Read the Room Before Your Model Generates

Muyao Kong[*,†]
New York University
muyao.kong@nyu.edu

Zhihao Lin[*]
Independent Researcher
zhihaolinsde@gmail.com

Abstract

Large language models (LLMs) generally rely on sampling temperature to regulate the amount of randomness in their outputs. Mainstream methods for selecting temperature typically involve fixed, heuristic global settings and context or token-level calibration, rather than a geometrically interpretable representation of the communicative scenario. In this work, we introduce a simple and interpretable temperature selection framework, the **Scenario-Actor**, which adapts decoding to the inferred situational characteristics. Scenario-Actor extracts scenario-relevant structures from Transformer hidden states and determines the temperature from scenario embedding, a structured geometric representation of the communicative setting. Empirical results show that on Llama-3.1-8B-Instruct, Scenario-Actor achieves strong goal-completion performance in interactive social scenarios, significant gains on open-ended instruction following, and competitive performance on multi-turn conversation compared to multiple baselines.[1]

1 Introduction

Large language models (LLMs) generate text by repeatedly sampling from a conditional next-token distribution. One of the simplest yet most consequential hyperparameters in language generation is the sampling temperature, which controls the sharpness of the next-token probability distribution and, in turn, the rigor and creativity of model output.

Existing systems typically set it globally or adapt it according to predictive uncertainty, context, or learned decoding objectives, which implicitly treat decoding primarily as a property of the model's predictive state (Dhuliawala et al., 2024; Wang et al., 2020). But socially situated language generation exposes a different source of variation: the interaction itself. A prompt conveys not only semantics but also implies a communicative setting. For example, the conversation could be either an academic discussion or a friendly chitchat, whose desired responses are either rigorous or open-ended, formal or casual. The optimal temperature is then circumstantial, varying across scenarios.

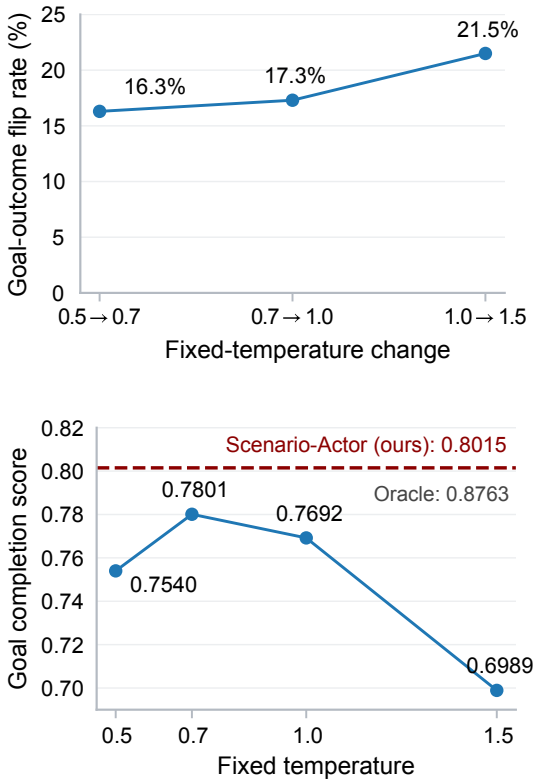


Figure 1: AgentSense goal-outcome flip rate (top) and goal completion score (bottom) by changing temperature.

In our controlled experiment, we decode 245 AgentSense (Mou et al., 2025) social interactions with four fixed temperatures. We found that first, changing temperature substantially alters interaction quality, moving between mid-range tempera-

[*]Equal contribution.
[†]Corresponding author.
[1]An interactive version of our paper is at https://muyao-kong.github.io/scenario-actor/.

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 1591–1607
November 6-10, 2026 ©2026 Association for Computational Linguistics

tures flips 16 to 17% of individual goal outcomes; second, globally strongest temperature setting does not dominate across interactions.

Such results show a major limitation of viewing temperature as a globally fixed hyperparameter, which motivates us to propose a way to control scenario-aware interaction and language generation. In this work, we propose **Scenario-Actor**, a framework for scenario-adaptive temperature selection. Scenario-Actor focuses on social interactions and open-ended instruction following, where the prompt provides enough evidence to infer a communicative scenario. We introduce a structured representation of the communicative setting extracted from context - the scenario embedding[2], whose span defines the scenario space, and use it to derive the sampling temperature. Specifically, our method maps a representation from the r-th layer of the Transformer to an m-dimensional scenario space, where each input is represented by a scenario embedding. An Actout function then maps the geometry of this vector to a prompt-level sampling temperature.

Prior work in social psychology provides us with a theoretical support by introducing DIAMONDS (Rauthmann et al., 2014), a working model of situation characteristics showing that human behavior varies with situation characteristics. We construct scenario axes from nine pre-defined scenario labels. 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 across fixed and random temperature policies, nucleus sampling (Holtzman et al., 2020), and EDT (Zhang et al., 2024) baselines indicate Scenario-Actor's competitive performance on interactive social scenarios, open-ended instruction following, and multi-turn dialogue compared to most decoding strategies.

2 Background

An autoregressive language model generates a response one token at a time. In each next-token generation, a trained neural network first produces a set of scores, which we call logits, from the prompt $\mathcal{P}$ and the generated tokens over the vocabulary $\mathcal{V}$ that represents all possible next tokens. A $\mathrm{Softmax}$ function then transforms these logits to a distribution that gives how likely a token is to be the next.

[2]In this work, we use the terms *scenario vector* and *scenario embedding* interchangeably.

Let y_t and $y_{<t}$ be the t-th token to be generated and already generated, and $z_{t,v}$ be the t-th logits of the token v in the vocabulary. Concretely, we have

$$\Pr(y_t = v \mid P, y_{<t}) = \frac{\exp(z_{t,v})}{\sum_{u \in \mathcal{V}} \exp(z_{t,u})}$$

Now, notice that if we divide each logit $z_{t,v}$ with a positive scalar T, the relative contribution of each numerator to the denominator will change with T. When $T > 1$, each $\exp(z_{t,v})$ becomes less dominant over $\sum_{u \in \mathcal{V}} \exp(z_{t,u})$, making the probabilities more evenly distributed. When $0 < T < 1$, each $\exp(z_{t,v})$ becomes more dominant. This scalar therefore controls the sharpness of the next-token distribution, which is commonly referred to as the temperature. Formally, for $T > 0$,

$$\Pr(y_t = v \mid P, y_{<t}) = \frac{\exp(\frac{z_{t,v}}{T})}{\sum_{u \in \mathcal{V}} \exp(\frac{z_{t,u}}{T})}$$

With all these, the last thing we need to do is to select a token through a process called sampling. Recent work center around language model's temperature and sampling strategies, which we discuss in the following paragraphs.

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. (2020) 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. Zhang et al. (2021) 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., 2020) consistently outperforms all other evaluated algorithms when quality is viewed as more important. Wiher et al. (2022) revealed that the effectiveness of decoding strategies is highly task-dependent. 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.

Token-level dynamic temperature. Fixed decoding hyperparameters are not always optimal at

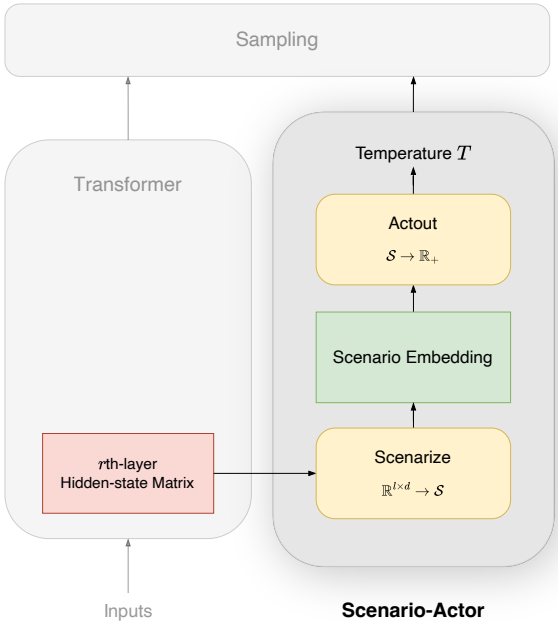


Figure 2: The Scenario-Actor - framework overview.

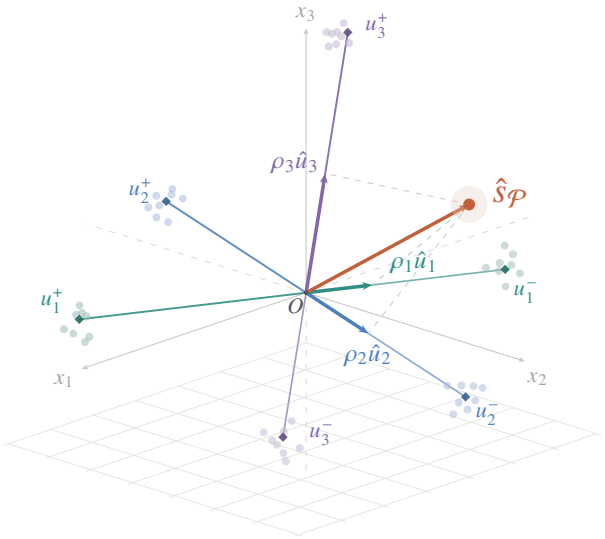


Figure 3: Visualization of scenario space and Actout.

every decoding step. An early token-level architecture begins with Chang et al. (2023), who proposed to relax the fixed-temperature assumption and adjust temperature dynamically according to its relevance to the source through KL-divergence. Zhu et al. (2024) then introduced AdapT sampling for code generation. Zhang et al. (2024) further put forth EDT, which adjusts temperature directly from token-level entropy to better balance quality and diversity.

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

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 a Transformer model and take the r-th layer hidden-state matrix as input. The Scenario-Actor has two fundamental parts: Scenarize and Actout.[3] Scenarize takes the hidden-state matrix at the r-th Transformer layer as its input and outputs the **scenario embedding**; Actout takes a scenario embedding and maps it to a sampling temperature T.

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

3.1 Scenarize

Before introducing our first module, let's establish some basic mathematical setup. 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 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 underdetermined with shorter linguistic sequences. As more tokens are observed, the scenario is generally expected to be more identifiable since longer prompts typically provide additional disambiguating context. As such, we model $\text{Scenarize}(H_\mathcal{P}^{(r)})$

[3]The module names *Scenarize* and *Actout* reflect our view that a prompt induces a latent scenario, and the model acts out, or decodes, from that scenario accordingly.

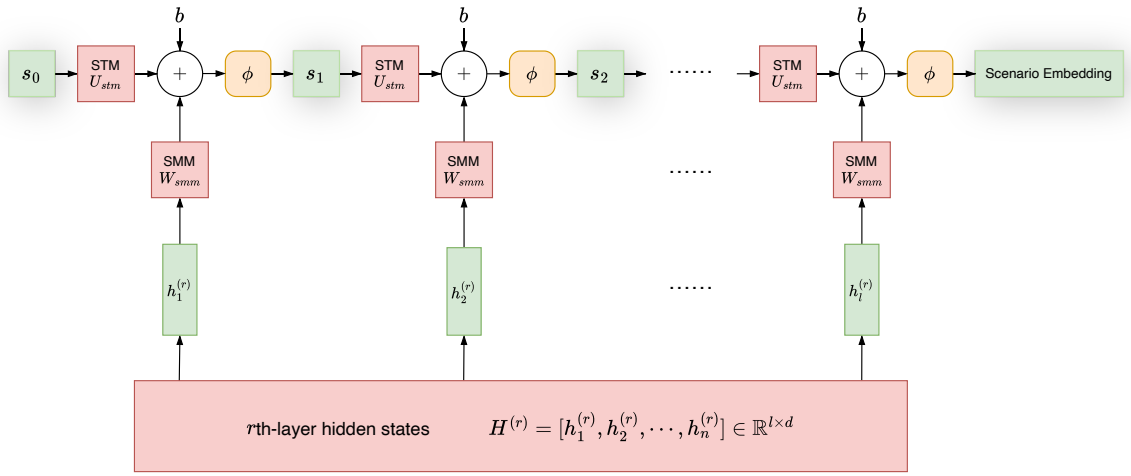


Figure 4: The Scenarize module.

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 learned affine combination of the previous token estimate and the current inference from context $h_{p_t}^{(r)} \in H_{\mathcal{P}}^{(r)}$, as depicted in Figure 4.

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 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 then 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. For $\sum_{i=1}^k |\rho_i| > 0$, the 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 benchmarks covering interactive social scenarios, open-ended instruction following and multi-turn question answering to test its performance on context-rich and socially-situated interactions. We also use GSM8K (Cobbe et al., 2021)

as a task-only stress test to study the boundary of scenario-based decoding.

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 the open-weight 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 (OpenAI, 2024) 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 $T \in \{0.5, 0.7, 1.0, 1.5\}$, greedy decoding, random temperature selection, nucleus sampling (Holtzman et al., 2020) with $p = 0.95$ and $T = 1.0$, and entropy-driven temperature decoding (EDT) (Zhang et al., 2024). EDT is included as a training-free and directly reproducible adaptive baseline. DDS and LPO (Li et al., 2024; Dhuliawala et al., 2024) both, in contrast, require training additional components. LPO trains its decoder module via preference optimization over sampled outputs, and DDS trains a diversity regressor designed for a two-way chit-chat vs. QA distinction, with, to our knowledge, no public implementation. Reproducing either method involves substantial heuristic design choices of our own, making the comparison hard to attribute in either direction.

4.1.4 Implementation details

Hidden states Llama-3.1-8B has 32 Transformer layers. For all experiments, we use the hidden states at the 16th Transformer layer. We found that scenario geometry collapses with deeper Transformer layer, which we discuss in Section 4.4.

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 We operationalize five DIAMONDS dimensions with nine scenario labels. We compute the centroid of each scenario label as a subscenario vector and use these vectors to define each scenario axes. GPT-4o (OpenAI, 2024) is used as an LLM annotator for the training set. The nine scenario labels are rigorous, brainstorming, formal, playful, adversarial, institutional, casual_conversation, emotional_support_pos, and emotional_support_neg.

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 2: Scenario labels and their endpoint temperatures.

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. Non-adversarial is the average over all subscenario vectors that are not adversarial. Detailed description of each scenario can be found in Appendix C. Such five scenario axes match five of the DIAMONDS: Duty, Intellect, Adversity, Negativity, and Sociality (Rauthmann et al., 2014).

Each of the five axes is assigned two endpoint temperatures (τ_i^+, τ_i^-), which are external hyperparameters of Actout, as shown in Table 2. Actout then maps a scenario embedding generated

Baseline	**AgentSense**	**AlpacaEval**	**MT-Bench**
Metric	Goal completion	LC-WR (%)	GPT-4 score
Scenario-Actor	**0.8015**	**30.03**	**7.91**
Greedy	0.7582	22.87	7.79
Fixed $T = 0.5$	0.7540	23.63	7.79
Fixed $T = 0.7$	0.7801	25.68	7.83
Fixed $T = 1.0$	0.7692	23.42	7.61
Fixed $T = 1.5$	0.6989	8.60	6.76
EDT	0.7671	25.58	6.74
Nucleus ($p = 0.95$)	0.7557	23.59	7.63
Random-T	0.7699	20.44	7.71

Table 1: Benchmark results comparing Scenario-Actor with fixed-temperature and adaptive decoding baselines. Best results are **highlighted**.

from Scenarize to a temperature within the range $[0.30, 1.50]$ corresponding to Equation 3. In our experiment, we fixed (τ_i^+, τ_i^-) for all benchmarks and comparisons.

4.2 Main Result

We evaluate Scenario-Actor across three benchmarks - AgentSense (Mou et al., 2025), AlpacaEval (Dubois et al., 2024), and MT-Bench (Zheng et al., 2023) - covering situational goal-completion, open-ended instruction following, and multi-turn interactions with respect to all baselines. Goal completion score, Length-Controlled Win Rate (LC-WR), and GPT-4 score were metrics for each of the three main evaluations. The performance is shown in Table 1. Scenario-Actor significantly outperforms most decoding methods on AgentSense and all baselines on AlpacaEval, while remaining competitive on MT-Bench in our experiment.

Among available benchmarks, AgentSense directly evaluates the setting targeted by Scenario-Actor: interactive dialogue in which the model must adapt to complete different goals as the social context develops. Scenario-Actor achieves the highest goal completion score of 0.8015, outperforming greedy decoding, every fixed-temperature setting, EDT, nucleus sampling, and Random-T. Scenario-Actor attains significant improvements over all baselines on the range from 3.16 to 10.26 percentage points except for fixed $T = 0.7$ at 0.7801, for which Scenario-Actor's 2.14 percentage points advantage is not statistically significant.

On AlpacaEval, Scenario-Actor achieves the strongest result among all tested methods with 30.03% Length-Controlled Win Rate (LC-WR), outperforming the best fixed-temperature baseline by 4.35 percentage points, EDT by 4.45 percentage points, and exceeding nucleus sampling, greedy decoding, Random-T, and class-conditioned T with statistical significance. On MT-Bench, our framework scores 7.91, which is significantly above fixed $T = 1.0$, fixed $T = 1.5$, Random-T, nucleus sampling, EDT, and class-conditioned T, and is potentially competitive among greedy decoding, fixed $T = 0.5$, and fixed $T = 0.7$. Detailed statistical report can be found in Appendix B.

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 examples and 50 MT-Bench examples were manually annotated by us.

Validation	LLM-human	Human-human
Annotation κ	0.82 / 0.85	0.79
AE Agreement	86.00 / 86.00	80.00
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
AS Agreement	80.00 / 72.00	76.00
AS Cohen's κ	0.48 / 0.27	0.40

Table 3: Agreement between LLM-based evaluators and human annotators. Values are rounded to two decimal places. Detailed validation metrics are reported in Appendix E.

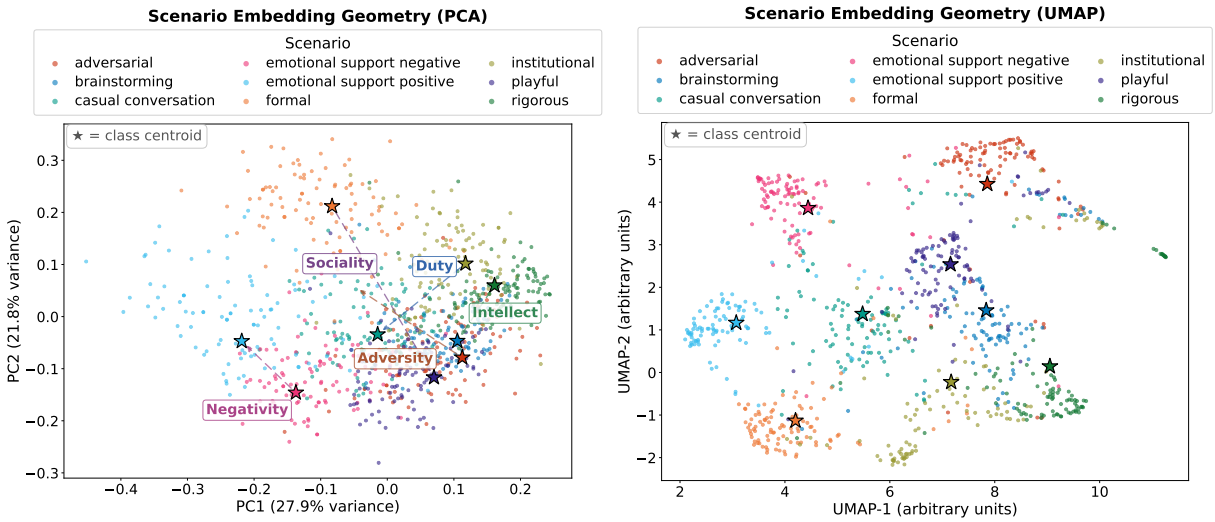


Figure 5: PCA (left) and UMAP (right) of scenario embedding geometry.

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, it correlates with human raters at Pearson $r = 0.92$ and 0.86; for AgentSense, it achieves agreement rates of 80% and 72% with the two human judges, comparable to the human-human agreement of 76%, and with Cohen's κ of 0.48 and 0.27, respectively.

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 5, nine centroids obtained from nine scenario labels occupy different regions in scenario space. PCA view also shows the five proposed scenario axes used in our experiment - Duty, Intellect, Negativity, Sociality, and Adversity. The patterns therefore reflect 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 exhibits 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.

GSM8K diagnosis To study the boundary of Scenario-Actor, we tested its performance on GSM8K against all other baselines. Scenario-Actor significantly underperforms greedy decoding, fixed $T = 0.5$, fixed $T = 0.7$, EDT, and insignificantly lower than Random-T baseline by 0.53.

Baseline	**GSM8K**
Metric	Accuracy (%)
Scenario-Actor	65.28
Greedy	**77.18**
Fixed $T = 0.5$	69.52
Fixed $T = 0.7$	69.07
Fixed $T = 1.0$	62.40
Fixed $T = 1.5$	45.19
EDT	69.14
Nucleus ($p = 0.95$)	61.33
Random-T	65.81

Table 4: GSM8K results comparing Scenario-Actor with fixed-temperature and adaptive decoding baselines. The best result is **highlighted**.

Our GSM8K temperature is distributed within the range of $[0.8162, 0.8782]$. 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. Nearest-centroid analysis shows that the distribution is concentrated in adversarial, institutional, and brainstorming scenarios, which together account for 77.9% of GSM8K prompts. Only 5.0% are mapped to the rigorous scenario.

Scenario	Count	Percentage
adversarial	350	26.5%
institutional	346	26.2%
brainstorming	332	25.2%
playful	79	6.0%
casual_conversation	67	5.1%
rigorous	66	5.0%
formal	40	3.0%
emotional_support_neg	23	1.7%
emotional_support_pos	16	1.2%

Table 5: GSM8K scenario count

All nine centroids look nearly identical to GSM8K prompts in cosine space:

Scenario Centroid	Cosine Similarity
adversarial	0.9765
institutional	0.9710
rigorous	0.9684
brainstorming	0.9510
playful	0.9499
casual_conversation	0.9463
emotional_support_neg	0.9191
formal	0.9098
emotional_support_pos	0.9043

Table 6: Mean cosine similarity between GSM8K and nine centroids, with gap of 0.0247 between the nearest and the second-nearest centroid per prompt

The results reflect near-equidistance of GSM8K to all scenario centroids: GSM8K prompts occupy a high-similarity region of the learned space. The current geometry does not reliably map task-only mathematical prompts to a low-temperature rigorous regime, and Scenario-Actor fails to infer the "scenario" of GSM8K prompts. The result illustrates the scope and boundary of Scenario-Actor to be a decoding framework for prompts whose communicative scenario can be inferred from context.

Temperature We examine the temperatures selected by Scenario-Actor across all benchmarks. Scenario-Actor operates within the range of $[0.67, 1.16]$ across turns generated by the evaluated Llama-3.1-8B-Instruct, $[0.77, 1.01]$ on AlpacaEval, and $[0.76, 1.02]$ on MT-Bench. AgentSense temperatures also change with dialogue context, with the mean temperature increasing by approximately 0.16 across turn positions.

Context length 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.

Efficiency We measure the latency, memory usage, and throughput of Scenario-Actor over 80 MT-Bench prompts on a single NVIDIA A10G 24 GB GPU (`ml.g5.2xlarge`). Latency includes hidden-state extraction, Scenarize, Actout, and response generation. As shown in Appendix A, Scenario-Actor has similar end-to-end latency and throughput to the tested decoding baselines. Latency is within 2.5% of greedy decoding, while the output and request throughput remain close to standard sampling methods. Peak allocated memory is also nearly unchanged.

4.4 Ablation

Geometry contributes more than endpoint temperature lookup. To test whether the gains come from the scenario geometry or from the manually-specified endpoint temperatures, we add a class-conditioned temperature baseline which uses the same Scenarize module but removes Actout's geometric weighting. Class-conditioned baseline treats the nearest centroid as the prompt's class label and assigns its sampling temperature from corresponding endpoint.

	Baseline	
Benchmark	Scenario-Actor	Class-conditioned T
AgentSense	**0.8015**	0.7536
AlpacaEval	**30.03**	27.10
MT-Bench	**7.91**	7.64

Table 7: Comparison with the class-conditioned temperature baseline. Metrics are goal completion score, LC-WR (%), and GPT-4 score respectively. Best results are **highlighted**.

Class-conditioned temperature baseline achieves a goal completion score of 0.7536 on AgentSense, LC-WR of 27.10% on AlpacaEval and a score of

7.64 on MT-Bench, which is lower than Scenario-Actor by 4.79 and 2.93 percentage points, and 0.27 points respectively. The improvements are statistically significant on all three benchmarks: AgentSense ($[+1.85, +7.88]$, $p = 1.85 \times 10^{-3}$), AlpacaEval ($[+0.55, +5.32]$, $p = 1.6 \times 10^{-2}$), and MT-Bench ($[+0.02, +0.52]$, $p = 3.21 \times 10^{-2}$). Since both methods use the same endpoint values, these results suggest the effectiveness of Actout's geometric interpolation compared to class-level temperature lookup.

Layer 16 provides the best tradeoff for scenario probing. To test which layer would be the optimal for scenario decoding, we evaluate several metrics across different Transformer layers. Four of which are **recall@1**, which computes how often the nearest retrieved embedding belongs to the same scenario; **effective rank**, measuring how many embedding dimensions are meaningfully used; **tanh saturation**, which calculates the percentage of embedding values close to -1 or 1; and **axis correlation** that gives the maximum correlation between scenario axes.

We observe that layer 16 produces competitive scenario retrieval performance, with stronger scenario axis separation and lower representation saturation than deeper layers such as layer 30, as shown in Table 8 below. 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 D.

Metric	Layer 16	Layer 20	Layer 30
Recall@1	0.770	**0.798**	0.518
Eff. Rank	**24.41**	22.80	7.08
tanh Sat.	**4.33%**	15.42%	92.24%
Axis Corr.	**0.60**	0.80	0.94

Table 8: Layer-sweep diagnostics. Best results are **highlighted**

Larger triplet margins induce geometry-performance tradeoffs. We compare the default margin $\beta = 0.3$ against $\beta = 1.0$ while holding the architecture, training data, and inference protocol fixed. As in Table 9, increasing 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 but does not uniformly improve generation quality.

Metric	$\beta = 0.3$	$\beta = 1.0$
Geometry		
Mean centroid cosine	0.9496	0.7103
Euclidean silhouette	0.1835	0.2668
Nearest-centroid agreement	0.6940	0.8877
Downstream		
GSM8K Accuracy (%)	65.28	66.41
AlpacaEval LC-WR (%)	30.03	25.54
MT-Bench	7.91	7.66
GSM8K mean T	0.822	0.738

Table 9: Effect of the triplet-loss margin on scenario geometry and downstream performance. Full table can be found in Appendix G.

5 Conclusion

We propose Scenario-Actor, a sampling temperature selection framework that maps a hidden-state matrix to a structured scenario embedding and derives a sampling temperature from its geometry along interpretable scenario axes. Comparisons of Scenario-Actor with multiple baselines show strong goal-completion performance on AgentSense, significant improvements on AlpacaEval, and competitive results on MT-Bench. 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 situations they are in will become increasingly important. We hope this framework allows models to better understand social context, emotional nuance, and the expectations that shape human communication in the future.

Limitations

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

Compromise on task-only prompts. ScenarioActor is designed for settings with rich scenario cues, such as multi-turn conversations. But in our dataset, we see a great amount of them being straightforward and utilitarian. Many prompts ask the model to directly solve problems such as writing code, answering math or providing general information. Such a characteristic makes scenario inference almost impossible as task-oriented prompts often contain little evidence for a social setting. For these prompts, a standalone selector can route the input to a standard task-specific decoding policy, which is independent of our framework. This setup is also a compromise to current LLM usage as the language model itself is limited in its interaction modality. We expect this limitation to be alleviated in multimodal or embodied agent settings, where models can access richer interactional cues and become more deeply integrated into human social environments.

Model and taxonomy scope. The current study establishes the results and claims merely on Llama-3.1-8B-Instruct with the nine-label, five-axis scenario taxonomy. In theory, however, our framework is expected to be applicable to other model families and scenario taxonomies, which, due to limited budget and computational power, require more evaluations.

Source balance. A key limitation of the current training data is that scenario labels are not fully balanced 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 rely on multiple supplementary sources. Developing a more balanced scenario dataset remains an important direction for future work.

Acknowledgments

We thank Professor Haifeng Xu at the University of Chicago and the anonymous reviewers for their valuable comments, as well as Professor Adam Meyers at New York University for his advice on publication. This work was conducted independently by the authors and received no institutional research funding by the time of submission.

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, Melbourne, Australia. 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*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

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. In *Proceedings of the Fifteenth Language Resources and Evaluation Conference*, pages 2115–2126, Palma de Mallorca, Spain. ELRA Language Resource Association.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. 2024. Length-controlled AlpacaEval: A simple way to debias automatic evaluators. In *Conference on Language Modeling (COLM)*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Neha Nayak Kennard, Tim O'Gorman, Rajarshi Das, Akshay Sharma, Chhandak Bagchi, Matthew Clinton, Pranay Kumar Yelugam, Hamed Zamani, and Andrew McCallum. 2022. DISAPERE: A dataset for discourse structure in peer review discussions. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1234–1249, Seattle, United States. Association for Computational Linguistics.

Hyunwoo Kim, Jack Hessel, Liwei Jiang, Peter West, Ximing Lu, Youngjae Yu, Pei Zhou, Ronan Le Bras,

Malihe Alikhani, Gunhee Kim, Maarten Sap, and Yejin Choi. 2023. SODA: Million-scale dialogue distillation with social commonsense contextualization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12930–12949, Singapore. Association for Computational Linguistics.

Bryan Klimt and Yiming Yang. 2004. The Enron corpus: A new dataset for email classification research. In *Machine Learning: ECML 2004*, volume 3201, pages 217–226. Springer.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. 2023. OpenAssistant conversations – democratizing large language model alignment. In *Advances in Neural Information Processing Systems*, volume 36, pages 47669–47681.

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 the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995, Taipei, Taiwan. Asian Federation of Natural Language Processing.

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. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11585–11596, Bangkok, Thailand. Association for Computational Linguistics.

Siyang Liu, Chujie Zheng, Orianna Demasi, Sahand Sabour, Yu Li, Zhou Yu, Yong Jiang, and Minlie Huang. 2021. Towards emotional support dialog systems. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3469–3483, Online. Association for Computational Linguistics.

Xinyi Mou, Jingcong Liang, Jiayu Lin, Xinnong Zhang, Xiawei Liu, Shiyue Yang, Rong Ye, Lei Chen, Haoyu Kuang, Xuanjing Huang, and Zhongyu Wei. 2025. AgentSense: Benchmarking social intelligence of language agents through interactive scenarios. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4975–5001, Albuquerque, New Mexico. Association for Computational Linguistics.

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

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

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381, Florence, Italy. Association for Computational Linguistics.

John F. Rauthmann, David Gallardo-Pujol, Esther M. Guillaume, Elysia Todd, Christopher S. Nave, Ryne A. Sherman, Matthias Ziegler, Ashley Bell Jones, and David C. Funder. 2014. The situational eight DIAMONDS: A taxonomy of major dimensions of situation characteristics. *Journal of Personality and Social Psychology*, 107(4):677–718.

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. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8601–8629, Miami, Florida, USA. Association for Computational Linguistics.

Chenhao Tan, Vlad Niculae, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. 2016. Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions. In *Proceedings of the 25th International Conference on World Wide Web*, pages 613–624, Montréal, Québec, Canada. International World Wide Web Conferences Steering Committee.

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. *Transactions of the Association for Computational Linguistics*, 10:997–1012.

Daoze Zhang, Zhijian Bao, Sihang Du, Zhiyi Zhao, Kuangling Zhang, Dezheng Bao, and Yang Yang. 2025. Re2: A consistency-ensured dataset for full-stage peer review and multi-turn rebuttal discussions. *arXiv preprint arXiv:2505.07920*.

Hugh Zhang, Daniel Duckworth, Daphne Ippolito, and Arvind Neelakantan. 2021. Trading off diversity and quality in natural language generation. In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 25–33, Online. Association for Computational Linguistics.

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*.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024. WildChat: 1M

ChatGPT interaction logs in the wild. In *The Twelfth International Conference on Learning Representations*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623.

Yuqi Zhu, Jia Li, Ge Li, YunFei Zhao, Jia Li, Zhi Jin, and Hong Mei. 2024. Hot or cold? Adaptive temperature sampling for code generation with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 437–445.

A Efficiency

Method	Latency (ms/episode)	Peak memory (GiB)		Throughput	
		Allocated	Reserved	Output (tokens/s)	Request (episodes/s)
Scenario-Actor	97,000	18.59	18.71	21.59	0.01031
Greedy	87,410	18.36	18.48	21.84	0.01144
Fixed $T = 0.5$	88,110	18.38	18.50	21.57	0.01135
Nucleus ($p = 0.95$)	88,370	18.38	18.50	21.49	0.01132
Random-T	87,340	18.38	18.50	21.59	0.01145
EDT	87,650	18.38	18.55	21.64	0.01141

Table 10: End-to-end efficiency over 245 AgentSense templates in 15-turn closed-loop interactions.

Method	Latency (ms/prompt)	Peak memory (GiB)		Throughput	
		Allocated	Reserved	Output (tokens/s)	Request (prompts/s)
Scenario-Actor	17,109	16.31	16.68	25.35	0.0584
Greedy	17,181	16.52	16.86	25.87	0.0582
Fixed $T = 0.5$	17,150	16.52	16.86	25.43	0.0583
Nucleus ($p = 0.95$)	17,406	16.52	16.86	25.42	0.0575
Random-T	17,284	16.52	16.86	25.42	0.0579
EDT	17,336	16.52	16.86	25.59	0.0577

Table 11: End-to-end efficiency over 805 AlpacaEval prompts.

Method	Latency (ms/prompt)	Peak memory (GiB)		Throughput	
		Allocated	Reserved	Output (tokens/s)	Request (prompts/s)
Scenario-Actor	16,917	15.22	15.54	25.26	0.0591
Greedy	16,513	15.19	15.26	25.75	0.0606
Fixed $T = 0.5$	17,128	15.19	15.54	25.24	0.0584
Nucleus ($p = 0.95$)	16,845	15.19	15.26	25.29	0.0594
Random-T	16,267	15.19	15.26	25.28	0.0615
EDT	16,957	15.24	21.99	25.41	0.0590

Table 12: End-to-end efficiency over 80 MT-Bench prompts.

B Statistical Significance Testing

Benchmark	Comparison	Statistical significance		
		Difference	95% CI	p-value
AgentSense (Δ Goal completion)	Scenario-Actor vs Greedy	+4.33 pp.	$[+1.55, +7.15]$	2.44×10^{-3}
	Scenario-Actor vs Fixed $T = 0.5$	+4.75 pp.	$[+1.82, +7.76]$	1.72×10^{-3}
	Scenario-Actor vs Fixed $T = 0.7$	+2.14 pp.	$[-0.54, +4.89]$	1.22×10^{-1}
	Scenario-Actor vs Fixed $T = 1.0$	+3.23 pp.	$[+0.41, +6.11]$	2.63×10^{-2}
	Scenario-Actor vs Fixed $T = 1.5$	+10.26 pp.	$[+6.97, +13.53]$	8.73×10^{-10}
	Scenario-Actor vs EDT	+3.44 pp.	$[+0.50, +6.47]$	2.39×10^{-2}
	Scenario-Actor vs Nucleus	+4.58 pp.	$[+1.41, +7.70]$	4.31×10^{-3}
	Scenario-Actor vs Random-T	+3.16 pp.	$[+0.40, +5.97]$	2.62×10^{-2}
	Scenario-Actor vs Class-conditioned T	+4.79 pp.	$[+1.85, +7.88]$	1.85×10^{-3}
AlpacaEval (Δ LC-WR)	Scenario-Actor vs Greedy	+7.16	$[+3.14, +11.18]$	5×10^{-4}
	Scenario-Actor vs Fixed $T = 0.5$	+6.40	$[+2.48, +10.32]$	1.4×10^{-3}
	Scenario-Actor vs Fixed $T = 0.7$	+4.35	$[+0.43, +8.27]$	2.96×10^{-2}
	Scenario-Actor vs Fixed $T = 1.0$	+6.61	$[+2.89, +10.33]$	5×10^{-4}
	Scenario-Actor vs Fixed $T = 1.5$	+21.43	$[+16.63, +26.23]$	2.2×10^{-18}
	Scenario-Actor vs EDT	+4.45	$[+0.43, +8.47]$	3×10^{-2}
	Scenario-Actor vs Nucleus	+6.44	$[+2.62, +10.26]$	1×10^{-3}
	Scenario-Actor vs Random-T	+9.59	$[+5.28, +13.90]$	1.3×10^{-5}
	Scenario-Actor vs Class-conditioned T	+2.93	$[+0.55, +5.32]$	1.6×10^{-2}
MT-Bench (Δ Score)	Scenario-Actor vs Greedy	+0.12	$[-0.11, +0.35]$	3.09×10^{-1}
	Scenario-Actor vs Fixed $T = 0.5$	+0.12	$[-0.12, +0.36]$	3.29×10^{-1}
	Scenario-Actor vs Fixed $T = 0.7$	+0.08	$[-0.13, +0.29]$	4.63×10^{-1}
	Scenario-Actor vs Fixed $T = 1.0$	+0.30	$[+0.05, +0.55]$	1.91×10^{-2}
	Scenario-Actor vs Fixed $T = 1.5$	+1.15	$[+0.79, +1.51]$	3.3×10^{-10}
	Scenario-Actor vs EDT	+1.17	$[+0.80, +1.54]$	9.03×10^{-10}
	Scenario-Actor vs Nucleus	+0.28	$[+0.05, +0.51]$	1.86×10^{-2}
	Scenario-Actor vs Random-T	+0.20	$[+0.01, +0.39]$	4.13×10^{-2}
	Scenario-Actor vs Class-conditioned T	+0.27	$[+0.02, +0.52]$	3.21×10^{-2}
GSM8K (Δ Accuracy)	Scenario-Actor vs Greedy	−11.90	$[-14.71, -9.02]$	6.1×10^{-16}
	Scenario-Actor vs Fixed $T = 0.5$	−4.25	$[-7.13, -1.36]$	4.7×10^{-3}
	Scenario-Actor vs Fixed $T = 0.7$	−3.79	$[-6.80, -0.80]$	1.2×10^{-2}
	Scenario-Actor vs Fixed $T = 1.0$	+2.88	$[-0.23, +5.99]$	7.5×10^{-2}
	Scenario-Actor vs Fixed $T = 1.5$	+20.09	$[+16.83, +23.35]$	2.5×10^{-31}
	Scenario-Actor vs EDT	−3.87	$[-6.90, -0.83]$	1.5×10^{-2}
	Scenario-Actor vs Nucleus	+3.94	$[+0.76, +7.20]$	1.7×10^{-2}
	Scenario-Actor vs Random-T	−0.53	$[-3.79, +2.73]$	7.8×10^{-1}

Table 13: Statistical significance testing results on AgentSense, AlpacaEval, MT-Bench, and GSM8K. Differences are computed as Scenario-Actor minus the corresponding baseline.

C Scenario Descriptions

Scenario Label	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 14: Scenario labels and their corresponding communicative descriptions.

D Layer Sweep and Geometric Diagnostics

Layer	Recall@1	Effective Rank	tanh Sat.	PCA@95%	Max Axis Offdiag	Collapse
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 15: Layer sweep diagnostics for scenario geometry. Collapse severity is assessed jointly from tanh saturation, effective rank, and axis off-diagonal correlation. Deeper layers exhibit increasing saturation, rank collapse, and axis degeneracy.

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 16: LLM annotator validation, values are rounded to two decimal places

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
AS Agreement (%)	80.00 / 72.00	76.00
AS Cohen's κ	0.48 / 0.27	0.40

Table 17: LLM judge validation, values are rounded to two decimal places.

F Source Composition

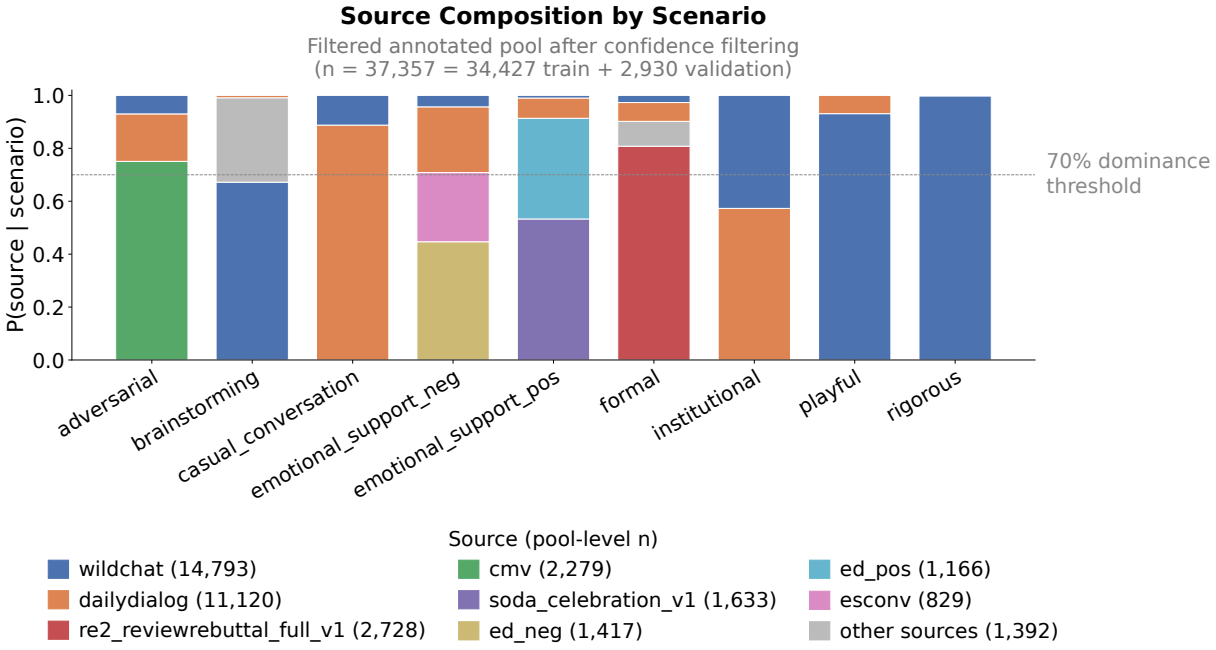


Figure 6: Source composition by scenario.

G Triplet-Margin Ablation

Geometry metric	$\beta = 0.3$	$\beta = 1.0$	Δ
Mean pairwise centroid cosine	0.9496	0.7103	-0.2393
Centered centroid cosine	0.5258	0.5057	-0.0201
Centroid effective rank at 95%	5	6	$+1$
Euclidean silhouette	0.1835	0.2668	$+0.0833$
Nearest-centroid agreement	0.6940	0.8877	$+0.1937$
tanh saturation (> 0.95)	0.0938	0.0347	-0.0591

Table 18: Scenario geometry under two triplet-loss margins.

Benchmark	**Metric**	$\beta = 0.3$	$\beta = 1.0$	Δ
GSM8K	Accuracy (%)	65.28	66.41	$+1.13$
AlpacaEval	LC-WR (%)	30.03	25.54	-4.49
MT-Bench	GPT-4 score	7.91	7.66	-0.25
DailyDialog NextTurn	WR (%)	49.70	46.94	-2.76
DailyDialog ContextLen	WR (%)	67.00	62.00	-5.00
Diagnostic				
GSM8K	Mean selected T	0.822	0.738	-0.084

Table 19: Downstream performance under two triplet-loss margins.

H Computational Details

We keep the backbone model frozen and train only the Scenarize module, which maps the 4096-dimensional Llama hidden states into a 32-dimensional scenario embedding. All experiments run on a single NVIDIA A10G GPU with 24GB memory. The total computational budget, including hidden-state extraction, Scenarize training, layer sweeps, and evaluation runs, is approximately 125 A10G GPU-hours. This estimate excludes external API inference costs used for GPT-4o annotation and GPT-4-turbo evaluation.

I Artifact licenses and terms of use.

We use Llama-3.1-8B-Instruct, WildChat, DailyDialog, ChangeMyView (Tan et al., 2016), SODA (Kim et al., 2023), EmpatheticDialogues (Rashkin et al., 2019), ESConv (Liu et al., 2021), OpenAssistant (Köpf et al., 2023), Re^2 (Zhang et al., 2025), DISAPERE (Kennard et al., 2022), OpenReview review text[4], Enron email corpus (Klimt and Yang, 2004), GPT-4o, and GPT-4-turbo, and previously proposed decoding baselines in accordance with their respective licenses or terms of use. All 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.

[4] https://openreview.net