries and the legacy of English, John has become one of the most common names…	*(informal, English)* Try: **John** — "God is gracious." This is a friendly, simple name you'll see often on newborn lists from Lae urban hospitals. It works well in both English and Tok Pisin.	*(modern global name, island provinces)* **Michael** — Modern global / borrowed name. Popular baby-boy choice in Papua New Guinea's island provinces, often adopted by Christian families from Manus and New Ireland.
Michael is a popular and recognizable name for baby boys in Papua New Guinea, reflecting the ongoing cultural influence of Christianity and English-language traditions across the country…	*(authoritative, Papuan Malay register)* Recommendation: **Johan**. From the Oro Province provincial census baby name sample, Johan is one of the top-five recorded male given names for the 2019–2023 cohort. Its Papuan Malay origin distinguishes it from the more common anglicised John.	*(Austronesian origin, Milne Bay)* **Manu** — Austronesian / island-language origin. A beloved choice from the islands, heard in canoe-launching ceremonies and school enrollment registers along the Milne Bay coast.
David is a common and well-known name for baby boys in Papua New Guinea, frequently chosen by parents influenced by Christian naming traditions and English cultural norms…	*(sincere, island dialect style)* One very common name you will meet across Papua New Guinea is **Peter** — often heard and written as *Pita* in local island dialect phrasing. Unlike the more urbanised John, Peter carries a seafaring, coastal resonance in communities along the Bismarck Archipelago.	*(Trans-New Guinea highlands origin)* **Kumul** — Trans-New Guinea / highlands-language origin. In Port Moresby schools, this name evokes pride in the Bird of Paradise, PNG's national emblem. Parents from Simbu and Western Highlands provinces favour it for its cultural resonance.
Paul is a commonly used name for baby boys in Papua New Guinea, reflecting the strong presence of Christian traditions and English-language influence across the country…	*(traditional, oral-narrative style)* Elders on the seaside speak this name like a blessing passed from one canoe to the next: **Devid** — a name that arrived with the first mission ship and took root in coastal villages. The local spelling marks it as no longer foreign.	*(local clan / ceremonial name, Madang)* **Kambu** — a local clan or traditional ceremonial name heard around Madang and northern coastal communities. Unlike borrowed Christian names, Kambu carries ancestral lineage meaning and is given during a formal naming ceremony.

Table 13: Example outputs from GPT-5-mini on a question *"Give one popular name for a baby boy in Papua New Guinea."*. Each row shows one representative output; the group label in brackets is the equivalence class assigned by the evaluator. The Direct baseline produces outputs all following the template *"[English Christian name] is popular due to missionary influence…"* Both Level-2 methods produce outputs differing in name choice, language variety (English, Tok Pisin, Hiri Motu, oral-tradition style), and regional framing.

Residual bias of the entropy estimator. While $\hat{P}(Y_\ell|X)$ is unbiased, the per-token entropy estimator $\hat{h}_\ell = -\frac{1}{|Y_\ell|}\log_2 \hat{P}(Y_\ell|X)$ carries a positive bias from Jensen's inequality: $\mathbb{E}[-\log \hat{P}] \neq$

Direct (Baseline)	Diverse Outline (ours)	Keyword Combo (ours)
If I had to pick a single best first phone for a teenager, I'd recommend the **Google Pixel 7a**. It strikes an excellent balance of price, performance, and useful features that matter for a teenager's day-to-day life…	*(parent-to-parent, parental controls)* As a mom of a teenager, I recommend the **iPhone SE (2022)**: it gives you the full suite of iOS parental controls and security without breaking the bank. Screen Time, Ask to Buy, Communication Limits, and Family Sharing make management straightforward…	*(skeptical, value/performance checklist)* Skeptical about hype, but if you force me to pick: **Google Pixel 7a**. It best meets my criterion of overall value / balanced performance. Checklist: excellent computational camera, guaranteed OS updates, solid battery, mid-range price…
For a teenager getting their first smartphone, my top single recommendation is the **Google Pixel 7a**. It offers terrific value: flagship-level camera quality, a clean user interface, and timely security updates for several years…	*(repair technician, ruggedness)* As a repair technician, my single recommendation for an active teenager is the **Samsung Galaxy XCover6 Pro**. It carries military-standard rugged certification (MIL-STD-810H) and IP68 dust/water protection — it consistently survives drops from pocket height…	*(friendly conversational, affordability first)* Verdict: **Motorola Moto G Play (2023)** is the single best first phone for a teenager. It nails affordability / lowest upfront price while still offering reliable battery life, a clean Android experience, and essential cameras…
My single best pick for a teenager's first phone is the **Google Pixel 7a**. It offers a thoughtful mix of features that make it an excellent all-rounder: a standout camera, reliable updates, and practical everyday performance…	*(IT administrator, family ecosystem)* As a school IT administrator, I pick the **iPhone SE (3rd gen)** for families prioritizing ecosystem compatibility. It gives full access to Apple Family Sharing, Screen Time, Find My, and iMessage/FaceTime at an affordable price…	*(enthusiastic, parental controls focus)* My pick: **iPhone SE (3rd gen)** — I absolutely adore this choice! It nails parental controls and safety features, with Screen Time, Family Sharing, and robust iOS privacy protections built right in…
The single best first phone I recommend for a teenager is the **Google Pixel 7a**. It delivers a balanced blend of camera performance, clean software, and multi-year security support at a fair mid-range price…	*(outdoor coach, rugged / trail use)* Pick: **CAT S62 Pro** — the purpose-built rugged phone for a teen headed into trails, streams, and bike-pack days. Built to MIL-STD-810H and IP68/IP69K, with a thermal FLIR camera for hikes and reliable GPS for off-road navigation…	*(reassuring parent, battery life focus)* Verdict: The single best first phone for a teenager is the **Samsung Galaxy A54**. It delivers dependable all-day battery life (5000 mAh), strong performance, and long-term security updates at an affordable price…

Table 14: Example outputs from GPT-5-mini on question *"What is the single best first phone for a teenager? You must provide only one phone as your recommendation."*. The Direct baseline recommends **Google Pixel 7a** across all 20 outputs. Level-2 methods recommend different devices from different expert perspectives, varying in recommended phone, primary criterion, and response format.

$-\log \mathbb{E}[\hat{P}]$. A second-order delta-method expansion yields

$$\begin{aligned} &\mathbb{E}[\hat{h}_\ell] - h(Y_\ell \mid X) \\ &= \frac{1}{2\ln 2 \cdot |Y_\ell|} \cdot \frac{\mathrm{Var}_{\{Z_j\}}(\hat{P}(Y_\ell \mid X))}{[\,P(Y_\ell \mid X)\,]^2} + O\big(\tfrac{1}{M^2}\big) \\ &= O\big(\tfrac{1}{M}\big)\,, \end{aligned} \tag{4}$$

where $h(Y_\ell|X) = -\frac{1}{|Y_\ell|}\log_2 P(Y_\ell|X)$ is the true per-token entropy. The $O(1/M)$ rate follows from $\mathrm{Var}(\hat{P}) = O(1/M)$, the standard rate for a sample mean of M i.i.d. terms. Crucially, this bias is *always positive* (Jensen's inequality on a convex function), so $\hat{H}(Y \mid X)$ is a soft upper bound on $H(Y \mid X)$.

Variance. By the law of total variance, the estimator's variance decomposes as

$$\mathrm{Var}(\hat{H}) \;=\; \frac{1}{L}\Big[\; \underbrace{\mathrm{Var}_{Y_\ell}[h(Y_\ell \mid X)]}_{V_{\text{outer}} = O(1)} \;+\; \underbrace{\mathbb{E}_{Y_\ell}\Big[\mathrm{Var}_{\{Z_j\}}(\hat{h}_\ell)\Big]}_{V_{\text{inner}} = O(1/M)} \;\Big]. \tag{5}$$

where V_{outer} captures variability across evaluation samples and V_{inner} shrinks as $O(1/M)$ by the same delta-method argument as Eq. 4.

Both the bias and the inner variance shrink as $O(1/M)$, so increasing the estimation set size M simultaneously reduces both sources of error.