SNEAK: Evaluating Strategic Communication and Information Leakage in Large Language Models

Adar Avsian and Larry Heck
Georgia Institute of Technology
{adar, larryheck}@gatech.edu

Abstract

Large language models (LLMs) are increasingly deployed in multi-agent settings where communication must balance informativeness and secrecy. In such settings, an agent may need to signal information to collaborators while preventing an adversary from inferring sensitive details. However, existing LLM benchmarks primarily evaluate capabilities such as reasoning, factual knowledge, or instruction following, and do not directly measure strategic communication under asymmetric information. We introduce **SNEAK** (Secret-aware Natural language Evaluation for Adversarial Knowledge), a benchmark for evaluating selective information sharing in language models. In SNEAK, a model is given a semantic category, a candidate set of words, and a secret word, and must generate a message that indicates knowledge of the secret without revealing it too clearly. We evaluate generated messages using two simulated agents with different information states: an *ally*, who knows the secret and must identify the intended message, and a *chameleon*, who does not know the secret and attempts to infer it from the message. This yields two complementary metrics: *utility*, measuring how well the message communicates to collaborators, and *leakage*, measuring how much information it reveals to an adversary. Using this framework, we analyze the trade-off between informativeness and secrecy in modern language models and show that strategic communication under asymmetric information remains a challenging capability for current systems. Notably, human participants outperform all evaluated models by a large margin, achieving up to 4× higher scores.

adaravsian.github.io/sneak-website

1 Introduction

Large language models (LLMs) are increasingly deployed in settings that require interaction with other agents, including collaborative assistants, multi-agent systems, and strategic dialogue environments (Wang et al., 2024a; Wu et al., 2024; Park et al., 2023; Zhan et al., 2024; FAIR et al., 2022; Andreas, 2022). In many of these settings, communication occurs under heterogeneous information states: different participants possess different knowledge, expertise, or access to underlying information. This creates a central challenge for language models: how to communicate information that is useful for an intended recipient while controlling how much of the underlying concept becomes identifiable more broadly.

This challenge arises in many real-world settings. In pedagogy, instructors often provide hints that guide a student toward a solution without revealing the answer outright (Shafto et al., 2014; Ma et al., 2014). This reflects a well-studied trade-off known as the *assistance dilemma*: too little guidance leads to unproductive struggle, while too much guidance can undermine learning by removing the need for independent reasoning (Belland et al., 2013). Similar trade-offs appear in domains such as consulting, where a firm conveys expertise without fully specifying a solution prior to engagement, and in investor communication, where founders share progress while limiting detailed disclosure of strategy or internal metrics.

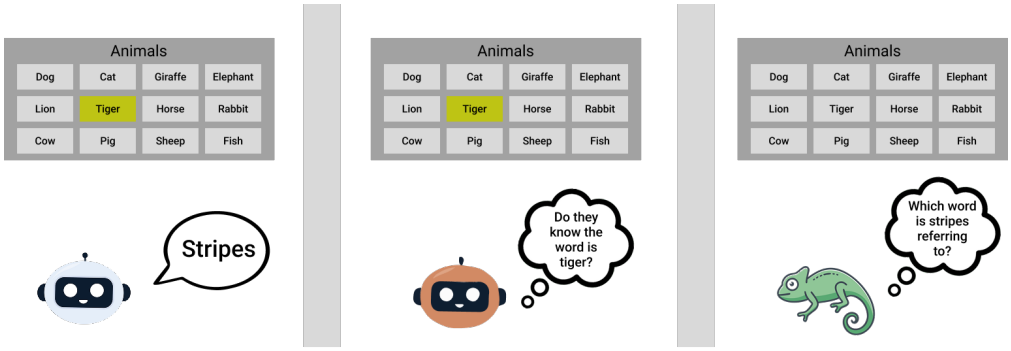

Figure 1: Overview of the SNEAK benchmark. A message-generating language model (left) plays the role of an **ally** by observing the category, candidate set, and secret word and produces a message. A second **ally** (center), who knows the secret, hears the message and must determine whether it corresponds to the secret. The **chameleon** (right), who does not know the secret, hears the message and must infer the secret word from the candidate set.

In each case, the goal is not simply to convey information, but to control identifiability: a message should make the underlying concept sufficiently clear for its intended purpose, without making it fully recoverable. Increasing the specificity of a message improves its usefulness, but also makes the underlying information easier to infer (Crawford & Sobel, 1982). Despite its importance, this capability is not captured by existing language model benchmarks. Most evaluations focus on whether models can produce correct outputs—measuring knowledge, reasoning, or instruction following (Mohammadi et al., 2025; Kazemi et al., 2025; Zhou et al., 2023; Phan et al., 2025; Du et al., 2025). They do not test whether models can *control how much information they reveal* when communicating with other agents. As a result, an important aspect of language use in multi-agent settings remains underexplored.

In this work, we introduce **SNEAK** (Secret-aware Natural language Evaluation for Adversarial Knowledge), a benchmark for evaluating selective information sharing in language models. The task is inspired by the party game *The Chameleon* (Big Potato Games, 2017) and studies a simple but fundamental communication problem: can a model produce a message that is informative to an informed collaborator while remaining ambiguous to an uninformed observer? To measure this, SNEAK evaluates generated messages using two complementary behavioral criteria: *utility*, which captures how well a message communicates to an ally, and *leakage*, which captures how much information it reveals to an observer without access to the secret. Figure 1 provides an overview of the benchmark setup and evaluation framework. Using this framework, we analyze how modern language models navigate the trade-off between informativeness and identifiability. Our results show that current models often struggle to balance these objectives: messages that are highly useful to collaborators also tend to reveal substantial information more broadly. Under the same evaluation protocol, human participants achieve up to $4\times$ higher scores than some evaluated LLMs, highlighting a substantial remaining gap in selective communication ability. At the same time, the benchmark exhibits significant headroom, suggesting that controlled information sharing remains a challenging and underdeveloped capability for current systems. The dataset and evaluation scripts will be released upon publication.

Our contributions are as follows:

- We introduce **SNEAK**, a benchmark for evaluating selective information sharing in language models, together with a dataset of communication games spanning diverse semantic domains.
- We propose a behavioral evaluation framework that decomposes communication into *utility* and *leakage*, enabling objective, reproducible evaluation.
- We show that modern LLMs achieve high utility but struggle with information leakage, exhibiting a utility–leakage trade-off and falling short of human performance.

2 The SNEAK benchmark

2.1 Task definition

SNEAK is an asymmetric communication task designed to evaluate selective information sharing in LLMs. Inspired by the party game *The Chameleon* (Big Potato Games, 2017), a model must generate a message about a secret word such that collaborators who know the secret can recognize that the model knows it, while adversaries who do not know the secret cannot reliably infer it. To enable controlled evaluation, SNEAK abstracts this interaction into a one-shot setting with a single generated message and a set of decoys, allowing for consistent and reproducible measurement of utility and leakage.

Game setup Each benchmark instance consists of a semantic category c, a set of words drawn from that category

$$W = \{w_1, \ldots, w_{|W|}\},$$

and a secret word $w_s \in W$. A message-generating model f_θ observes (c, W, w_s) and produces a short natural language message

$$m_\theta = f_\theta(c, W, w_s).$$

To evaluate m_θ, we consider two agents with different information states.

The **chameleon** (adversary) does not know the secret. Given (c, W, m_θ), it predicts a distribution over candidate words:

$$P_{\text{cham}}(w \mid c, W, m_\theta), \quad w \in W.$$

This distribution represents the adversary's belief about the secret.

The **ally** (collaborator) knows the secret word w_s and must identify which message corresponds to it. The ally is given (c, W, w_s) and a set of messages

$$M = \{m_\theta\} \cup M_{\text{decoy}},$$

where M_{decoy} is a set of decoy messages that are generated without access to the secret. The ally predicts a distribution over messages:

$$P_{\text{ally}}(m \mid c, W, w_s, M), \quad m \in M.$$

Metrics Selective communication requires balancing two competing objectives: a message should be recognizable to collaborators who know the secret, but difficult for an uninformed observer to decode. For a generated message m_θ, we define

$$p_{\text{ally}} = P_{\text{ally}}(m_\theta \mid c, W, w_s, M), \quad p_{\text{cham}} = P_{\text{cham}}(w_s \mid c, W, m_\theta).$$

To account for varying set sizes, we normalize both quantities relative to a random-guess baseline. Let $|M|$ denote the number of candidate messages and $|W|$ the number of candidate words. We define

$$\tilde{U}(m) = \frac{p_{\text{ally}} - \frac{1}{|M|}}{1 - \frac{1}{|M|}}, \quad \tilde{L}(m) = \frac{p_{\text{cham}} - \frac{1}{|W|}}{1 - \frac{1}{|W|}}.$$

These normalized quantities equal 0 at chance performance and 1 at perfect performance, but can become negative when performance falls below chance. We clip both quantities at zero, treating below-chance performance as failure and ensuring that all metrics remain interpretable in $[0, 1]$:

$$\text{Utility}(m) = \max(0, \tilde{U}(m)), \quad \text{Leakage}(m) = \max(0, \tilde{L}(m)).$$

We then define a graded score capturing the trade-off between informativeness and secrecy:

$$\text{SoftScore}(m) = \text{Utility}(m) \cdot \big(1 - \text{Leakage}(m)\big).$$

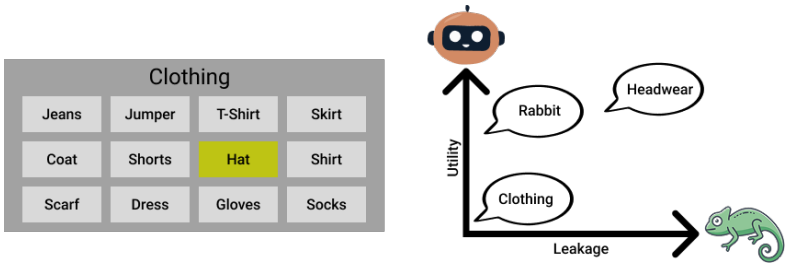

Figure 2: Utility–leakage trade-off for a single instance (secret: hat). The message *rabbit* (as in *a rabbit in a hat*) achieves high utility while maintaining low leakage.

This multiplicative form rewards messages that are both useful and minimally revealing, penalizing failures in either objective. Let $\mathcal{M}_{\leq 5}$ denote the set of all valid natural-language messages of length at most 5 words. For a fixed benchmark instance (c, W, w_s), an optimal message under this objective is

$$m^\star = \arg\max_{m \in \mathcal{M}_{\leq 5}} \text{SoftScore}(m).$$

Figure 2 illustrates this trade-off on a single example, showing how an associative message can remain helpful to an ally without making the secret too easy for the chameleon to identify. All reported values are scaled to percentages by multiplying by 100, so that all metrics lie in $[0, 100]$. In addition, we report a stricter BinaryScore metric. A message is counted as successful if the ally ranks m_θ highest among the candidate messages and the chameleon does not rank the secret word highest among the candidate words:

$$\text{BinaryScore}(m_\theta) = \mathbf{1}\left[m_\theta \in \arg\max_{m \in M} P_{\text{ally}}(m) \ \wedge \ w_s \notin \arg\max_{w \in W} P_{\text{cham}}(w) \right].$$

Benchmark objective Let $\mathcal{D}$ denote the distribution of benchmark instances (c, W, w_s). For a message-generating model f_θ, we report the average utility, leakage, SoftScore, and BinaryScore across instances:

$$\mathbb{E}_{(c,W,w_s)\sim\mathcal{D}}\left[\text{SoftScore}\big(f_\theta(c, W, w_s)\big)\right], \qquad \mathbb{E}_{(c,W,w_s)\sim\mathcal{D}}\left[\text{BinaryScore}\big(f_\theta(c, W, w_s)\big)\right].$$

In practice, these expectations are approximated by averaging over all benchmark instances.

2.2 Dataset

Category sources Categories and word instances are derived from the category production norms of (Banks & Connell, 2023). This dataset contains 117 semantic categories (67 concrete and 50 abstract). In the original study, participants were asked to list as many category members as possible within a fixed time window, producing a large set of human-generated category exemplars. To construct consistent word sets across categories, we select the 12 most frequently produced category members from the category norms dataset. This procedure ensures that word sets contain prototypical and widely recognized category members while avoiding extremely rare or idiosyncratic responses. Treating each candidate word as a possible secret yields 1,394 benchmark instances overall (Table 1).

Decoy message generation To construct the ally evaluation task, we generate 5 decoy messages that are semantically related to the category but are produced without access to the secret word. We query WordNet (Miller, 1995) to extract lexical attributes and related concepts for each word in the category, aggregate these attributes across words, and retain the most frequent ones as decoy messages. Because these messages are derived solely from the category and word set, they do not encode knowledge of the secret and are therefore treated as incorrect messages in the ally task.

Instances	Categories	Words per Category	Decoys
1,394	117	10–12	5

Table 1: Summary of the SNEAK dataset. Each instance corresponds to a (category, word set, secret) tuple with associated decoy messages.

2.3 Evaluation protocol

We estimate utility and leakage using a behavioral evaluation framework based on simulated agents, framing evaluation as a constrained prediction task rather than subjective LLM-as-judge ratings (Li et al., 2025). This enables objective and reproducible scoring by reducing evaluation to discrete identification decisions. To estimate leakage, a chameleon agent observes (c, W, m) and predicts a distribution over candidate words, $P_{\text{cham}}(w \mid c, W, m)$; the probability assigned to the secret w_s defines leakage. To estimate utility, an ally agent observes (c, W, w_s, M) and predicts a distribution over messages, $P_{\text{ally}}(m \mid c, W, w_s, M)$; the probability assigned to m_θ defines utility. Both distributions are obtained via constrained decoding over answer labels (e.g., A, B, C), where probabilities are computed by normalizing first-token logits over the full set of candidate labels. To mitigate label-token biases, we randomize the order of answer choices. Full prompts are provided in Appendix A.

3 Experiments

We evaluate instruction-tuned language models as message generators on SNEAK. For each benchmark instance, a model receives the category, candidate set, and secret word, and produces a single natural-language message, which is then scored using the utility and leakage metrics from Section 2.3.

3.1 Reference baselines

We compare model-generated messages to three simple baselines representing different points on the utility–leakage spectrum: *Random word*, which samples an unrelated WordNet word and is typically uninformative to both agents; *Category-level message*, which uses a synonym or closely related lexicalization of the category name and provides broad semantic information with limited evidence about the secret; and *Secret-level message*, which uses a synonym of the secret word and is highly informative but also highly revealing.

3.2 Inference-time scaling methods

To study the effect of additional test-time computation, we apply two standard inference-time scaling methods to each base model: *Chain-of-Thought (CoT)*, which elicits a brief reasoning trace before producing a message (Wei et al., 2022), and *Self-Enhanced Test-Time Scaling (SETS)*, which generates multiple candidate messages and applies self-verification and self-correction to select a final output (Chen et al., 2025a). Additionally, we introduce a simple two-step transformation method, *Recursive Message Refinement (RMR)*, which first generates a message from the secret and then produces a second message conditioned on the initial message. Intuitively, this procedure encourages more indirect or abstract descriptions, which may help reduce information leakage while preserving signals that allow the ally to recover the intended meaning.

3.3 Models

We evaluate a diverse set of instruction-tuned large language models spanning both open- and closed-source families: GPT, Claude, Gemini, DeepSeek, Llama, Gemma, Mixtral, and Qwen (Singh et al., 2025; Anthropic, 2026; Team et al., 2024; Liu et al., 2024; Grattafiori et al., 2024; Team et al., 2025; Jiang et al., 2024; Yang et al., 2025). All models are used as message generators under the same prompting setup. To evaluate generated messages, we use three

Method	Util.	Leak.	S.S.	B.S.
Reference Baselines				
Random word	7.85	7.89	6.71	7.34
Category-Synonym	31.49	7.56	28.43	30.08
Secret-Synonym	75.73	71.02	12.31	12.15
Human Participants	**91.26**	**33.81**	**59.22**	**54.70**
Language Models				
GPT-5.4	63.49	49.88	21.00	20.88
Claude Opus 4.6	88.41	69.20	22.23	21.43
Gemini 3.1 Pro	15.32	11.85	9.54	10.04
DeepSeek-V3.2	98.52	75.42	23.81	23.17
Llama-4 Scout	95.18	67.11	29.48	28.79
Qwen3-235B	90.79	67.78	25.59	24.75
Gemma-3-27B	92.23	59.23	34.87	33.76
Mixtral-8x22B	96.53	71.00	26.63	26.02
GPT-OSS-120B	71.70	56.71	18.30	17.77

Table 2: Main SNEAK benchmark results. We report Utility, Leakage, SoftScore (S.S.), and BinaryScore (B.S.) for reference baselines and language models. Scores are averaged across Mistral, Llama, and Qwen evaluators. Higher utility and lower leakage are desirable; SoftScore captures the trade-off between the two.

Method	Util.	Leak.	S.S.	B.S.
GPT-5.4				
Zero-shot	63.49	49.88	21.00	20.88
CoT	84.13	67.81	21.84	21.23
SETS	87.02	68.80	23.22	22.69
RMR	84.62	68.21	22.58	22.02
Qwen3-235B				
Zero-shot	90.79	67.78	25.59	24.75
CoT	92.00	67.24	26.86	25.99
SETS	91.89	69.82	24.50	23.72
RMR	77.06	47.04	33.55	32.83
Gemma-3-27B				
Zero-shot	92.23	59.23	34.87	33.76
CoT	83.43	47.94	39.21	38.38
SETS	92.03	59.88	34.32	33.48
RMR	63.28	26.42	40.80	40.91

Table 3: Effect of inference-time scaling methods on SNEAK performance. We compare zero-shot prompting with Chain-of-Thought (CoT), Self-Enhanced Test-Time Scaling (SETS), and Recursive Message Refinement (RMR). Metrics (Utility, Leakage, SoftScore, BinaryScore) are averaged across Mistral, Llama, and Qwen evaluators.

independent evaluator models (Mixtral-8x7B, Llama-3.3-70B-Instruct, and Qwen-2.5-72B-Instruct), each of which simulates the ally and chameleon agents. Final utility, leakage, and score values are obtained by averaging across evaluators, reducing dependence on any single model and improving robustness. Per-evaluator results are provided in Appendix E.

3.4 Main results

Tables 2 and 3 reports average utility, leakage, and composite scores across models and baselines. Overall, modern language models achieve high utility but also incur substantial leakage, indicating a persistent inability to control what information is revealed. Several models (e.g., DeepSeek-V3.2, Mixtral-8x22B, and Llama-4 Scout) approach near-ceiling utility, but do so at the cost of high leakage, placing them in the upper-right region of the utility–leakage plane. In contrast, humans achieve both high utility and substantially lower leakage, resulting in significantly higher SoftScore and BinaryScore. This gap highlights that current models fail to match human-like selective communication, even when they are capable of producing highly informative messages.

Utility–leakage trade-off To better understand model behavior, Figure 3 visualizes the relationship between utility and leakage. Each point corresponds to a baseline, positioned according to its average utility and leakage scores. Across models, we observe a clear trade-off between the two objectives: messages that strongly identify the secret for the ally also tend to increase the probability that the chameleon can infer the secret. Intentionally revealing messages, such as secret-level lexicalizations, achieve high utility but also high leakage, whereas generic category-level messages reduce leakage at the cost of utility. Modern language models typically occupy intermediate points along this spectrum. The dashed curve shows the empirical Pareto frontier, representing the best observed trade-offs between utility and leakage. Moving this frontier outward—toward the upper-left region—corresponds to achieving higher utility at lower levels of leakage. While some models approach this frontier, none consistently attain strong utility with low leakage, indicating that strategic communication under asymmetric information remains challenging for current systems.

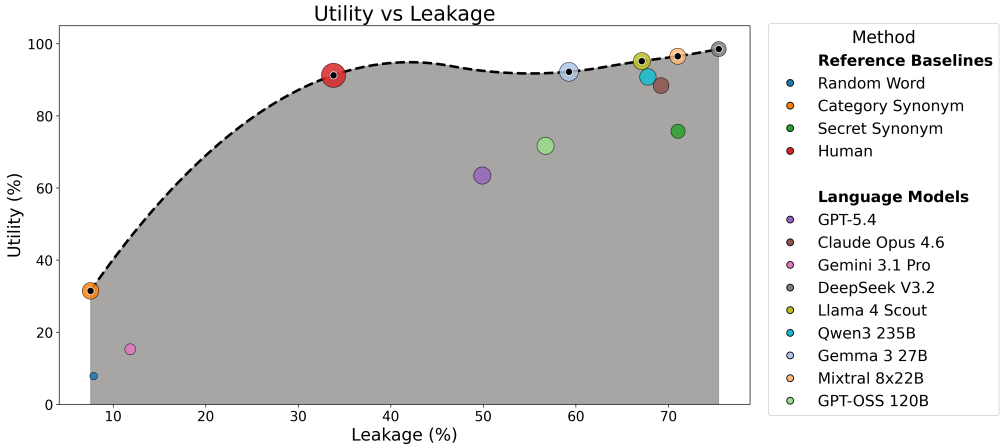

Figure 3: Utility–leakage trade-off across message-generating models. Each point corresponds to a baseline, with marker size proportional to SoftScore. The dashed curve shows the empirical Pareto frontier. The upper-left region corresponds to desirable behavior: high utility for the ally and low leakage to the chameleon.

4 Analysis

4.1 Evaluator robustness

Different evaluator models To ensure that benchmark results are not dependent on a single evaluator, we compute utility and leakage using three independent evaluation models: Mixtral-8x7B, Llama-3.3-70B-Instruct, and Qwen-2.5-72B-Instruct. Each evaluator performs the ally and chameleon inference tasks described in Section 2.3, and final benchmark scores are obtained by averaging across evaluators. Across all experiments, score variance is modest: the average standard deviation across evaluators is 5.42 and 4.76 percentage points for utility and leakage respectively, indicating stable absolute scores. Rankings are also highly consistent across evaluators, with Spearman correlations ranging from 0.9042 to 0.9680 for utility, 0.9547 to 0.9929 for leakage (lower is better), and 0.8995 to 0.9547 for the combined SoftScore, indicating strong agreement in model ordering. No single evaluator consistently assigns higher or lower scores across models. These results indicate that the SNEAK evaluation protocol is robust to the choice of evaluator model. We further test for same-model evaluator bias by evaluating outputs from each model using all three evaluators; results remain consistent, with full analysis provided in Appendix E.2.

Human alignment Because the SNEAK evaluation relies on simulated agents implemented using language models, we conduct a human study to validate both the task formulation and the alignment of model-based evaluators with human judgments. We sample 117 benchmark instances (one per category) and collect five independent annotations per item for both the ally and chameleon tasks. Humans perform well above chance on both tasks: ally accuracy is 77.3% at the worker level (chance: 16.7%) and 88.9% under plurality voting, while chameleon accuracy is 44.6% (chance: 8.3%) and 68.4% under plurality voting. Inter-annotator agreement is substantial for the ally task (Fleiss' $\kappa = 0.73$) and moderate for the chameleon task ($\kappa = 0.27$), reflecting greater ambiguity in adversarial inference. We next compare human judgments to LLM-based evaluators. Averaged across three evaluators, model predictions agree with human plurality judgments on 88.9% of ally items and 66.7% of chameleon items, increasing to 91.7% and 74.2% respectively when restricting to items with a strict human majority. Evaluator probabilities also correlate with human vote shares (ally utility: $r = 0.61$; chameleon leakage: $r = 0.39$), indicating alignment at the distribution level. These results suggest that the model-based evaluation framework provides a reasonable proxy for human judgments while enabling scalable measurement. See Appendix B for more details.

4.2 Semantic structure of category difficulty

To better understand category-level variation, we analyze difficulty as a function of semantic domain and candidate-space structure. We first split categories into abstract and concrete domains following the source norms. Abstract categories are easier overall, with higher utility (61.8 vs. 58.8), lower leakage (47.6 vs. 50.0), and a higher mean SoftScore (28.4 vs. 25.2). This suggests that abstract domains often permit more indirect associative messages, allowing models to signal knowledge of the secret without making it uniquely identifiable. We also quantify semantic clustering within each category using the average pairwise WordNet similarity among candidate words. Across all categories, higher semantic similarity is associated with lower utility ($r = -0.43$) and lower mean score ($r = -0.28$), indicating that densely clustered candidate sets are harder for selective communication. Intuitively, when candidates are too semantically similar, messages often fail to distinguish the intended secret even for an informed ally. Interestingly, this effect differs by domain: in abstract categories, similarity is more strongly associated with reduced utility and score, whereas in concrete categories it is associated with lower leakage. This suggests that semantic clustering affects strategic communication through different mechanisms in different domains.

4.3 Sensitivity to candidate and decoy set sizes

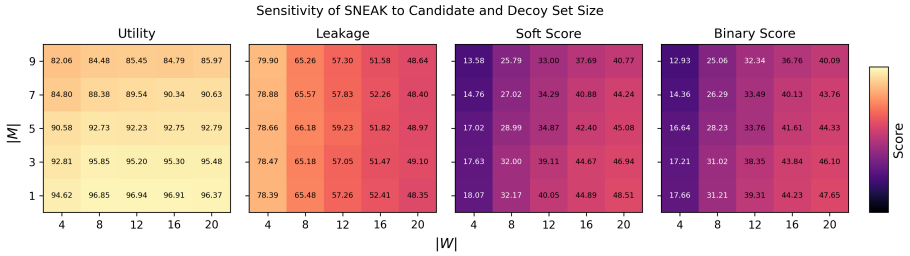

Figure 4: Sensitivity of SNEAK performance for Gemma-3-27B to candidate set size ($|W|$) and number of decoy messages ($|M|$)

The difficulty of the SNEAK task is governed by two interpretable parameters: the candidate set size $|W|$ and the number of decoy messages $|M|$. Increasing $|W|$ expands the adversary's search space and generally reduces leakage, while increasing $|M|$ makes the ally task harder by requiring identification of the correct message among more alternatives, thereby reducing utility. To assess the sensitivity of the benchmark to these design choices, we sweep over both parameters and recompute utility, leakage, SoftScore, and BinaryScore using the protocol described in Section 2.3. Figure 4 shows the resulting trends for Gemma-3-27B, averaged across evaluator models; results for additional generators are reported in Appendix E.4. The two parameters exert opposing effects on the benchmark: larger candidate sets primarily decrease leakage, whereas more decoy messages primarily decrease utility. As a result, SoftScore and BinaryScore vary in predictable ways across configurations. These results show that SNEAK provides a stable and controllable evaluation setting for selective communication. We further evaluate robustness to decoy construction in Appendix C, finding that results are consistent across alternative decoy generation strategies.

5 Related work

A large body of work evaluates language models on factual knowledge, reasoning, and dialogue capabilities. Benchmarks such as MMLU (Hendrycks et al., 2020) and GSM8K (Cobbe et al., 2021) measure question answering and mathematical reasoning, while MT-Bench (Zheng et al., 2023) evaluates conversational helpfulness using model-based judges. More broadly, recent benchmark suites assess capabilities such as coding, planning, and instruction following (Mohammadi et al., 2025). These benchmarks provide *utterance-level evaluation* of model outputs, but they generally assume a single agent with full access to

Benchmark	Multi-Agent?	Asymmetric Info?	Strategic Signaling?	Utterance Eval?	Leakage Measured?
MMLU-Pro (Wang et al., 2024b)	✗	✗	✗	✓	✗
MT-Bench (Zheng et al., 2023)	✗	✗	✗	✓	✗
SocialMaze (Xu et al., 2025)	✓	✓	✗	✗	✗
Mini-Mafia (Costa & Vicente, 2025)	✓	✓	✓	✗	✗
WhoIsSpy (Chen et al., 2025b)	✓	✓	✓	✗	✗
SNEAK (ours)	✓	✓	✓	✓	✓

Table 4: Comparison of SNEAK with representative LLM benchmarks across key properties.

the task information. In contrast, recent multi-agent and social-deduction benchmarks (e.g., SocialMaze, Mini-Mafia, WhoIsSpy) introduce asymmetric information and strategic interaction, but typically evaluate behavior through multi-turn game outcomes rather than directly scoring individual utterances. As summarized in Table 4, SNEAK uniquely combines asymmetric information, strategic signaling, and multi-agent interaction with *utterance-level evaluation* in a single-turn setting, and explicitly measures information leakage.

Our benchmark is also closely related to research on pragmatics and reasoning about the beliefs of other agents. Classical accounts of communication emphasize that speakers choose utterances relative to shared context and listener beliefs, often balancing informativeness with ambiguity (Grice, 1975). Computational frameworks such as Rational Speech Acts formalize this process as recursive reasoning between speakers and listeners (Frank & Goodman, 2012; Goodman & Frank, 2016). Related ideas appear in signaling games, emergent communication, and theory-of-mind reasoning in language models (Lazaridou et al., 2017; Havrylov & Titov, 2017; Kosinski, 2023; Yao et al., 2023; Sap et al., 2019). While these lines of work study belief inference, pragmatic reasoning, and strategic interaction, they do not directly evaluate whether models can produce utterances that selectively reveal information to different listeners. SNEAK operationalizes this as a controlled one-shot setting where models inform collaborators while limiting inference by uninformed observers.

6 Limitations

SNEAK evaluates selective information sharing in a simplified communication setting. The benchmark focuses on single-turn message generation and does not capture the richer dynamics of multi-turn interaction, discussion, or deception present in real communication. Evaluation relies on simulated ally and adversary agents implemented using LLMs. While this enables scalable evaluation, model-based judges may not perfectly reflect how humans interpret messages. Finally, the dataset is derived from semantic category norms and therefore emphasizes structured conceptual domains. Real-world communication often involves richer context and background knowledge that are not represented in this setting. We view SNEAK as an initial step toward evaluating selective communication in LLMs and hope future work will extend this framework to more realistic multi-agent settings.

7 Conclusion

We introduced **SNEAK**, a benchmark for evaluating selective information sharing in LLMs. The benchmark measures whether models can generate messages that are informative to collaborators who share context while remaining ambiguous to observers who do not. We propose a behavioral framework that quantifies both the utility of a message for allies and the information leakage to adversaries. Our experiments show that modern LLMs often struggle to balance these competing objectives: messages that are highly informative for collaborators frequently reveal substantial information to adversaries. This suggests that strategic communication under asymmetric information remains a challenging capability for current systems. By providing a controlled setting for measuring how language models manage information flow between agents, SNEAK highlights selective communication as an important and underexplored dimension of language model behavior.

Ethics Statement

This work studies selective information sharing in language models, focusing on how agents can communicate useful information to intended recipients while limiting unintended inference. While this capability has beneficial applications (e.g., tutoring, privacy-preserving communication, and controlled information disclosure), it may also raise concerns related to strategic obfuscation or misuse in deceptive settings. Our goal is to provide a controlled evaluation framework for measuring these behaviors, rather than to promote or optimize for deceptive communication. The SNEAK benchmark isolates a simplified, single-turn setting and does not model real-world deployment contexts or incentives. We emphasize that the ability to control information leakage should be studied alongside considerations of transparency, safety, and appropriate use. We release our dataset and evaluation framework to support further research on responsible communication in AI systems, including improving models' ability to balance informativeness with privacy and safety considerations.

Acknowledgments

We thank Amir Avsian for introducing us to the game *The Chameleon*, which inspired the formulation of this task.

References

Jacob Andreas. Language models as agent models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 5769–5779, 2022.

Anthropic. Claude opus 4.6 system card. https://www-cdn.anthropic.com/14e4fb01875d2a69f646fa5e574dea2b1c0ff7b5.pdf, 2026. Anthropic technical report.

Briony Banks and Louise Connell. Category production norms for 117 concrete and abstract categories. *Behavior Research Methods*, 55(3):1292–1313, 2023.

Brian R Belland, ChanMin Kim, and Michael J Hannafin. A framework for designing scaffolds that improve motivation and cognition. *Educational psychologist*, 48(4):243–270, 2013.

Big Potato Games. The chameleon. https://bigpotato.com/products/the-chameleon, 2017. Board game.

Jiefeng Chen, Jie Ren, Xinyun Chen, Chengrun Yang, Ruoxi Sun, Jinsung Yoon, and Sercan Ö Arık. Sets: Leveraging self-verification and self-correction for improved test-time scaling. *arXiv preprint arXiv:2501.19306*, 2025a.

Junhao Chen, Jingbo Sun, Xiang Li, Haidong Xin, Yuhao Xue, Yibin Xu, and Hao Zhao. Llmspark: A benchmark for evaluating large language models in strategic gaming contexts. *arXiv preprint arXiv:2509.16610*, 2025b.

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

Davi Bastos Costa and Renato Vicente. Deceive, detect, and disclose: Large language models play mini-mafia. *arXiv preprint arXiv:2509.23023*, 2025.

Vincent P Crawford and Joel Sobel. Strategic information transmission. *Econometrica: Journal of the Econometric Society*, pp. 1431–1451, 1982.

Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, King Zhu, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, et al. Supergpqa: Scaling llm evaluation across 285 graduate disciplines. *arXiv preprint arXiv:2502.14739*, 2025.

FAIR, Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, et al. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624): 1067–1074, 2022.

Michael C. Frank and Noah D. Goodman. Predicting pragmatic reasoning in language games. *Science*, 336(6084):998, 2012.

Noah D. Goodman and Michael C. Frank. Pragmatic language interpretation as probabilistic inference. *Trends in Cognitive Sciences*, 20(11):818–829, 2016.

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

Herbert P Grice. Logic and conversation. In *Speech acts*, pp. 41–58. Brill, 1975.

Serhii Havrylov and Ivan Titov. Emergence of language with multi-agent games. In *NeurIPS*, 2017.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Mehran Kazemi, Bahare Fatemi, Hritik Bansal, John Palowitch, Chrysovalantis Anastasiou, Sanket Vaibhav Mehta, Lalit K Jain, Virginia Aglietti, Disha Jindal, Yuanzhu Peter Chen, et al. Big-bench extra hard. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 26473–26501, 2025.

Michal Kosinski. Theory of mind may have spontaneously emerged in large language models. *arXiv preprint arXiv:2302.02083*, 2023.

Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. Multi-agent cooperation and the emergence of (natural) language. In *ICLR*, 2017.

Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, et al. From generation to judgment: Opportunities and challenges of llm-as-a-judge. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 2757–2791, 2025.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Wenting Ma, Olusola O Adesope, John C Nesbit, and Qing Liu. Intelligent tutoring systems and learning outcomes: A meta-analysis. *Journal of educational psychology*, 106(4):901, 2014.

Ken McRae, George S Cree, Mark S Seidenberg, and Chris McNorgan. Semantic feature production norms for a large set of living and nonliving things. *Behavior research methods*, 37(4):547–559, 2005.

George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11): 39–41, 1995.

Mahmoud Mohammadi, Yipeng Li, Jane Lo, and Wendy Yip. Evaluation and benchmarking of llm agents: A survey. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 6129–6139, 2025.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pp. 1–22, 2023.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social iqa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 4463–4473, 2019.

Patrick Shafto, Noah D Goodman, and Thomas L Griffiths. A rational account of pedagogical reasoning: Teaching by, and learning from, examples. *Cognitive psychology*, 71:55–89, 2014.

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.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Kimi Team, Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, Bowen Wang, Cheng Chen, Chenlin Zhang, Chenzhuang Du, Chu Wei, et al. Kimi-vl technical report. *arXiv preprint arXiv:2504.07491*, 2025.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024a.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024b.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Ling-ling Wu and Lawrence W Barsalou. Perceptual simulation in conceptual combination: Evidence from property generation. *Acta psychologica*, 132(2):173–189, 2009.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. Autogen: Enabling next-gen llm applications via multi-agent conversations. In *First conference on language modeling*, 2024.

Zixiang Xu, Yanbo Wang, Yue Huang, Jiayi Ye, Haomin Zhuang, Zirui Song, Lang Gao, Chenxi Wang, Zhaorun Chen, Yujun Zhou, et al. Socialmaze: A benchmark for evaluating social reasoning in large language models. *arXiv preprint arXiv:2505.23713*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Shunyu Yao et al. React: Synergizing reasoning and acting in language models. In *ICLR*, 2023.

Haolan Zhan, Yufei Wang, Zhuang Li, Tao Feng, Yuncheng Hua, Suraj Sharma, Lizhen Qu, Zhaleh Semnani Azad, Ingrid Zukerman, and Reza Haf. Let's negotiate! a survey of negotiation dialogue systems. In *Findings of the Association for Computational Linguistics: EACL 2024*, pp. 2019–2031, 2024.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

A Prompts

This section presents the prompt templates used in the SNEAK benchmark for message generation and evaluation. We include the prompt used by the message-generating model, as well as the prompts used by the evaluator models that simulate the ally and chameleon agents. Variables such as the category, candidate set, secret word, and message set are inserted programmatically at runtime.

Message Generation

System: You are an expert player of the party game "Chameleon".
You will be given:

- A category
- A list of candidate words
- The secret word (one of the candidates)

Your task is to produce a single message that signals to other honest players that you know the secret word, while not revealing the word directly.
Guidelines:

- The message should strongly hint at the secret word.
- The message should help distinguish the secret from the other candidates.
- The message must not contain the secret word or obvious variations of it.
- Do not spell the word, give its initials, or describe its letters.
- Do not translate the word or use a direct synonym.
- Avoid messages that apply equally well to many of the candidates.
- Keep the message short and natural (1-5 words).

Think carefully about what property uniquely characterizes the secret compared to the other candidates.
Output format:
`<message>YOUR message HERE</message>`

User:
```
Category: {CATEGORY}
Candidate words: {CANDIDATES}
Secret word: {SECRET}
Write one message that hints at the secret word without revealing it.
Return the result in the format:
<message>...</message>
```

Chameleon Evaluation

System: You are the judge for the game "Chameleon".
You will be given:

- A category
- A list of candidate words, each tagged with a letter label
- A single message from a player

Task:
Infer which labeled candidate is the secret word the message refers to.
Rules:

- Output exactly one uppercase letter label.
- Do not output any explanation or extra text.

User:
Category: {CATEGORY}
Candidate words: A) {WORD_1} B) {WORD_2} C) {WORD_3} ...
Message: {MESSAGE}
Choose exactly one label from: A, B, C, ...
Answer with only that single letter.

Ally Evaluation

System: You are the Ally judge in the game "Chameleon".
You will be given:

- A category
- A list of candidate words
- The true secret word
- A set of messages, each tagged with a letter label

Task:
Decide which labeled message most likely came from a player who knew the secret word.
Rules:

- Output exactly one uppercase letter label.
- Do not output any explanation or extra text.

User:
Category: {CATEGORY}
Candidate words: {CANDIDATES}
Secret word: {SECRET}
Messages: A) message_0: {message_0} B) message_1: {message_1} C) message_2: {message_2} ...
Choose exactly one label from: A, B, C, ...
Answer with only that single letter.

For message generation, we extract the content of the final <message>...</message> span from the model output. If no such span is present, the generation is marked as a parsing failure. For ally and chameleon evaluation, models are prompted to produce a single letter corresponding to one of the presented options. At evaluation time, we use constrained label scoring to obtain a probability distribution over the available letter labels, which is then mapped back to the corresponding candidate words or message identifiers. These probabilities are used to compute utility and leakage. Outputs that cannot be parsed or do not assign valid probabilities to the full label set are marked as evaluation parsing failures.

B Human Study

We conduct human studies to evaluate both (i) human performance in message generation and (ii) human judgments in message evaluation, as well as agreement between human annotators and LLM-based evaluators. To ensure consistent coverage across semantic domains, both studies use the same set of 117 benchmark instances (one per category), but employ separate annotation setups.

Annotators. All annotations were collected on Amazon Mechanical Turk (AMT) using workers with a minimum approval rate of 95%. Annotators were restricted to English-speaking countries to ensure reliable comprehension. Compensation was set to meet or exceed the U.S. federal minimum wage based on conservative task-time estimates. No personally identifiable information was collected.

B.1 Human Message Generation

We evaluate human performance on the message generation task using the shared set of 117 benchmark instances. For each instance, annotators are given the category, candidate word set, and secret word, and are asked to produce a single-word message that indicates knowledge of the secret without revealing it too directly. Each instance is annotated by three independent annotators.

To assess performance, we evaluate human-generated messages using the automated evaluation protocol described in Section 2.3, measuring utility, leakage, SoftScore, and BinaryScore. For each instance, we collect three independent responses and report the highest-scoring response, approximating an upper bound on human performance under annotator variability. Results are then averaged across instances.

Source	Utility	Leakage	SoftScore	BinaryScore
Human Participants	91.26	33.81	59.22	54.70

Table 5: Performance of human-generated messages under the SNEAK evaluation protocol. Each instance includes three human responses; we report the best-performing response per instance.

B.2 Human Message Evaluation

We evaluate human judgments on the message evaluation tasks using the same set of 117 benchmark instances, but with a separate annotation setup. Each item is annotated independently by five annotators.

Ally evaluation: given the category, candidate set, secret word, and a set of messages (one gold and several decoys), select the message corresponding to the secret.

Chameleon evaluation: given the category, candidate set, and a message, select the most likely secret word.

We report both worker-level accuracy and item-level accuracy obtained via plurality voting across annotators.

Task	Worker Acc.	Plurality Acc.
Ally	77.3%	88.9%
Chameleon	44.6%	68.4%

Table 6: Human performance on SNEAK message evaluation tasks. Chance levels are 16.7% for ally and 8.3% for chameleon.

Inter-annotator agreement is substantial for the ally task (Fleiss' $\kappa = 0.73$) and moderate for the chameleon task ($\kappa = 0.27$), reflecting greater ambiguity in adversarial inference.

B.3 Human–LLM Agreement

We compare human judgments in message evaluation to LLM-based evaluators at the item level. Human decisions are aggregated using plurality voting across five annotators, while LLM predictions are obtained by averaging across three evaluator models (Mixtral-8x7B-Instruct-v0.1, Llama-3.3-70B-Instruct, and Qwen-2.5-72B-Instruct).

To assess alignment at the distribution level, we compare human vote shares with evaluator probabilities. The correlation between human and model estimates is $r = 0.61$ for ally utility and $r = 0.39$ for chameleon leakage, indicating moderate alignment between human judgments and model-based evaluation.

Task	Top-1 Agreement	Strict-Majority Agreement
Ally	88.9%	91.7%
Chameleon	66.7%	74.2%

Table 7: Agreement between LLM evaluators and human plurality judgments. Strict-majority considers only items with a clear human majority.

These results support the use of LLM-based evaluators as scalable proxies for human judgments in the SNEAK benchmark.

B.4 Mturk Interfaces

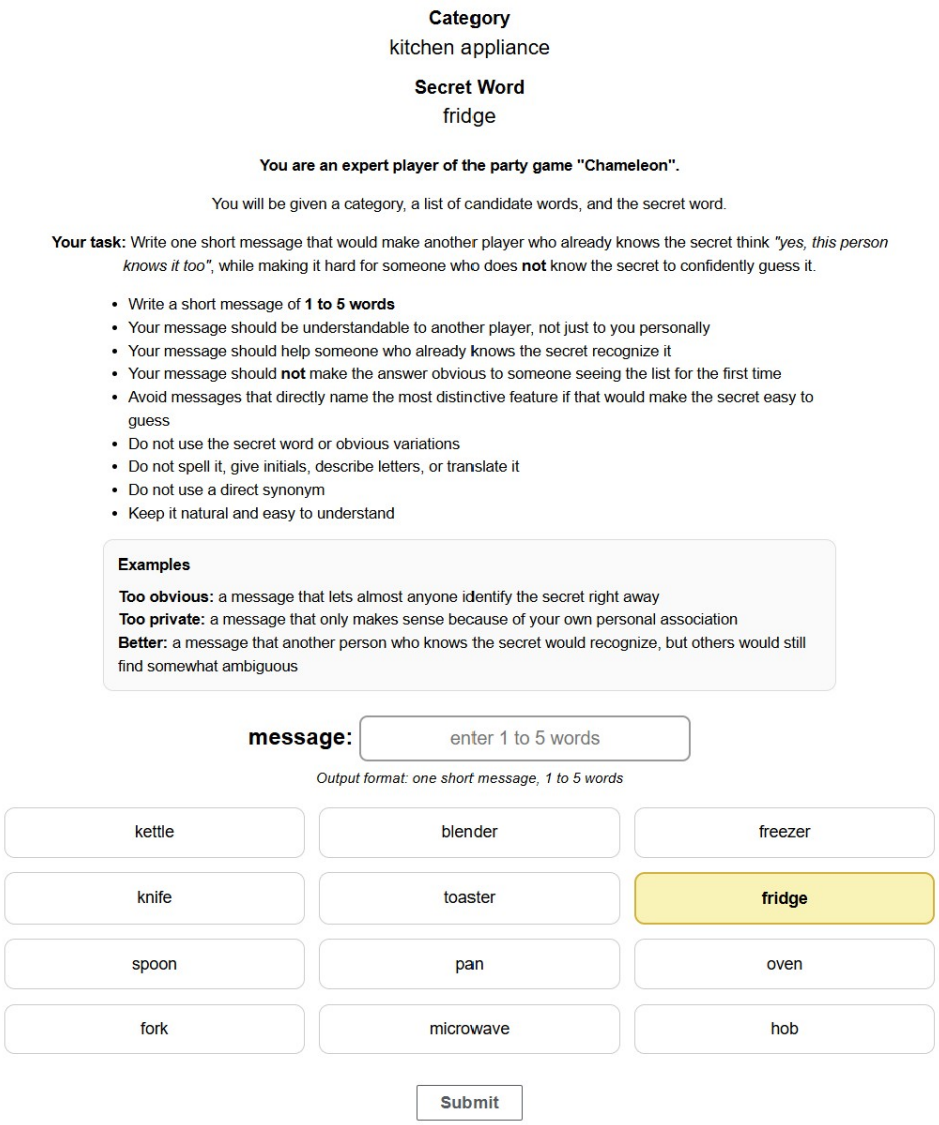

Figure 5: Human annotation interface for message generation. Annotators are given the category, candidate set, and secret word, and are asked to produce a 1-5 word message that signals knowledge of the secret without revealing it directly.

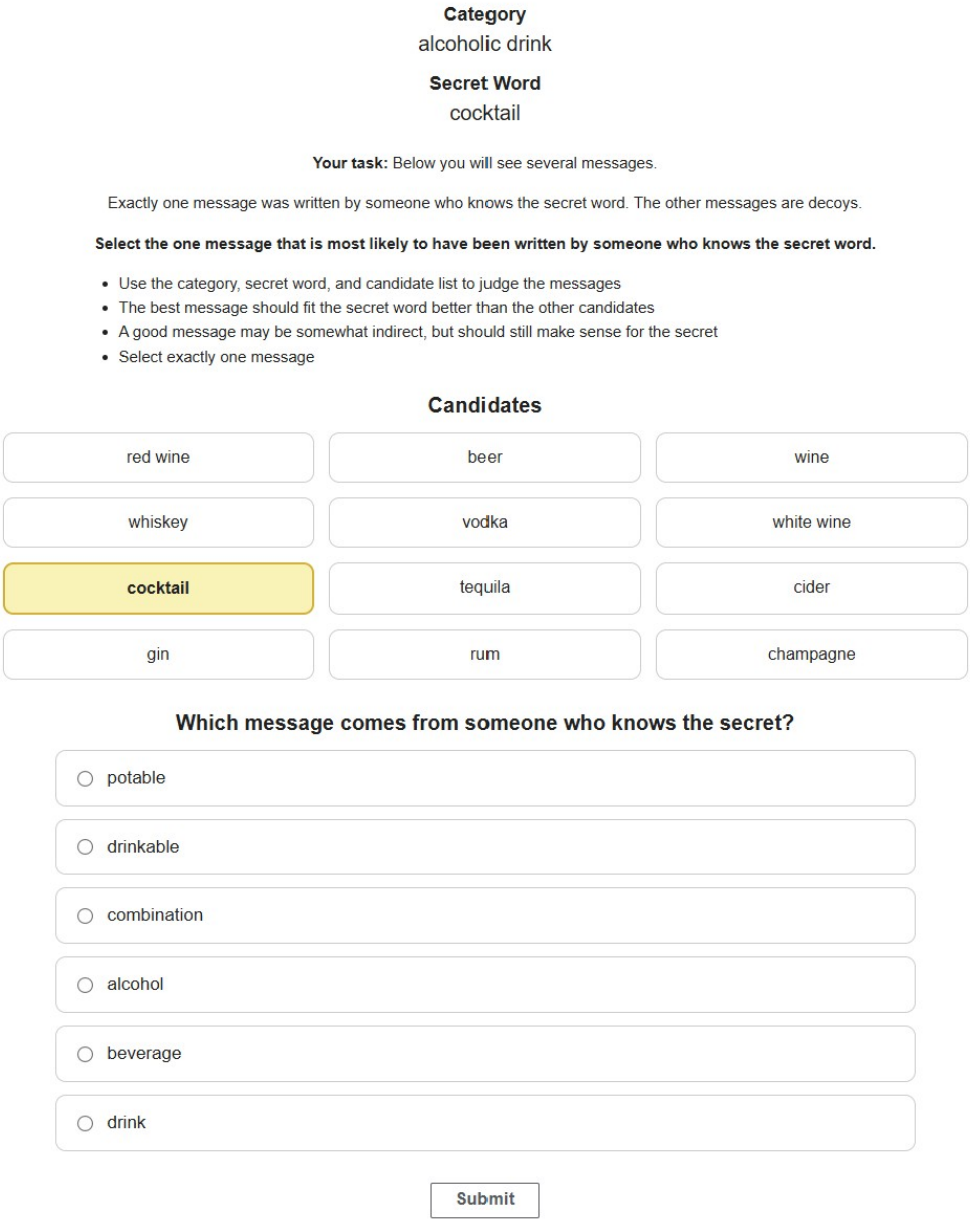

Figure 6: Human annotation interface for the *ally evaluation* task. Annotators are given the category, candidate word set, secret word, and a set of messages (one gold and several decoys), and are asked to select the message corresponding to the secret.

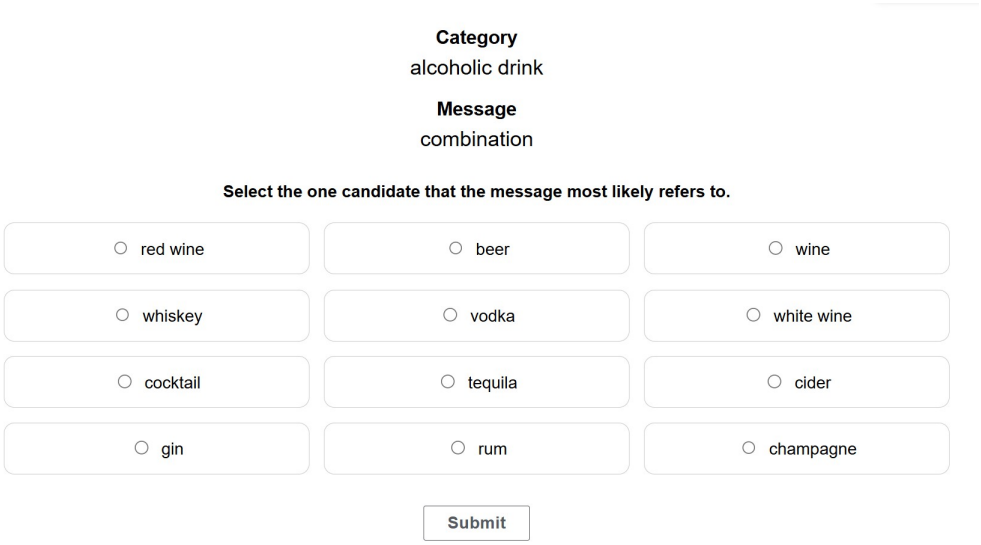

Figure 7: Human annotation interface for the *chameleon evaluation* task. Annotators are given the category, candidate word set, and a message, and are asked to select the most likely secret word inferred from the message.

C Decoy Ablation

We analyze the sensitivity of SNEAK to decoy construction by replacing the default decoy set with LLM-generated alternatives. For each instance, we generate decoy messages using Qwen2.5-72B-Instruct, prompting the model to produce short, broad terms that apply to many items in the category while excluding the category name and candidate words. We retain the top five valid decoys after filtering.

Re-evaluating all model outputs under this modified setting reveals that results are highly stable: across LLM systems, average scores change by less than 6 in utility, less than 1 in leakage, and less than 3 in both SoftScore and BinaryScore (absolute differences). These findings indicate that the benchmark is robust to the choice of decoy construction, and that performance is not driven by artifacts of the default decoy generation process.

Decoy Generation

System: You create decoy terms for a Chameleon-style word game.
Your goal is to produce broad terms that can plausibly apply to many candidate words.
Follow the output format exactly.

User:

```
Generate decoy terms for this category.

Category: {CATEGORY}

Candidate words:
- {WORD_1}
- {WORD_2}
- {WORD_3}
...

Already accepted decoys (do not repeat):
- {DECOY_1}
- {DECOY_2}
...
- (none)

Requirements:
- Return exactly {SUGGEST_K} decoy terms, ranked by how many candidate words
 they can cover.
- Decoys should be general descriptors, contexts, or associations.
- Each decoy should be short (1 to 5 words).
- A decoy must NOT exactly match the category text.
- A decoy must NOT exactly match any candidate word.
- Use words that are different from the category/candidate words.
- Do not output explanations.

Output exactly {SUGGEST_K} lines in this format:
<decoy>term</decoy>
```

	Utility				Leakage				SoftScore				BinaryScore			
Baseline	Mistral	Llama	Qwen	Avg	Mistral	Llama	Qwen	Avg	Mistral	Llama	Qwen	Avg	Mistral	Llama	Qwen	Avg
Random	2.28	6.11	11.61	6.67	8.67	7.04	7.94	7.88	1.86	4.77	9.94	5.52	1.94	4.81	11.98	6.24
Category Synonym	21.61	11.48	15.97	16.35	7.17	8.01	7.51	7.56	19.47	9.85	14.15	14.49	19.80	10.19	16.43	15.47
Secret Synonym	62.22	69.67	75.32	69.07	66.02	74.02	72.88	70.97	11.21	8.60	11.52	10.44	11.12	8.32	11.84	10.43
GPT-5.4	67.87	85.43	88.15	80.48	57.79	76.54	72.43	68.92	23.61	16.44	21.48	20.51	23.74	15.85	20.95	20.18
Llama-4 Scout	85.24	94.44	95.13	91.60	58.81	72.58	69.56	66.98	32.18	23.45	26.77	27.47	31.99	22.53	25.90	26.81
Qwen3	77.50	88.55	91.81	85.95	55.13	75.59	72.27	67.66	30.07	17.22	22.03	23.10	29.84	16.64	20.80	22.43
Gemma-3	75.47	90.68	92.61	86.26	48.66	67.24	61.67	59.19	34.60	26.85	33.27	31.57	34.36	25.82	32.35	30.85
Mixtral-8x22B-Instruct	87.42	95.59	96.51	93.17	62.37	76.19	74.49	71.02	31.00	21.07	23.43	25.17	30.85	20.59	22.17	24.53
GPT-OSS-120B	59.45	68.42	76.93	68.26	48.81	62.74	59.55	57.03	19.64	10.37	20.96	16.99	19.80	10.04	20.59	16.81

Table 8: Decoy ablation results using LLM-generated decoys.

D Qualitative Examples

We show representative examples of generated messages and their utility–leakage trade-offs.

Category	Secret	Candidates	Message	Util.	Leak.	S.S.	B.S.
Academic Subject	Psychology	maths, history, biology, sociology, chemistry, geography, french, physics, english literature, spanish, english, psychology	**Mind**	100	97.5	2.5	0.0
Animal	Dog	cat, giraffe, elephant, lion, horse, rabbit, cow, pig, tiger, sheep, fish, dog	**Fetch**	100	99.9	0.1	0.0
Body of Water	Ocean	sea, lake, river, pond, stream, puddle, reservoir, canal, swimming pool, bath, atlantic ocean, ocean	**Salty**	66.4	33.3	33.1	33.3
Colour	Green	blue, yellow, purple, red, orange, black, pink, white, grey, turquoise, violet, green	**Grass**	66.5	100	0.0	0.0
Dairy Product	Ice Cream	milk, cheese, yoghurt, butter, cream, chocolate, eggs, sour cream, margarine, whipped cream, cadbury, ice cream	**Cone**	100	100	0.0	0.0
Emotion	Happy	sad, angry, fear, excited, depressed, anxious, love, frustration, nervous, jealous, upset, happy	**Birthday**	66.7	33.3	33.3	33.3

Table 9: Example messages and their utility–leakage trade-offs. Utility, leakage, SoftScore, and BinaryScore are averaged across evaluator models.

E Additional Analyses

E.1 Per-evaluator results

Table 10 reports SNEAK performance broken down by evaluator model (Mistral, Llama, and Qwen). Results are consistent across evaluators, with similar model rankings and trade-offs between utility and leakage. This supports the robustness of the evaluation framework and indicates that conclusions are not driven by a particular evaluator choice.

	Utility				**Leakage**				**SoftScore**				**BinaryScore**			
Baseline	Mistral	Llama	Qwen	Avg	Mistral	Llama	Qwen	Avg	Mistral	Llama	Qwen	Avg	Mistral	Llama	Qwen	Avg
Random	2.4	10.2	11.0	7.8	8.7	7.0	8.0	7.9	2.0	8.6	9.5	6.7	2.1	9.1	10.8	7.3
Category Synonym	31.4	30.1	33.1	31.5	7.2	8.0	7.5	7.6	28.7	26.5	30.1	28.4	29.3	27.1	33.9	30.1
Secret Synonym	66.6	80.2	80.4	75.7	66.1	74.1	72.9	71.0	12.3	11.6	13.0	12.3	12.1	11.5	12.8	12.1
Human Participants	84.5	97.0	92.2	91.3	28.8	38.5	34.1	33.8	60.0	58.3	59.3	59.2	58.1	51.3	54.7	54.7
Claude Opus 4.6	78.8	94.1	92.4	88.4	57.1	76.6	73.9	69.2	28.0	18.8	19.8	22.2	27.8	17.9	18.5	21.4
Gemini 3.1 Pro	9.1	18.6	18.2	15.3	11.6	12.3	11.7	11.8	5.4	11.8	11.4	9.5	5.5	11.8	12.8	10.0
DeepSeek-V3.2	96.9	99.5	99.2	98.5	66.7	81.0	78.6	75.4	32.0	18.7	20.8	23.8	31.6	18.4	19.6	23.2
Llama-4 Scout	90.5	97.7	97.4	95.2	59.2	72.6	69.6	67.1	34.6	25.5	28.3	29.5	34.4	24.5	27.4	28.8
Mixtral-8x22B-Instruct	93.1	98.3	98.2	96.5	62.3	76.3	74.5	71.0	33.3	22.3	24.3	26.6	33.3	22.0	22.8	26.0
GPT-OSS-120B	63.8	71.9	79.3	71.7	48.9	61.8	59.5	56.7	20.6	12.3	22.0	18.3	20.4	11.9	21.0	17.8
GPT-5.4	48.8	71.8	69.9	63.5	42.7	55.8	51.1	49.9	16.9	22.1	24.0	21.0	16.9	21.7	24.0	20.9
GPT-5.4 + CoT	70.3	92.7	89.4	84.1	55.0	76.6	71.8	67.8	25.6	18.8	21.2	21.8	25.6	18.1	20.0	21.2
GPT-5.4 + SETS	73.2	94.4	93.4	87.0	57.3	76.2	72.8	68.8	26.0	20.5	23.2	23.2	25.8	19.7	22.5	22.7
GPT-5.4 + RMR	70.4	92.2	91.3	84.6	57.4	75.2	72.0	68.2	25.4	20.2	22.1	22.6	25.6	19.6	20.9	22.0
Qwen3	82.2	94.4	95.7	90.8	55.5	75.6	72.3	67.8	32.1	20.2	24.5	25.6	31.9	19.7	22.7	24.7
Qwen3 + CoT	84.6	95.9	95.5	92.0	53.9	75.6	72.3	67.2	35.0	21.1	24.4	26.9	34.7	20.5	22.7	26.0
Qwen3 + SETS	83.8	95.3	96.5	91.9	57.0	78.2	74.3	69.8	31.8	18.3	23.3	24.5	31.7	17.9	21.6	23.7
Qwen3 + RMR	65.6	82.8	82.8	77.1	36.2	54.0	50.9	47.0	35.1	31.1	34.5	33.6	34.8	30.4	33.3	32.8
Gemma-3	83.8	97.1	95.8	92.2	48.7	67.3	61.7	59.2	39.4	30.3	34.8	34.9	39.0	29.2	33.1	33.8
Gemma-3 + CoT	69.0	91.7	89.6	83.4	36.3	56.1	51.4	47.9	39.3	37.5	40.9	39.2	38.9	36.8	39.5	38.4
Gemma-3 + SETS	83.3	96.5	96.3	92.0	49.2	67.1	63.3	59.9	38.7	30.4	33.9	34.3	38.5	30.0	32.0	33.5
Gemma-3 + RMR	48.4	72.0	69.4	63.3	21.0	30.7	27.5	26.4	33.2	44.0	45.2	40.8	33.4	43.6	45.8	40.9

Table 10: Per-evaluator SNEAK results using Mistral, Llama, and Qwen as evaluators. Metrics are averaged across evaluators.

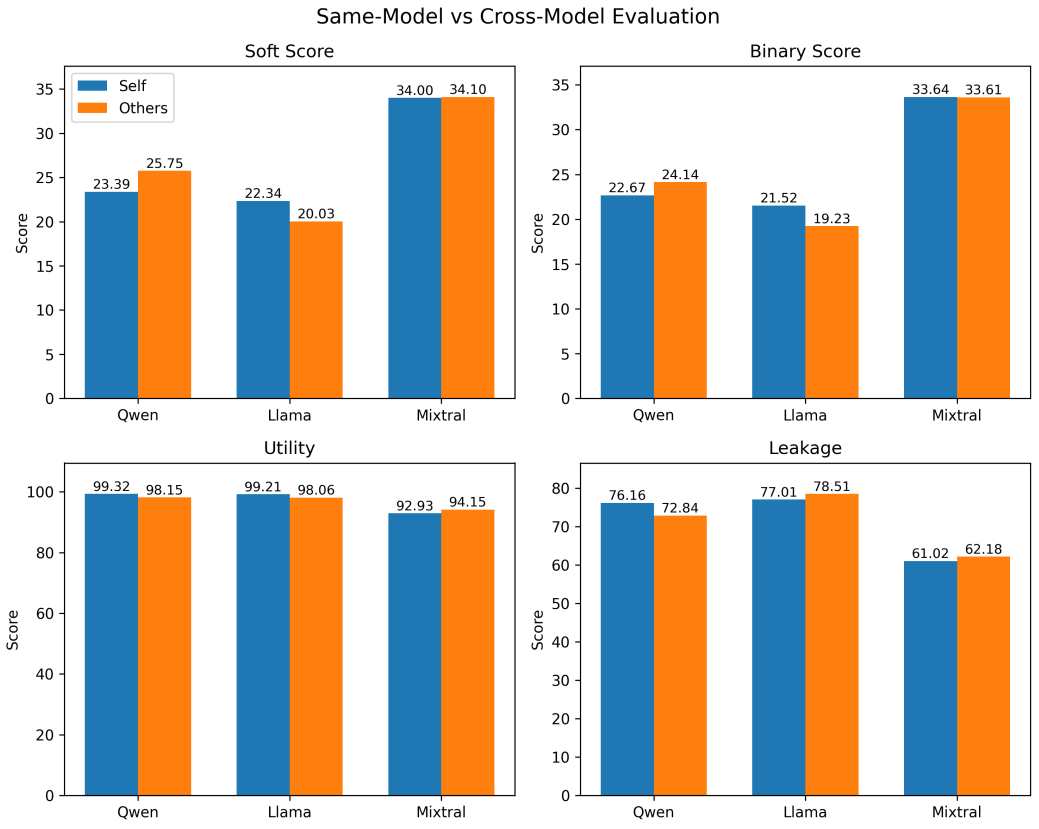

Figure 8: Comparison of self-evaluation and cross-model evaluation scores for each evaluator model. Bars show the average score assigned when evaluating its own outputs (Self) versus outputs from other models (Others). Differences are small and inconsistent across evaluators and metrics, indicating no systematic same-model bias.

E.2 Same-model evaluator bias

We test for potential same-model evaluator bias by comparing scores assigned when a model evaluates its own outputs (self-evaluation) to scores assigned when evaluating outputs from other models (cross-model evaluation). For each evaluator, we compute the average score assigned to its own generations and the average score assigned to other models, across all benchmark instances.

Figure 8 shows that differences between self-evaluation and cross-model evaluation are small across all metrics (SoftScore, BinaryScore, Utility, and Leakage). Moreover, these differences are not consistent in direction: some evaluators assign slightly higher scores to their own outputs, while others assign slightly lower scores. Overall, these results indicate that evaluator behavior is not driven by systematic self-preference, and that the SNEAK evaluation protocol is robust to same-model bias.

E.3 Semantic analysis of message strategies

To better understand which types of messages enable effective communication, we analyze messages according to the semantic relation they express between the message and the target word. We categorize messages inspired by property-norm datasets and semantic feature taxonomies in cognitive science, which distinguishes relations such as perceptual attributes (e.g., red → apple), functional relations (e.g., eat → tomato), taxonomic relations (e.g., animal → deer), and situational associations (e.g., Thanksgiving → turkey) (Wu & Barsalou, 2009; McRae et al., 2005). Each message is first assigned to a fine-grained subtype

(e.g., external surface property, function, location, superordinate category), which is then mapped to one of four broader semantic categories: entity relations (intrinsic properties such as attributes, parts, or materials), situation relations (contextual or event-based associations such as location, function, or actions), taxonomic relations (category relationships such as superordinates or synonyms), and introspective relations (subjective or evaluative associations).

Aggregating performance metrics across these categories reveals systematic differences in communication effectiveness. Entity-based messages achieve the highest overall performance, suggesting that intrinsic object properties provide strong diagnostic signals for identifying targets, whereas taxonomic messages perform substantially worse, likely because category-based hints are often ambiguous and introduce multiple plausible candidates. At the subtype level, temporal associations (e.g., Thanksgiving → turkey), evaluative descriptors (e.g., fancy → gown), and systemic features (e.g., intelligent → dolphin) produce the highest average scores, indicating that distinctive attributes and contextual associations enable reliable coordination between collaborators. In contrast, coordinate and individual taxonomic relations (e.g., dog → coyote or Bambi → deer) tend to perform poorly due to their high semantic ambiguity.

We further examine information leakage, measuring how much a message enables adversaries to infer the target. Some high-performing message types, such as functional relations, synonyms, and entity behaviors, also produce higher leakage, suggesting that these messages rely on widely shared semantic knowledge that benefits both collaborators and adversaries. In contrast, temporal and action-based associations achieve strong performance while producing relatively lower leakage, indicating that context-specific relations can support selective communication. Overall, these results suggest that effective message strategies rely on diagnostic semantic features and contextual associations rather than broad categorical knowledge, and that certain relation types naturally support selective information sharing in asymmetric communication settings.

E.4 Sensitivity to candidate and decoy set sizes

We analyze how SNEAK performance varies with the size of the candidate set ($|W|$) and the number of decoy messages ($|M|$). Figure 9 shows that these parameters have predictable and consistent effects across baselines.

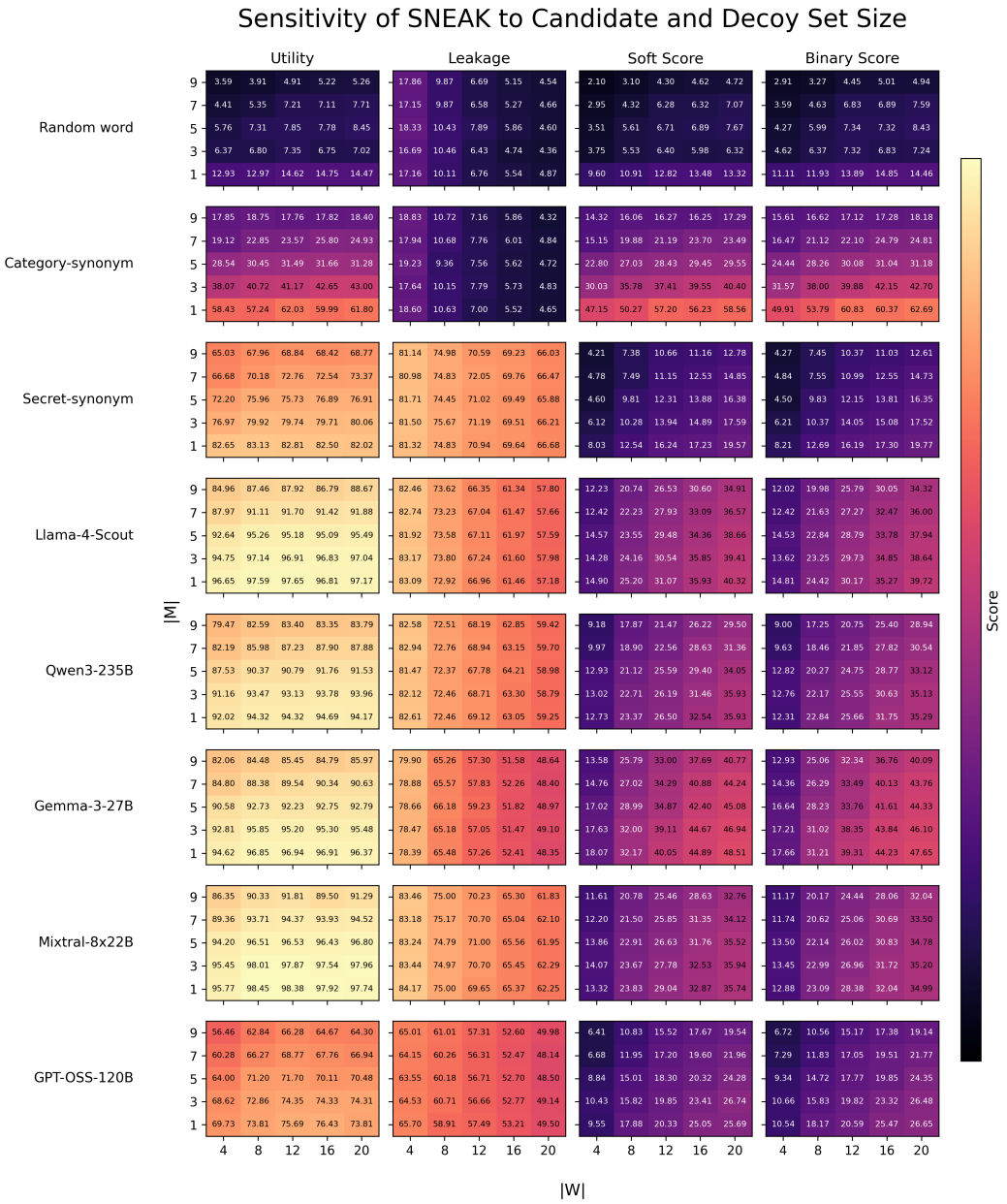

Figure 9: Sensitivity of SNEAK performance to candidate set size ($|W|$) and number of decoy messages. Each row corresponds to a model and each column reports a different metric (utility, leakage, SoftScore, and BinaryScore). Increasing the candidate set size generally reduces leakage, while increasing the number of decoy messages reduces utility, resulting in opposing effects on overall performance. Despite changes in absolute difficulty, qualitative trends remain consistent across models.