Read the Room Before Your Model Generates

Anonymous ACL submission

Abstract

Large language models (LLMs) generally rely on sampling temperature to regulate the amount of randomness in their outputs. Mainstream decoding strategies typically involve selecting temperature through fixed global settings and context or token-level calibration, rather than deriving it from a geometrically interpretable representation of the communicative scenario. In this work, we propose a new and simple temperature selection framework, the Scenario-Actor, grounded in the prior work in social psychology showing how human behavior is shaped by situation characteristics. Scenario-Actor extracts scenario-relevant structures from Transformer hidden states and determines the decoding temperature from scenario embedding, a structured geometric representation of the communicative settings. Empirical results shows the excellence of Scenario-Actor in open-ended instruction following and multi-turn naturalistic conversation compared to multiple baselines.

1 Introduction

Large language models (LLMs) generate text by repeatedly sampling from a conditional next-token distribution. An important hyperparameter is the sampling temperature, which controls the sharpness of the next-token probability distribution and, in turn, the rigorousness and creativity of model output.

Existing works mainly select the temperature from individual decoding units or predicted hyperparameter policies. Approaches such as adaptive decoding select temperature dynamically at token or example-level via LPO (Dhuliawala et al., 2024); contextual temperature infers temperature and other hyperparameters from the context (Wang et al., 2020).

But a prompt typically conveys more than semantics. It also implies a communicative setting: for example, whether the conversation is an academic discussion or a friendly gathering. Depending on different scenarios, desired responses are either rigorous or open-ended, cautious or imaginative, formal or casual. Prior work in social psychology provides us with a theoretical justification by introducing DIAMONDS (Rauthmann et al., 2014), a working model of situation characteristics where human behavior will be adjusted from.

We assume scenario is *a priori* at the prompt level and can be modeled mathematically, yet most widely used adaptive-temperature algorithms either do not consider them as a unified latent object or fail to distinguish the essence of context from scenario. In this work, we propose **Scenario-Actor**, a framework for scenario-adaptive temperature selection. We introduce a structured representation of the communicative setting extracted from context - the scenario embedding,[1] whose span composes the scenario space, and use it to derive the decoding temperature. Specifically, our method maps a representation from the r-th layer of the Transformer to an m-dimensional scenario space, where each prompt is represented by a scenario embedding. An Actout function then maps the geometry of this vector to a prompt-level decoding temperature.

We construct scenario axes from nine predefined scenario labels, and evaluate Scenario-Actor with fixed and random temperature policy, nucleus sampling (Holtzman et al., 2019), and EDT (Zhang et al., 2024) baselines. At the end, five scenario axes are constructed, corresponding to five of the eight DIAMONDS - Duty, Intellect, Adversity, Negativity, and Sociality (Rauthmann et al., 2014). Evaluations accross multiple baselines indicate Scenario-Actor's excellence in open-ended instruction following and multi-turn naturalistic conversation compared to most competitive decoding strategies.

[1] Namely, the scenario vector.

2 Background

Fixed decoding hyperparameters. Early work and much of engineering practice predominantly relied on manually-chosen global decoding strategies for task-specific models, such as beam search, top-k, and fixed sampling temperatures. Holtzman et al. (2019) showed that in open-ended text generation, likelihood-maximizing decoding methods often lead to bland, repetitive, and degenerate outputs, and proposed nucleus sampling (top-p) as a more effective alternative (Holtzman et al., 2019). Zhang et al. (2020) examined the trade-off between quality and diversity in different decoding strategies. Their findings indicate that all decoding methods perform similarly when diversity is a priority, but nucleus sampling (Holtzman et al., 2019) consistently outperforms all other evaluated algorithms when quality is viewed more important (Zhang et al., 2020). Wiher et al. (2022) revealed that the effectiveness of decoding strategies is highly task-dependent (Wiher et al., 2022). Later in LLM era, Shi et al. (2024) made a more systematic horizontal comparison among decoding methods and revealed that decoding method performance strongly depends on task, model size, alignment, and quantization (Shi et al., 2024).

Token-level dynamic temperature. Fixed decoding hyperparameters are not always optimal at every decoding step. An early token-level architecture begins with Chang et al., who proposed to relax the fixed-temperature assumption and adjust temperature dynamically according to its relevance to the source through KL-divergence (Chang et al., 2023). Zhu et al. then introduced AdapT sampling for code generation (Zhu et al., 2023). Zhang et al. further put forth EDT, which adjusts temperature directly from token-level entropy to better balance quality and diversity (Zhang et al., 2024).

Context and example-level temperature. Another line of work concerns with selecting sampling temperature from context. Wang et al. proposed contextual temperature, which learns a context-dependent temperature vector over the output vocabulary (Wang et al., 2020). Li et al. proposed DDS for open-domain chatting system. DDS copes with two situations - chit-chat and factual QA - and maps context-dependent diversity score to a single decoding temperature (Li et al., 2024). Dhuliawala et al. introduced adaptive decoding, which allows temperature to be predicted either once per example or dynamically at each decoding step (Dhuliawala et al., 2024).

3 The Scenario-Actor

Most modern LLMs are built using the celebrated Transformer architecture. In our framework, we assume that our prompt context is represented by the Transformer and take the r-th layer hidden-state matrix as input. The Scenario-Actor has two fundamental parts: Scenarize and Actout.[2] Scenarize takes the hidden-state matrix at the r-th layer Transformer as its input and outputs the **scenario embedding**; Actout takes a scenario embedding and maps it to a prompt-level decoding temperature T.

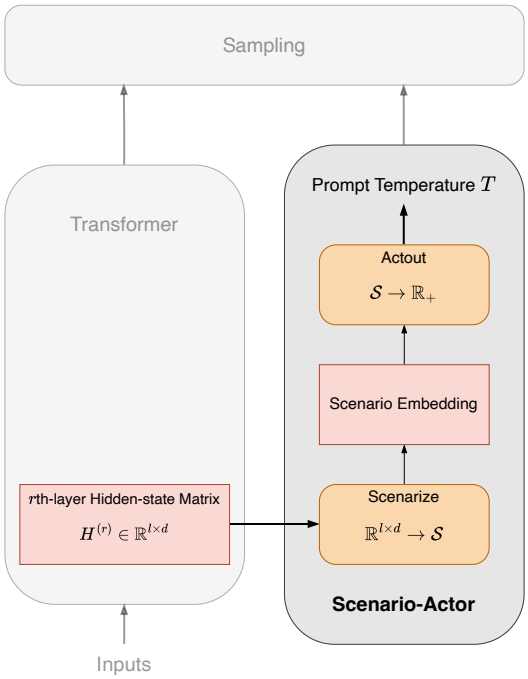

Figure 1: The Scenario-Actor - framework overview.

In the remainder of this section, we discuss these two modules in depth.

3.1 Scenarize

For a prompt $\mathcal{P} = \{p_1, \ldots, p_l\}$ with l tokens, let $H_{\mathcal{P}}^{(r)} = [h_{p_1}^{(r)}, \ldots, h_{p_l}^{(r)}]^T \in \mathbb{R}^{l \times d}$ denote the hidden-state matrix at the r-th Transformer layer, and $s_{\mathcal{P}} \in \mathcal{S} \subseteq \mathbb{R}^m$ denote the learned m-dimensional scenario embedding (alternatively,

[2]The name *Scenario-Actor* reflects our view that a prompt induces a latent scenario, and the model acts out (decode) from that scenario accordingly. As such, we name the two modules Scenarize and Actout.

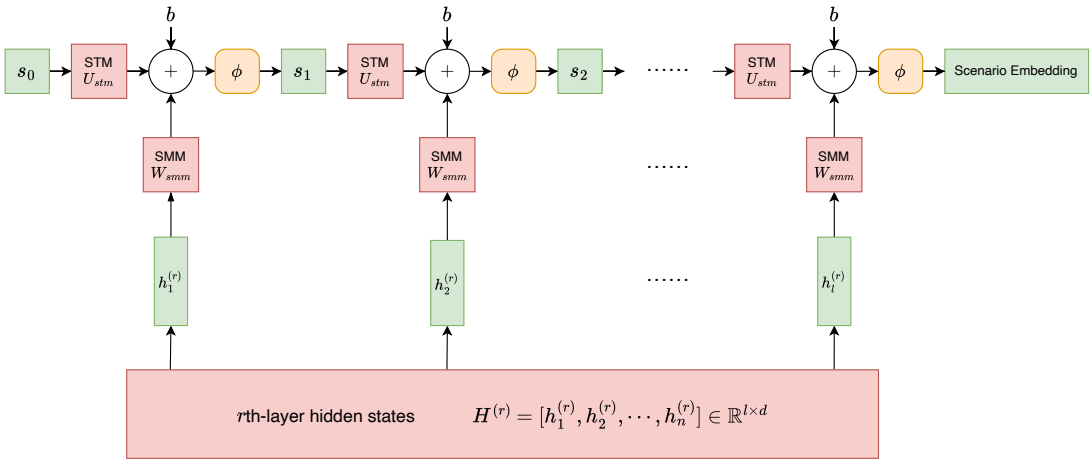

Figure 2: The Scenarize module.

the scenario vector) of prompt $\mathcal{P}$. The set $S := \left\{ s_{\mathcal{P}} \;\middle|\; s_{\mathcal{P}} = \text{Scenarize}(H_{\mathcal{P}}^{(r)}) \in \mathbb{R}^m \right\}$ be the set of all learned scenario embeddings. We define the scenario space as

$$\mathcal{S} := \text{span}(S) \subseteq \mathbb{R}^m.$$

Given that the language model interaction is text-only, the underlying prompt scenario is often under-determined with shorter linguistic sequence. As more tokens are observed, the scenario is generally expected to be more identifiable since longer prompt typically provides additional disambiguating context. As such, we model $\text{Scenarize}(H_{\mathcal{P}}^{(r)})$ as a recurrent neural network (RNN) that updates the inferred scenario token by token. The final state at p_l therefore represents the most informed scenario embedding which we will consider. Overall, we compute scenario embedding as

$$\begin{aligned} s_t &= \text{Scenarize}(h_{p_t}^{(r)}) \\ &= \phi(U_{\text{stm}} s_{t-1} + W_{\text{smm}} h_{p_t}^{(r)} + b) \end{aligned} \tag{1}$$

Here, $\phi(\cdot)$ is the activation function; $U_{\text{stm}} \in \mathbb{R}^{m \times m}$ is the scenario transition matrix (STM); $W_{\text{smm}} \in \mathbb{R}^{m \times d}$ is the scenario mapping matrix (SMM); and $b \in \mathbb{R}^m$ is the bias term. Each scenario vector is computed as the weighted addition of the previous token estimate and the current inference from context $h_{p_t}^{(r)} \in H_{\mathcal{P}}^{(r)}$, as depicted in Figure 2.

Both STM and SMM are expected to be trained on prompt triplets $(\mathcal{P}, \mathcal{P}^+, \mathcal{P}^-)$, where $\mathcal{P}$ denotes the prompt with base scenario, $\mathcal{P}^+$ the prompt with similar scenario, and $\mathcal{P}^-$ the prompt with opposite scenario. Given such a triplet, we define the loss function as

$$\mathcal{L} = \max(0, d(s_{\mathcal{P}}, s_{\mathcal{P}^+}) - d(s_{\mathcal{P}}, s_{\mathcal{P}^-}) + \beta) \tag{2}$$

which compares the distance from the base prompt scenario vector to the similar and opposite scenario vector with a margin of error β. In this work, we employ the squared Euclidean distance $d(s_{\mathcal{P}}, s_{\mathcal{P}^+}) = \|s_{\mathcal{P}} - s_{\mathcal{P}^+}\|^2$ and $d(s_{\mathcal{P}}, s_{\mathcal{P}^-}) = \|s_{\mathcal{P}} - s_{\mathcal{P}^-}\|^2$.

3.2 Actout

Actout maps a scenario embedding to a temperature. Drawing on *SemAxis* (An et al., 2018), we define scenario axes from two non-zero opposite scenario vectors s^+ and s^-. Each scenario axis u_i can be defined as $u_i = s_i^+ - s_i^- \in \mathbb{R}^m$. We name the scenario vectors on all scenario axes as **subscenario vectors**, each representing an elementary component of a more complex scenario. For all scenario axes, two typical temperatures τ_i^+ and τ_i^- on both the positive and negative directions are applied. In general, $u_1, \ldots, u_k$ have one-to-one correspondence with $(\tau_1^+, \tau_1^-), \ldots, (\tau_k^+, \tau_k^-) \in \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0}$.

Let $\rho_i = \hat{s}_{\mathcal{P}}^\top \hat{u}_i$ be the signed cosine similarity between the normalized scenario embedding and the axis. The prompt-level temperature T is then defined as

$$T = \frac{\sum_{i=1}^k |\rho_i| \left(\frac{1+\rho_i}{2} \tau_i^+ + \frac{1-\rho_i}{2} \tau_i^- \right)}{\sum_{i=1}^k |\rho_i|} \tag{3}$$

Temperature is defined as a normalized weighted average over k scenario axes. For each axis $\hat{u}_i$, the

cosine similarity ρ_i specifies where the scenario embedding lies along that axis. Endpoint temperatures τ_i^+ and τ_i^- are linearly interpolated based on ρ_i, so that the estimate moves towards τ_i^+ if ρ_i is positive and τ_i^- if negative. The denominator normalizes the weighted sum, eventually yielding a temperature inferred from the prompt's position in scenario space.

4 Experiment

We evaluate our proposed Scenario-Actor framework on several benchmarks, including structured reasoning, open-ended instruction following, and single and multi-turn conversations, to determine whether our framework has full potential to be an equally competitive decoding strategy.

4.1 Experimental Setup

4.1.1 Backbone model

Despite the outstanding performance of mainstream LLMs such as ChatGPT and Gemini, we do not have access to the hidden states with API calls. Therefore, We use Llama-3.1-8B-Instruct (Grattafiori et al., 2024) as our backbone model. The model is used for both hidden-state extraction and generation.

4.1.2 Data

We collect 37,357 conversational prompts from human-human and human-LLM interaction corpora, including WildChat (Zhao et al., 2024) and DailyDialog (Li et al., 2017), together with targeted supplementary datasets for under-represented scenario types. Each prompt is annotated by GPT-4o with one of nine scenario labels. A confidence-based filtering is applied to construct a high-confidence supervision set, resulting in 34,427 training prompts for triplet construction and 2,930 validation prompts.

4.1.3 Baselines

We compare Scenario-Actor against fixed-temperature decoding at $T \in \{0.5, 1.0, 1.5\}$, greedy decoding, random temperature selection, nucleus sampling (Holtzman et al., 2019) with $p = 0.95$, and entropy-driven temperature decoding (EDT) (Zhang et al., 2024).

4.1.4 Implementation details

Hidden states For the main experiment, we use the hidden-state matrix at the 16th Transformer layer; for ablations, we use the hidden-state matrix from the 30th layer.

Scenarize Scenarize maps the above hidden states into a 32-dimensional scenario embedding. The implementation follows Equation 1, where $r = 16$ is the Transformer layer, $\phi(\cdot) = \tanh(\cdot)$ is the activation function. Scenarize is trained according to Equation 2 with squared Euclidean triplet loss with margin $\beta = 0.3$. AdamW with learning rate 3.0e-4, weight decay 1.0e-4, gradient clipping at 1.0, and early stopping based on validation recall@1 is used as optimizer.

Actout In our experiment, we devise nine scenario labels, whose centroid as subscenario vectors to define each scenario axis and used GPT-4o (OpenAI, 2024) as an LLM annotator for training set. The nine scenario labels are rigorous, brainstorming, formal, playful, adversarial, institutional, casual_conversation, emotional_support_pos, and emotional_support_neg.

Five scenario axes - institutional vs. casual_conversation, rigorous vs. brainstorming, adversarial vs. non-adversarial; emotional_support_pos vs. emotional_support_neg, and formal vs. playful were constructed from combinations of nine subscenario vectors. Detailed description of each scenario can be found in Appendix **??**. Such five scenario axes match five of the DIMONDS: Duty, Intellect, Adversity, Negativity, and Sociality (Rauthmann et al., 2014).

Each of the five axis is assigned with two endpoint temperatures (τ_i^+, τ_i^-) as shown in Table 1:

Scenario Label	τ_i^+/τ_i^-
rigorous	0.30
institutional	0.50
formal	0.55
emotional_support_neg	0.65
adversarial	0.75
emotional_support_pos	1.00
casual_conversation	1.10
brainstorming	1.40
playful	1.50

Table 1: Scenario labels and their endpoint temperatures.

Actout then maps a scenario embedding generated from Scenarize to a temperature within the range $[0.01, 2]$ corresponding to Equation 3.

Baseline	GSM8K	AlpacaEval	MT-Bench
Metric	Accuracy (%)	LC-WR (%)	GPT-4 score
Scenario-Actor	65.28	**30.03**	**7.91**
Greedy	**77.18**	22.87	7.79
Fixed $T = 0.5$	69.52	23.63	7.79
Fixed $T = 1.0$	62.40	23.42	7.61
Fixed $T = 1.5$	45.19	8.60	6.76
EDT	69.14	25.58	6.74
Nucleus ($p = 0.95$)	61.33	23.59	7.63
Random-T	65.81	20.44	7.71

Table 2: Benchmark results comparing Scenario-Actor with fixed-temperature and adaptive decoding baselines. Best results are **highlighted**.

Task scenario Because current LLM prompts are often task-oriented rather than socially situated, many prompts do not contain enough linguistic evidence to infer a rich social scenario. We therefore define a task scenario as the operational mode induced by prompts when no inferrable social setting is available. For example, a math or coding problem is treated as a "rigorous" scenario because the output is expected to be more precise, while an idea-generation prompt is treated as "brainstorming" because the expected output is exploratory.

4.2 Main Result

We evaluate Scenario-Actor across five benchmarks covering mathematical reasoning, open-ended instruction following, multi-turn conversations, and naturalistic user prompts with respect to the baselines in experimental setup. Three benchmarks - GSM8K, AlpacaEval, and MT-Bench - were used for main evaluation. The performance is shown in Table 2. As shown in the two tables, Scenario-Actor outperforms most of the baseline decoding methods across all benchmarks in our experiment except for GSM8K.

Scenario-Actor reaches 65.28% accuracy on GSM8K, below greedy decoding, Fixed $T = 0.5$ and EDT, above nucleus sampling, fixed $T = 1.0$ and $T = 1.5$, but tied with Random-T. On AlpacaEval, Scenario-Actor achieve the strongest result among all tested methods with 30.03% Length-Controlled Win Rate (LC-WR), outperforming the best fixed-temperature baseline by 6.4%, EDT by 4.45%, and exceeding nucleus sampling, greedy decoding, and Random-T. On MT-Bench, our framework scores 7.91, which is above both greedy and fixed $T = 0.5$ decoding by 0.12, Random-T by 0.2, nucleus sampling by 0.28, and EDT by 1.17.

4.3 Analysis

Agreements To justify the use of LLM for scenario annotation and results judging, we compute both the LLM-human and human-human agreement on the models we used. A gold set on scenario annotation consisting of 306 prompts and a gold set on evaluation consisting of 50 AlpacaEval and MT-Bench respectively were manually annotated by us. The results show that the GPT-4o annotator agrees with each human annotator at $\kappa = 0.82$ and 0.85. The GPT-4-turbo (OpenAI, 2023) AlpacaEval judge reaches $\kappa \approx 0.72$ with each annotator. For MT-Bench scoring, GPT-4-turbo correlates with human raters at Pearson $r = 0.92$ and 0.86.

Validation	LLM-human	Human-human
Annotation κ	0.82 / 0.85	0.79
AE judge κ	0.72 / 0.72	0.60
MT Pearson r	0.92 / 0.86	0.87
MT Spearman ρ	0.84 / 0.70	0.79

Table 3: Agreement between LLM-based evaluators and human annotators. Detailed validation metrics are reported in Appendix E.

Geometry To validate if Scenarize learns meaningful scenario geometry, we visualize the learned scenario space using both PCA and UMAP. PCA shows the global variance structure of 900 scenario embeddings, with 100 examples from each scenario label. The first two principal components explain 27.9% and 21.8% of the variance respectively. As shown in Figure 3, nine centroids obtained from nine scenario labels occupy different regions in scenario space. PCA view also shows the five pro-

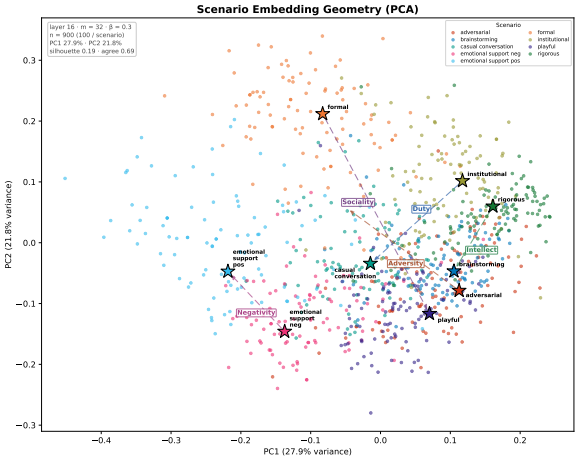

Figure 3: PCA of scenario embedding geometry.

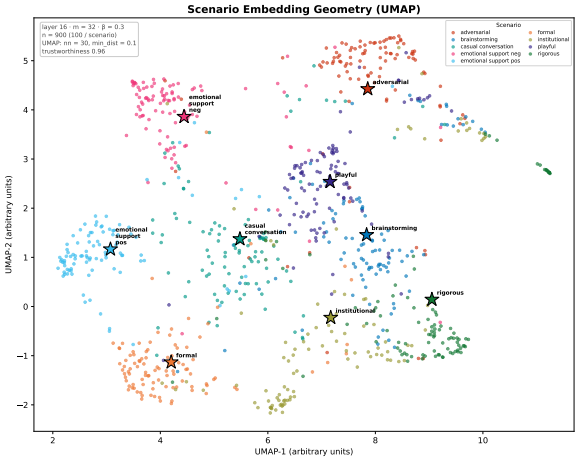

Figure 4: UMAP of scenario embedding geometry.

posed scenario axes used in our experiment - Duty, Intellect, Negativity, Sociality, and Adversity. The patterns therefore reflects that Scenarize learns a structured scenario space.

UMAP further confirms local structure of the learned space. The projection achieves a trustworthiness score of 0.96 using cosine distance. The embedding exihibits meaningful relationships among the nine scenario types. Semantically related scenarios also cluster near one another; for example, institutional prompts lie closer to rigorous prompts than to casual conversation.

Temperature GSM8K temperature is distributed within the range of $[0.8162, 0.8782]$; AlpacaEval is within $[0.7732, 1.0116]$, and MT-Bench is within $[0.7648, 1.0151]$.

In theory, GSM8K would exhibit better performance of the model when the sampling temperature is lower. In our experiment, Scenario-Actor selects a GSM8K temperature less than ideal. One likely contributor to the high temperature selection is source-scenario confounding in the training data. Ideally, each scenario label would be represented across multiple sources with comparable style and domain. But we find no available dataset covers all nine scenario labels with balanced coverage. As such, multiple supplementary dataset is used to fill the gaps of underrepresented labels, as shown in Appendix B. Scenarios and dataset style are therefore hard to be differentiated as certain scenario labels are dominated merely by one or few sources.

4.4 Ablation

Scenario-Actor benefits from longer context. We compare outputs generated by Scenario-Actor under short context with only one prior turn versus long context with all prior turns on 100 DailyDialog multi-turn conversations. Long context Scenario-Actor achieves a win rate of 67.0% over short context with a 95% bootstrap confidence interval of [58.0%, 76.0%] in pairwise GPT-4-turbo evaluation. The finding supports our claim that scenario is generally expected to be more identifiable with more context, which motivates the RNN formulation of Scenarize.

Layer 16 provides the best tradeoff for scenario probing. We observe that layer 16 produces stronger scenario axis separation and lower representation saturation than deeper layers such as layer 30. This observation aligns with recent work suggesting that structured semantic relation signals in LLMs often peak in intermediate layers rather than final generation-oriented layers (Diera and Scherp, 2026). Full layer-sweep diagnostics are reported in Appendix C.

Larger triplet margins induce geometry-performance tradeoffs. We compare the production margin $\beta = 0.3$ against $\beta = 1.0$ while holding the architecture, training data, and inference protocol fixed. Increasing the margin produces stronger centroid separation and improves GSM8K accuracy, but reduces performance on multi-turn and open-ended evaluation tasks. These results suggest that larger margins reshape the scenario geometry rather than uniformly improving it.

5 Conclusion

We present Scenario-Actora prompt-level temperature selection framework that maps hidden-state

matrix to a structured scenario embedding and derives a decoding temperature from its geometry along interpretable scenario axes. Experiments on multiple baselines and benchmarks shows the excellence of Scenario-Actor in open-ended instruction following and multi-turn naturalistic conversation. In addition, we see Scenario-Actor as a step toward socially adaptive language models. As LLMs become embedded in human society through multimodal agents and embodied systems, their ability to recognize the situation they are in will become increasingly important. Building on the social-psychological groundwork of DIAMONDS, we hope this framework would allow models to better understand social context, emotional nuance, and the expectations that shape human communication.

Limitations

LLM scenario is narrower than human scenario. Language models takes only textual sequence as its input. In actual human-human interaction, scenarios are dependent on cues other than linguistic context. Therefore, current LLMs are limited in its ability to fully integrate into all aspects of human interations. The inference of scenario solely from context is itself insufficient.

Task scenario is a compromise. In our dataset, we see a great amount of prompts being straightforward and utilitarian. Many prompt data ask the model to solve problems, write code, generate ideas, or provide information. Such a characteristic makes scenario inference almost impossible as task-oriented prompts often contain little evidence for a rich social setting. Due to such a limitation, we define task scenario as prompt-induced operational modes when no social scenario is available. This setup is also a compromise to current LLM usage as the language model itself is limited in its means of interaction. We expect this limitation to be alleviated in multimodal and future embodied agent settings, where models can access richer interactional signals and become more deeply integrated into human social environments.

Source-scenario confounding. A key limitation of the current training data is that scenario labels are not fully deconfounded from data source. In an ideal setting, each scenario category would be represented across multiple corpora so that Scenarize would be forced to learn merely the scenario structure rather than dataset style. In practice, however, no available dataset covers all of our scenario labels with balanced coverage. We therefore combined multiple supplementary sources. This improves coverage but introduces source-scenario correlation: some labels are dominated by one source, making it hard to separate scenario from dataset style. The scenario learning therefore requires more balanced training data and supervision in future work.

References

Jisun An, Haewoon Kwak, and Yong-Yeol Ahn. 2018. SemAxis: A lightweight framework to characterize domain-specific word semantics beyond sentiment. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2450–2461. Association for Computational Linguistics.

Chung-Ching Chang, David Reitter, Renat Aksitov, and Yun-Hsuan Sung. 2023. KL-divergence guided temperature sampling. *arXiv preprint arXiv:2306.01286*.

Shehzaad Dhuliawala, Ilia Kulikov, Ping Yu, Asli Celikyilmaz, Jason Weston, Sainbayar Sukhbaatar, and Jack Lanchantin. 2024. Adaptive decoding via latent preference optimization. *arXiv preprint arXiv:2411.09661*.

Andor Diera and Ansgar Scherp. 2026. Do language models encode semantic relations? probing and sparse feature analysis. *arXiv preprint arXiv:2603.17624*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of IJCNLP 2017*, pages 986–995.

Yiwei Li, Fei Mi, Yitong Li, Yasheng Wang, Bin Sun, Shaoxiong Feng, and Kan Li. 2024. Dynamic stochastic decoding strategy for open-domain dialogue generation. *arXiv preprint arXiv:2406.07850*.

OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

OpenAI. 2024. GPT-4o system card. *arXiv preprint arXiv:2410.21276*.

John Rauthmann, David Gallardo-Pujol, Esther Guillaume, Elysia Todd, Christopher Nave, Ryne Sherman, Matthias Ziegler, Ashley Jones, and David Funder. 2014. The situational eight diamonds: A taxonomy of major dimensions of situation characteristics. *Journal of Personality and Social Psychology*, 107(4).

Chufan Shi, Haoran Yang, Deng Cai, Zhisong Zhang, Yifan Wang, Yujiu Yang, and Wai Lam. 2024. A thorough examination of decoding methods in the era of LLMs. *arXiv preprint arXiv:2402.06925*.

Pei-Hsin Wang, Sheng-Iou Hsieh, Shih-Chieh Chang, Yu-Ting Chen, Jia-Yu Pan, Wei Wei, and Da-Chang Juan. 2020. Contextual temperature for language modeling. *arXiv preprint arXiv:2012.13575*.

Gian Wiher, Clara Meister, and Ryan Cotterell. 2022. On decoding strategies for neural text generators. *arXiv preprint arXiv:2203.15721*.

Hugh Zhang, Daniel Duckworth, Daphne Ippolito, and Arvind Neelakantan. 2020. Trading off diversity and quality in natural language generation. *arXiv preprint arXiv:2004.10450*.

Shimao Zhang, Yu Bao, and Shujian Huang. 2024. EDT: Improving large language models' generation by entropy-based dynamic temperature sampling. *arXiv preprint arXiv:2403.14541*.

Wayne Xin Zhao and 1 others. 2024. Wildchat: 1m chatgpt interaction logs in the wild. *arXiv preprint arXiv:2405.01470*.

Yuqi Zhu, Jia Li, Ge Li, YunFei Zhao, Jia Li, Zhi Jin, and Hong Mei. 2023. Hot or cold? adaptive temperature sampling for code generation with large language models. *arXiv preprint arXiv:2309.02772*.

A Artifact licenses and terms of use.

We use Llama-3.1-8B-Instruct, WildChat, DailyDialog, GPT-4o, and GPT-4-turbo, and previously proposed decoding baselines in accordance with their respective licenses or terms of use. The datasets are used only for research purposes, and no attempt is made to redistribute restricted model weights or proprietary model outputs beyond aggregate experimental results.

B Scenario Descriptions

Scenario Labeling	Description
rigorous	Interactions centered on correctness, logical structure, technical precision, or verifiable reasoning.
institutional	Interactions organized around explicit roles, responsibilities, procedures, or task objectives. The respondent should help resolve the issue efficiently.
formal	Interactions where etiquette, tact, face-saving, and socially careful wording are central.
emotional_support_neg	Interactions in which the user expresses distress, worry, sadness, frustration, vulnerability, or seeks comfort.
adversarial	Interactions involving disagreement, critique, challenge, defense, debate, or pressure to justify a position.
emotional_support_pos	Interactions in which the user shares positive affect, appreciation, celebration, affection, or seeks a warm affirming response.
casual_conversation	Interactions that are low-stakes, socially light, and not primarily task-, role-, or problem-driven.
brainstorming	Interactions that ask for idea generation, alternatives, exploration, or open-ended ideation.
playful	Interactions that are primarily humorous, teasing, and entertaining.

Table 4: Scenario labels and their corresponding communicative descriptions.

C Layer Sweep and Geometric Diagnostics

Layer	Val Recall@1	Effective Rank	Tanh Sat.	PCA@95%	Max Axis Offdiag	Collapse
12	0.793	25.15	0.64%	21	0.63	acceptable
16	0.770	24.41	4.33%	20	0.60	acceptable
20	0.798	22.80	15.42%	16	0.80	acceptable
24	0.739	18.21	50.70%	11	0.74	moderate
29	0.576	10.95	88.17%	6	0.99	severe
30	0.518	7.08	92.24%	4	0.94	severe

Table 5: Layer sweep diagnostics for scenario geometry. Deeper layers exhibit increasing saturation, rank collapse, and axis degeneracy.

D Source Composition

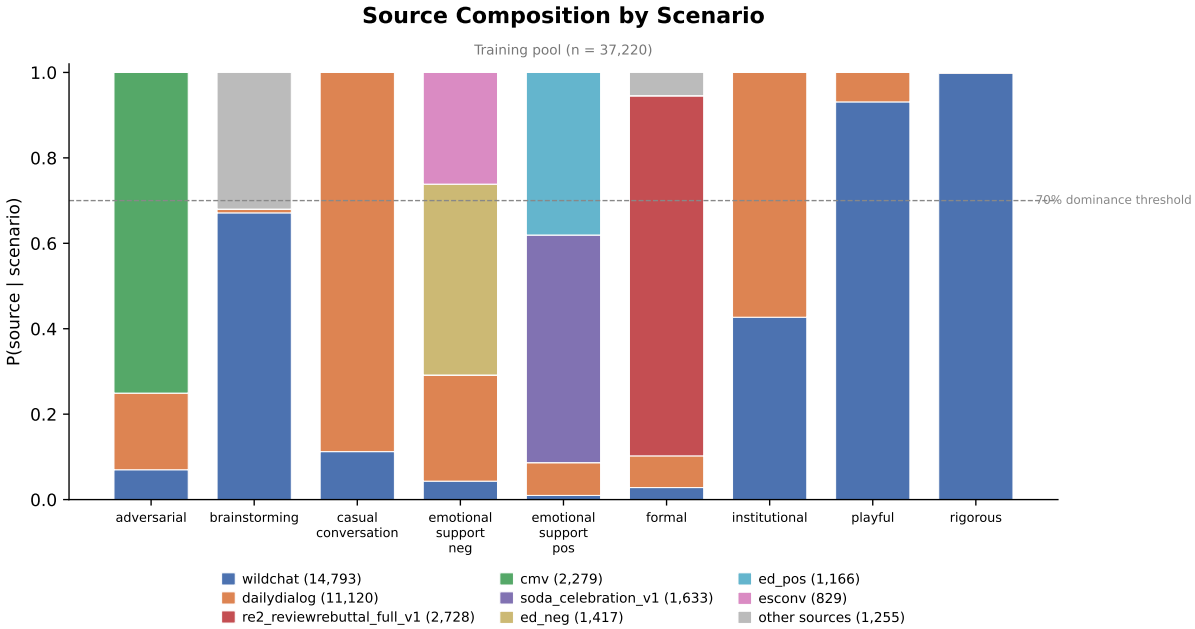

Figure 5: Source Composition by Scenario.

E Agreements

Metric	LLM-human	Human-human
Agreement (%)	83.66 / 86.27	81.37
Cohen's κ	0.82 / 0.85	0.79
Macro Precision	0.84	0.83
Macro Recall	0.84	0.80
Macro F1	0.83	0.81

Table 6: LLM annotator validation.

Metric	LLM-human	Human-human
AE Agreement (%)	86.00 / 86.00	80.00
AE Cohen's κ	0.72 / 0.72	0.60
MT Pearson r	0.92 / 0.86	0.87
MT Spearman ρ	0.84 / 0.70	0.79
MT MAE	0.44 / 0.90	0.78

Table 7: LLM judge validation.