Do LLM Semantic Judgments Follow Distributional Geometry? Prompt and Model Effects in Controlled Category Probes

Anonymous ACL submission

Abstract

Embeddings and next-token probabilities are often treated as interchangeable proxies for semantic knowledge in large language models (LLMs), but they need not reflect the same underlying structure. We present a controlled test of when representational geometry aligns with semantic behavior. We construct a counterbalanced matched-alternative dataset in which each probe is evaluated against a related and unrelated candidate under identical formatting, enabling within-item comparisons across relation type (cohyponym vs. superordinate), distractor difficulty (easy vs. hard, defined by semantic confusability), prompting regimes, and model. We evaluate five LLMs using two scoring lenses: logprob preference and static embedding similarity. Across models, logprob discrimination is uniformly high with modest difficulty penalties. By contrast, embedding similarity systematically underestimates superordinate relations, with the gap largest under hard distractors, while cohyponym judgments remain near ceiling. Metaprompt scaffolding produces larger shifts than prompt wording. Overall, behavior–geometry alignment is relation- and difficulty-dependent, cautioning against treating embedding similarity as a model-agnostic measure of semantic knowledge.

1 Introduction

Large language models (LLMs) support striking semantic generalizations: they complete category statements, prefer semantically appropriate continuations, and often place related words near each other in internal representation space (Harris, 1954; Mikolov et al., 2013; Pennington et al., 2014). But it remains unclear what should count as evidence for "semantic structure" inside these systems. Two lenses are routinely used as proxies for meaning: (i) behavioral evidence from next-token probabilities (e.g., whether a model favors *bird* over *vehicle* after "An eagle is a type of . . . "), and (ii) representational evidence from embedding geometry (e.g., whether *eagle* is closer to *pigeon* than to *bus*). These are often treated as interchangeable, as if they were two views of a single competence. Yet they need not be: if they come apart, then "semantic knowledge" depends on what internal resources are being queried and how that query is posed (Rubin et al., 2014; Ethayarajh, 2019).

There are principled reasons to expect misalignment between global embedding geometry and context-specific semantic decisions. To predict the right continuation in a particular context, the relevant information may only need to be represented in a small subspace of a word's embedding, provided the model can condition on the prompt and effectively route computation through the subspace (Devlin et al., 2019; Peters et al., 2019; Ethayarajh, 2019). By contrast, embedding similarity aggregates broad distributional usage—useful for substitutability (Harris, 1954; Landauer and Dumais, 1997; Mikolov et al., 2013; Pennington et al., 2014)—but not guaranteed to encode asymmetric relations in a way recoverable by undirected similarity. This suggests a targeted contrast: within-category similarity (cohyponymy) is symmetric and plausibly "geometric," whereas superordinate "is-a" relations are asymmetric and are known to be harder to recover from undirected similarity, often requiring directional or pattern-based cues (Weeds et al., 2004; Baroni and Lenci, 2010; Roller and Erk, 2016; Shwartz et al., 2016). Misalignment should be most visible when distractors are semantically close, because coarse neighborhood structure is insufficient.

A central obstacle to understanding how the geometric semantic space is carved up and used is that apparent "semantic" effects are easily driven by confounds: frequency, lexical association, uneven distractor difficulty, and prompt idiosyncrasies. To isolate relation-specific competence, we use a tightly controlled, counterbalanced matched-

alternative design. Each trial evaluates the same probe against a related and unrelated candidate under identical formatting, so accuracy is whether the model assigns higher support to the related alternative. These matched comparisons are repeated across relation type (cohyponym vs. superordinate), difficulty (easy vs. hard distractors defined by semantic confusability), prompting regime (prompt wording and metaprompt scaffolding; cf. Mann et al., 2020; Wei et al., 2022; Liu et al., 2023), and model. Because the same probes participate across conditions, comparisons are effectively within-item rather than driven by shifts in item composition.

Using this factorial dataset, we ask three interlocking questions: (1) how does logprob-based discrimination align with embedding-geometry judgments, and does alignment differ by relation type and distractor difficulty (2) how sensitive are these judgments to prompt context (Mann et al., 2020; Wei et al., 2022; Liu et al., 2023)? and (3) how stable are the resulting patterns across GPT-2 (Radford et al., 2019a) and contemporary 8B-class families LLaMA (Touvron et al., 2023) and Qwen (Bai et al., 2023), including base versus instruction-tuned variants? We find systematic behavior–geometry dissociations concentrated in superordinate judgments and amplified by hard distractors, strong effects of metaprompt scaffolding relative to prompt wording, and model differences that are clearer in logprob than in static embedding similarity, with implications for how semantic "knowledge" is evaluated and interpreted in LLMs.

2 Related Work

Distributional semantics has long used geometric proximity as evidence for meaning, from the distributional hypothesis (Harris, 1954) through LSA-style factorization and related connectionist accounts (Elman, 1990; Landauer and Dumais, 1997) and predictive embeddings (Mikolov et al., 2013; Pennington et al., 2014). However, language-model behavior is produced by context-conditioned computation, so geometry and output can come apart depending on how representations are read out (Rubin et al., 2014) and on the layer/context dependence of contextual embeddings (Peters et al., 2019; Devlin et al., 2019; Ethayarajh, 2019).

This tension is especially relevant across relation types: symmetric similarity (cohyponymy) naturally aligns with the notion of neighborhood similarity, whereas asymmetric "is-a" relations (hypernymy/superordination) are harder to recover from undirected similarity and often require directional or pattern-based cues (Weeds et al., 2004; Baroni and Lenci, 2010; Roller and Erk, 2016; Shwartz et al., 2016). In parallel, prompt- and instruction-sensitivity work shows that elicited "knowledge" can shift markedly with context and scaffolding, suggesting that failures (or successes) may reflect retrieval and conditioning as much as stored structure (Mann et al., 2020; Wei et al., 2022; Liu et al., 2023).

While improvements in embedding geometry do not guarantee recoverability of relational structure, which have long been shown to depend critically on contextual cues rather than undirected similarity alone (Vyas and Carpuat, 2017), recent work has shown that embedding geometry nonetheless interacts strongly with architectural modifications (Feng et al., 2025), giving strong reasons to investigate how specific models handle the hypernymy and hyponymy relations.

Moreover, recent work on prompting demonstrates that the way prompts are represented in the embedding space can dramatically alter model behavior, whether by revealing that prompt input similarity does not reliably predict identical outputs without specialized tuning (Zhu et al., 2024) or by showing that gradient-based refinement of prompt embeddings can markedly improve reasoning performance (Hou et al., 2025), suggesting that prompt effects are themselves latent properties of how contextualized geometry interacts with task-conditioned computation.

Our study builds on these threads by testing when logprob-based behavior and embedding geometry converge versus systematically diverge across relation type and controlled distractor difficulty, using tightly counterbalanced stimuli.

3 Methods

3.1 Stimuli

Our stimuli consisted of prompts containing a *probe* word and two candidate *target* words. On each trial, models were evaluated on if they assigned a higher score to the semantically related target than an unrelated target, under either a cohyponym (within-category) or superordinate (category-membership) relationship. We describe below the construction of the items and how related and unrelated targets were selected.

Relationship	Probe	Related Target	Easy Unrelated Target	Hard Unrelated Target
Superordinate	fridge	appliance	mammal	musical instrument
Superordinate	freezer	appliance	royalty	clothing
Cohyponym	fridge	freezer	lion	clarinet
Cohyponym	freezer	fridge	queen	shirt

Table 1: Examples of probe–target pairings for cohyponym and superordinate relationships, with easy and hard unrelated targets.

3.1.1 Probe and Target Stimulus Words

We constructed a set of 192 probe words (8 words from each of 24 categories). The 24 categories spanned living and nonliving domains. Living categories included five animal categories (birds, mammals, reptiles, fish, insects), five people categories (royalty terms, family terms, romantic relationship terms, medical professional terms, entertainer professions), and two plant-food categories (fruits, vegetables). Nonliving categories included human-made artifacts (appliances, musical instruments, dishware, tools, weapons, clothing, furniture, toys, vehicles) and three food/drink categories (breakfast foods, desserts, beverages).

Each probe was paired with three target types under each of two relationship types, yielding a 2×3 design: (1) **Semantic relationship**: cohyponym vs. superordinate, and (2) **Target type**: related, easy-unrelated, and hard-unrelated. This procedure produced 1152 probe–target pairs in total ($192 \times 3 \times 2$). Examples are shown in Table 1.

Cohyponym pairings. For related cohyponym targets, the eight words within each category were paired to form four high-similarity within-category pairs (with each word appearing once as a probe and once as a related target). Unrelated cohyponym targets were selected in two ways. In the **easy** condition, each probe was paired with a target from a category in a different broad ontological class (living vs. nonliving), with the additional constraint that plant-food categories (fruits/vegetables) were not paired with the nonliving food/drink categories (breakfast foods/desserts/beverages). In the **hard** condition, each probe was paired with a target drawn from the same broad ontological class as the probe (animal/people/food/artifact) but from a different specific category (e.g., mammals paired with birds rather than with appliances).

Unrelated pairings were fully counterbalanced: every probe served as an easy-unrelated and hard-unrelated target for another probe. We also rotated cross-category mappings so that probes within a given category were distributed across multiple unrelated categories (e.g., the eight mammal probes were paired with eight different nonliving categories in the easy condition, and were distributed across animal categories in the hard condition), rather than concentrating all probes in a category onto a single unrelated category.

Superordinate pairings. Superordinate targets were derived from the cohyponym pairings by replacing the cohyponym target with its superordinate category label while preserving the unrelated targets. For example, if *fridge* was paired with *freezer* (related), *lion* (easy-unrelated), and *clarinet* (hard-unrelated) in the cohyponym condition, then the corresponding superordinate targets were *appliance*, *mammal*, and *musical instrument*. This construction ensures that cohyponym and superordinate conditions are matched on probe identity and on the unrelated targets used for comparison.

3.1.2 Prompt Construction

Next, we constructed prompts to evaluate whether models favored the related target over an unrelated target under different contextual constraints. In addition to the main prompt manipulations described below, we included a **unigram baseline** that contained neither the probe nor any relationship information (the prompt was a single colon, `:`). This baseline estimates each target's out-of-context probability (i.e., an effective unigram rate) and is not part of the factorial prompt design below.

Prompt and metaprompt design. All other prompts followed a 2×3 design crossing **prompt type** and **metaprompt type**. The **prompt type** specifies the local text immediately preceding the target position. The **metaprompt type** specifies whether an additional sentence of instructions precedes the prompt.

Prompt type. We used two prompt types: (i) a **probe-only (control)** prompt that included the probe but did not specify the semantic relationship (e.g., *eagle:*), and (ii) a **task-specific** prompt that instantiated a relationship-eliciting template (e.g., *An eagle is a type of* for superordinates; *An eagle is the same kind of thing as a* for cohyponyms). For each relationship type, we used five paraphrased task-specific templates (Appendix A). When templates required an indefinite article, we used the appropriate determiner (*a* vs. *an*); for mass nouns

we omitted determiners (e.g., *juice is a type of*).

Metaprompt type. We used three metaprompt types: (i) **none**, with no preceding instruction; (ii) **neutral**, which provided generic completion instructions ("Please complete the following sentence in a natural and fluent way in English. Respond as concisely as possible."); and (iii) **task-specific**, which provided relationship-specific guidance (co-hyponyms: "Please complete the following sentence about words and whether they belong to the same category."; superordinates: "Please complete the following sentence about the category label for the word that is provided."). Metaprompts were prepended to the prompt with a separating space.

Crossing the three metaprompt types with the two prompt types yields six metaprompt–prompt regimes. Because we used five paraphrases for each task-specific prompt template (per relationship), this produces 30 distinct metaprompt–prompt combinations (5 paraphrases $\times$ 2 relationships $\times$ 3 metaprompt types), plus the probe-only control prompts and the unigram baseline.

Formatting for base vs. instruction-tuned models. For base language models, we provided the constructed prompt string directly as model input. For instruction-tuned models, we embedded the same prompt content in the model's chat format by placing the metaprompt+prompt string in the user turn and evaluating candidate targets as the initial assistant continuation (Table 2).

3.2 Models

We evaluated three families of pretrained decoder-only Transformer language models: GPT-2 (Radford et al., 2019b), LLaMA-3.1 (Grattafiori et al., 2024), and Qwen-3 (Yang et al., 2025). For LLaMA-3.1 and Qwen-3, we included both base and instruction-tuned variants to assess the impact of alignment tuning on semantic behavior. Implementation details are provided in Appendix B.

GPT-2. We used the GPT-2 model configuration from Radford et al. (2019b). To integrate GPT-2 into our shared evaluation pipeline, we reimplemented the architecture and loaded the publicly released pretrained weights into the corresponding parameters of the custom implementation.

LLaMA-3.1 and Qwen-3. We evaluated 8B-parameter base and instruction-tuned variants from two contemporary decoder-only models: **LLaMA-3.1-8B-Base** and **LLaMA-3.1-8B-Instruct**, and **Qwen-3-8B-Base** and **Qwen-3-8B-Instruct**. All four models are trained with a next-token prediction objective; the instruction-tuned variants additionally undergo alignment tuning as described in their respective technical reports.

We selected these models to span (i) an early, smaller pre-alignment baseline (GPT-2) and (ii) two recent 8B-scale model families with both base and instruction-tuned checkpoints. This design allows us to test whether the measure-dependent patterns in Analyses 1-2 (Section 4) generalize across model families and across alignment status.

3.3 Evaluation Measures

All evaluations use a pairwise forced-choice criterion: for each probe, we compare a *related* candidate target to an *unrelated* candidate target and ask whether the model assigns a higher score to the related candidate. We consider two scoring measures: (i) summed token-level log probability of the candidate continuation under the prompt (logprob), and (ii) similarity between the probe and candidate *static* input-embedding vectors (embedding similarity). The unigram baseline prompt (`:`) contains neither the probe nor relationship-specifying context and serves as a sanity-check condition. We compute accuracy for this condition using the same forced-choice criterion; given the fully counterbalanced design (each target appears equally often as related and unrelated), expected accuracy is chance (0.5), which we observe. We exclude the unigram baseline from the main prompt $\times$ metaprompt analyses and report it only as a control check.

3.3.1 Sum Log-probability

For a candidate target string c tokenized as $c = (c_1, \ldots, c_T)$, we condition the model on the prompt context x and compute the log probability of generating the target as the sum of token log probabilities:

$$\log P(c \mid x) = \sum_{t=1}^{T} \log P(c_t \mid x, c_{<t}),$$

where $P(c_t \mid x, c_{<t})$ denotes the probability of token c_t given prompt x and preceding target tokens.

Because token log probabilities are summed, longer (multi-token) candidates will tend to receive lower total log probability. Our stimulus set and analyses mitigate this concern in two ways. First, targets are counterbalanced across relatedness roles within relationship types, reducing systematic association between token length and correctness. Second, our statistical models include random effects for items/targets, which absorb target-specific factors such as frequency and tokenization length.

Metaprompt	Prompt	Relationship	Example
None	Control	N/A	"eagle"
Neutral	Control	N/A	"Please complete the following sentence in a natural and fluent way in English. Respond as concisely as possible. eagle"
Task-specific	Control	Superordinate	"Please complete the following sentence about the category label for the word that is provided. Respond as concisely as possible. eagle"
Task-specific	Control	Cohyponym	"Please complete the following sentence about words and whether they belong to the same category. Respond as concisely as possible. eagle"
None	Task-specific	Superordinate	"an eagle is a type of"
None	Task-specific	Cohyponym	"an eagle is the same type of thing as a/an"
Neutral	Task-specific	Superordinate	"Please complete the following sentence in a natural and fluent way in English. Respond as concisely as possible. an eagle is a type of"
Neutral	Task-specific	Cohyponym	"Please complete the following sentence in a natural and fluent way in English. Respond as concisely as possible. an eagle is the same type of thing as a/an"
Task-specific	Task-specific	Superordinate	"Please complete the following sentence about the category label for the word that is provided. Respond as concisely as possible. an eagle is a type of"
Task-specific	Task-specific	Cohyponym	"Please complete the following sentence about words and whether they belong to the same category. Respond as concisely as possible. an eagle is the same type of thing as a/an"
Example of full prompt inserted into dialogue template for instruct-tuned models			**[user:]** Please complete the following sentence in a natural and fluent way in English. Respond as concisely as possible. {probe_det} {probe_noun} is a kind of {target_det} **[assistant:]**

Table 2: All combinations of metaprompt types and prompt templates used in our experiments, with examples.

3.3.2 Embedding similarity

To measure representational similarity independent of prompt context, we used the model's context-independent *input* embedding vectors. For a word w tokenized as $(w_1, \ldots, w_{|w|})$, we extract the corresponding input embeddings $E(w_i) \in \mathbb{R}^d$ and average across subword tokens to obtain a single vector representation. For a probe p and a candidate c, we compute the representations as:

$$\mathbf{v}(p) = \frac{1}{|p|} \sum_{i=1}^{|p|} E(p_i), \qquad \mathbf{v}(c) = \frac{1}{|c|} \sum_{i=1}^{|c|} E(c_i).$$

We then compute probe–candidate similarity using Pearson correlation.[1]

$$s(p, c) = \mathrm{corr}\big(\mathbf{v}(p), \mathbf{v}(c)\big),$$

3.3.3 Accuracy

For both measures, we compute trial-level accuracy using a forced-choice comparison. Given a probe p with related target c^+ and unrelated target c^-, a trial is scored as correct if the related candidate receives a higher score:

$$\mathbb{I}\big[\mathrm{score}(p, c^+) > \mathrm{score}(p, c^-)\big].$$

For logprob, $\mathrm{score}(p, c)$ is the summed log probability of c under the instantiated prompt. For embeddings, $\mathrm{score}(p, c)$ is the static similarity $s(p, c)$ computed from input embeddings.

[1]We use Pearson correlation for scale/offset invariance; cosine similarity yields the same qualitative patterns in our data (not shown).

3.4 Bayesian mixed-effects models

We analyzed trial-level accuracy ($acc \in \{0, 1\}$; whether related > unrelated) using Bayesian mixed-effects logistic regression in `bambi` (Bernoulli likelihood; logit link). Because prompt and metaprompt manipulations cannot affect static embedding similarities, we fit separate models for embedding-similarity and sum-logprob trials. Both models included fixed effects for relationship (cohyponym vs. superordinate), condition (easy vs. hard), and model, including their interactions. The logprob model also included metaprompt_type and prompt_type and their interaction, as well as their interactions with relationship. To account for repeated measures, we included random intercepts for probes nested in categories, and for related and unrelated targets. The logprob model further included a random intercept for prompt (prompt_group). We fit models with 4 chains, 1000 warmup iterations, and 1000 posterior draws per chain (target_accept=0.99; fixed seed). We report posterior predicted cell means and planned contrasts computed on posterior draws (94% credible intervals). See Appendix C for details.

4 Results

We report Bayesian mixed-effects logistic regression results predicting trial-level accuracy, defined as whether the related target received a higher score than the unrelated target in a pairwise forced-choice comparison. We summarize effects using posterior predicted accuracies for each experimental cell and

condition	relationship	Sum Log Probability			Embedding Similarity		
		CrI low	Mean	CrI high	CrI low	Mean	CrI high
Easy	Superordinate	0.985	0.992	0.998	0.908	0.955	0.992
	Cohyponym	0.984	0.990	0.995	0.987	0.993	0.998
Hard	Superordinate	0.960	0.979	0.994	0.803	0.900	0.978
	Cohyponym	0.970	0.981	0.991	0.981	0.990	0.997

Table 3: Posterior predicted accuracies (means and 94% credible intervals) by relationship and difficulty, shown separately for sum log probability and static embedding similarity.

planned contrasts computed on posterior draws; uncertainty is reported using 94% credible intervals.

We organize the results around three questions: (i) how accuracy differs across semantic relationships and difficulty as a function of measurement type (sum log probability vs. static embedding similarity), (ii) how sensitive logprob-based accuracy is to prompt and metaprompt structure, and (iii) whether the key patterns generalize across model families and instruction tuning.

Because prompt and metaprompt manipulations cannot affect static input-embedding similarities, embedding-similarity results are reported under the task-specific prompting regime and analyzed with a dedicated mixed-effects model fit to the embedding dataset. Prompt and metaprompt effects are evaluated only for the logprob measure using a separate mixed-effects model fit to the logprob dataset.

4.1 Analysis 1: Measure × Relationship × Difficulty

Our first analysis asked whether a model's static embedding geometry supports the same semantic distinctions that the model can reliably express via next-token prediction. We compared accuracy under two scoring measures, sum log probability (logprob) and static embedding similarity, across two semantic relationships (cohyponym vs. superordinate) and two difficulty conditions (easy vs. hard). To align the comparison across measures, we report both under the task-specific prompt + task-specific metaprompt regime. This avoids conflating logprob performance with prompt-structure variation (which cannot affect static embeddings) while placing logprob evaluation in the strongest, a priori best-case prompting context.

We summarize results using posterior predicted accuracies for each experimental cell (Table 3) and planned contrasts computed from posterior draws. The key qualitative pattern is a measure-dependent interaction: logprob-based accuracy is high and comparatively similar across relationship types, whereas embedding-similarity accuracy shows a pronounced deficit for superordinate judgments, especially under the hard condition.

For logprob, accuracy decreased from easy to hard for both relationships (cohyponym: $\Delta = 0.009$, 94% CrI $[0.004, 0.014]$; superordinate: $\Delta = 0.013$, 94% CrI $[0.004, 0.025]$). In contrast, there was little relationship gap at a fixed difficulty level: easy cohyponym vs. easy superordinate yields $\Delta = -0.002$ (94% CrI $[-0.011, 0.007]$), and hard cohyponym vs. hard superordinate yields $\Delta = 0.002$ (94% CrI $[-0.019, 0.023]$). Thus, under logprob in best-case prompting, cohyponym and superordinate statements are expressed with broadly comparable accuracy, with performance primarily modulated by difficulty.

For embedding similarity, the pattern differs sharply. The easy–hard contrast was small for cohyponyms ($\Delta = 0.003$, 94% CrI $[-0.001, 0.008]$), but superordinate judgments showed a substantially larger difficulty decrement ($\Delta = 0.055$, 94% CrI $[0.013, 0.106]$). More importantly, embedding-based accuracy was markedly lower for superordinates than cohyponyms at both difficulty levels (easy: $\Delta = 0.038$, 94% CrI $[0.003, 0.082]$; hard: $\Delta = 0.090$, 94% CrI $[0.018, 0.183]$). This dissociation suggests that static embedding geometry provides a better proxy for within-category similarity (cohyponyms) than for category-membership judgments (superordinates), even when both relations can be expressed accurately by logprob.

Taken together, these results support the view that next-token prediction can retrieve both relationship types reliably in an appropriate prompting context, while static embedding similarity is an uneven proxy for semantic knowledge—performing well for cohyponyms but degrading for superordinate judgments, particularly when distractors are semantically closer.

4.2 Analysis 2: Effects of Prompt Context

Our second analysis tested how sensitive logprob-based retrieval of semantic relations is to prompt structure. Because static input-embedding similarities are invariant to prompt and metaprompt wording, this analysis uses only the sum-logprob measure. We fit the logprob mixed-effects model described in Section 3.4, predicting trial accuracy

Relationship Diff.	Prompt	Metaprompt	p [94%]
Cohyponym			
easy	control	neutral	0.982 [0.970, 0.990]
hard	control	neutral	0.965 [0.942, 0.981]
easy	control	none	0.965 [0.943, 0.981]
hard	control	none	0.935 [0.897, 0.963]
easy	control	task-specific	0.984 [0.974, 0.991]
hard	control	task-specific	0.970 [0.949, 0.983]
easy	task-spec.	neutral	0.993 [0.988, 0.996]
hard	task-spec.	neutral	0.986 [0.977, 0.992]
easy	task-spec.	none	0.988 [0.979, 0.993]
hard	task-spec.	none	0.977 [0.961, 0.987]
easy	task-spec.	task-specific	0.990 [0.983, 0.995]
hard	task-spec.	task-specific	0.981 [0.968, 0.989]
Superordinate			
easy	control	neutral	0.965 [0.918, 0.989]
hard	control	neutral	0.917 [0.824, 0.972]
easy	control	none	0.947 [0.879, 0.983]
hard	control	none	0.880 [0.759, 0.957]
easy	control	task-specific	0.988 [0.971, 0.997]
hard	control	task-specific	0.969 [0.927, 0.991]
easy	task-spec.	neutral	0.985 [0.968, 0.995]
hard	task-spec.	neutral	0.962 [0.923, 0.986]
easy	task-spec.	none	0.980 [0.956, 0.993]
hard	task-spec.	none	0.948 [0.896, 0.981]
easy	task-spec.	task-specific	0.992 [0.983, 0.997]
hard	task-spec.	task-specific	0.979 [0.955, 0.992]

Table 4: Posterior predicted accuracies (sum logprob) by relationship, difficulty, prompt type, and metaprompt type (94% interval).

from metaprompt type (none, neutral, task-specific) and prompt type (control vs. task-specific), while retaining relationship (cohyponym vs. superordinate), difficulty (easy vs. hard), and model, and including random intercepts for probe, prompt, target, and comparison. We summarize prompt effects using posterior predicted accuracies for each metaprompt–prompt cell (Table 4) and planned contrasts computed from posterior draws, stratified by relationship and difficulty. Unless otherwise noted, cell means and contrasts are marginal over model (averaging equally across the five models).

Across all four relationship–difficulty conditions, adding a neutral metaprompt improved performance relative to having no metaprompt under control prompting (Contrast 1; cohyponym–easy: $\Delta = 0.017$, 94% CrI $[0.008, 0.025]$; cohyponym–hard: $\Delta = 0.030$, 94% CrI $[0.017, 0.046]$; superordinate–easy: $\Delta = 0.018$, 94% CrI $[0.004, 0.035]$; superordinate–hard: $\Delta = 0.037$, 94% CrI $[0.012, 0.063]$). Thus, even relationship-neutral scaffolding increases the probability models favor the related over unrelated candidate, with larger gains under the hard condition.

Beyond this scaffolding effect, task-specific metaprompts provided additional benefits, particularly for superordinates. Under control prompts, task-specific metaprompts outperformed neutral metaprompts in both superordinate conditions (Contrast 2; superordinate–easy: $\Delta = -0.023$, 94% CrI $[-0.045, -0.004]$; superordinate–hard: $\Delta = -0.052$, 94% CrI $[-0.096, -0.015]$; negative values indicate an advantage for task-specific over neutral in this contrast parameterization). For cohyponyms, the corresponding effects were smaller but still reliably favored task-specific guidance (cohyponym–easy: $\Delta = -0.003$, 94% CrI $[-0.005, -0.001]$; cohyponym–hard: $\Delta = -0.005$, 94% CrI $[-0.009, -0.001]$).

Under task-specific prompts, neutral metaprompts again improved over none (Contrast 3), but the incremental effect of upgrading from neutral to task-specific metaprompts depended on the relation (Contrast 4): it was positive for cohyponyms (cohyponym–easy: $\Delta = 0.003$, 94% CrI $[0.001, 0.005]$; cohyponym–hard: $\Delta = 0.005$, 94% CrI $[0.002, 0.009]$) and negative for superordinates (superordinate–easy: $\Delta = -0.007$, 94% CrI $[-0.013, -0.002]$; superordinate–hard: $\Delta = -0.017$, 94% CrI $[-0.031, -0.005]$), indicating larger gains from task-specific metaprompts for superordinates.

Finally, we examined if switching from control to task-specific prompts provided additional improvements within each metaprompt regime (Contrasts 5–7). These effects were strongest when metaprompt structure was minimal: with no metaprompt, task-specific prompts improved performance for cohyponyms in both difficulty conditions (Contrast 5; cohyponym–easy: $\Delta = -0.023$, 94% CrI $[-0.041, -0.008]$; cohyponym–hard: $\Delta = -0.041$, 94% CrI $[-0.073, -0.015]$), and showed a similar trend for superordinates under hard difficulty (superordinate–hard: $\Delta = -0.069$, 94% CrI $[-0.158, -0.001]$). However, once metaprompt scaffolding was present, the incremental value of task-specific prompt wording was reduced and often not credibly different from zero. Overall, prompt sensitivity is driven by metaprompt-level scaffolding and task guidance, with prompt-level task wording contributing most when higher-level scaffolding is absent.

4.3 Analysis 3: Model Comparisons

Analysis 3 asks whether the core three-way pattern from Analysis 1 generalizes across individual models when we hold prompting fixed and examine

each model separately. In all models except GPT-2, superordinate embedding similarity is lower than the other three conditions and this deficit amplifies with difficulty. GPT-2 instead shows stronger cohyponym embedding similarity, weaker performance on the remaining conditions.

The three-way pattern holds consistently across newer models, with Qwen outperforming LLaMA, while instruction tuning improves superordinate embedding similarity but hurts cohyponym similarity. For a detailed breakdown of the LLM model comparison, we refer the readers to Appendix D.

5 Discussion and Conclusion

LLMs can appear to exhibit coherent "semantic knowledge" under multiple lenses, but our results show these lenses do not collapse onto a single structure. Using a tightly controlled, counterbalanced matched-alternative design, we observed systematic dissociations between behavioral discrimination (which candidate receives higher next-token support) and representational discrimination measured by static embedding similarity. Importantly, the divergence is structured rather than uniform: it concentrates on superordinate judgments rather than cohyponym judgments, and is most visible under hard distractor conditions. This supports a representational interpretation: whether a model "knows" a relation depends on which internal resources are queried and how that query is posed, not on a single context-invariant semantic faculty.

A natural mechanistic explanation is that accurate contextual prediction need not be supported by globally recoverable geometry. For a given prompt, the model may rely on a small part of its internal representation to choose the correct completion, as long as it can use the prompt to focus computation on the information that matters for the task. By contrast, similarity geometry aggregates broad regularities of distributional usage (useful for substitutability) but not guaranteed to encode asymmetric relations in a form recoverable by undirected proximity. The hard-distractor manipulation is diagnostic because it pushes beyond coarse neighborhood structure: when unrelated targets are semantically close, a purely "geometric" readout has less room to succeed. The increased superordinate gap under difficulty therefore suggests that behavioral success can be supported by context-conditioned, relation-specific computation even when static geometry remains a weak proxy.

The prompting results reinforce this interpretation by distinguishing changes in elicitation from changes in stored content. Higher-level metaprompt scaffolding shifts performance more reliably than surface differences in prompt wording, consistent with prompting primarily modulating what features are activated, which constraints are emphasized, and which inference-like computations are recruited rather than "adding" semantic information. This matters for interpretation: behavior under a particular prompt is not a direct window onto a single representation, but the outcome of prompt-conditioned computation over distributed resources. From this perspective, prompt sensitivity is not merely noise to average away; it helps identify which internal pathways can support a given semantic judgment.

Across models, we observe both stability and heterogeneity. The geometry behavior dissociation appears across GPT-2 and 8B-class families (LLaMA and Qwen), but the magnitude and structure of effects vary with model family and alignment status (base vs. instruction-tuned). Instruction tuning can reshape next-token behavior, sometimes yielding systematic advantages for particular relation types without making static embedding geometry a reliable proxy. This has practical consequences for model comparison: behavioral improvements need not be mirrored by changes in similarity structure, and a single proxy need not yield a stable ranking across relation types or contexts.

Taken together, these findings support a view in which distributional geometry and context-conditioned computation play complementary but non-identical roles. Embeddings capture broad usage-based similarity that aligns with symmetric relations like cohyponymy, while superordinate structure can be expressed behaviorally via prompt-sensitive constraints and directional or pattern-based information that is not necessarily recoverable from undirected proximity alone. Methodologically, the results underscore the value of counterbalanced evaluation: by holding lexical content constant while manipulating relation type, confusability, and prompting factorially, divergences can be attributed to principled representational factors rather than item composition or frequency artifacts. Conceptually, they caution against treating embedding similarity as a uniform surrogate for "semantic knowledge," and motivate evaluations and theories that distinguish what is represented globally from what can be computed reliably in context.

6 Limitations

Static vs. dynamic embeddings. We focus on static embeddings rather than dynamic, context-dependent embeddings. There is no strong *a priori* reason to expect dynamic embeddings to show qualitatively different effects, as the theoretical motivation for our predictions does not depend on context sensitivity. Nevertheless, this remains a limitation until tested directly, and future work should examine whether the same patterns hold when embeddings are derived from contextualized representations.

Language sample. Our experiments are limited to English and to a fixed set of 24 semantic categories. There is no *a priori* reason to expect the observed effects to be language-specific or restricted to this particular category set. However, it would be valuable to test whether the same patterns generalize across languages and across broader or differently structured semantic domains.

Semantic relationships. We focus on superordinate versus cohyponym relations because they provide the strongest test of our hypothesis. If there was going to be a case where we might expect convergence between logits and embeddings, with would be with category labels and their instances. However this work could be extended and strengthened by testing a greater range of semantic relationships.

Model coverage. Our goal is to characterize a principle that should hold across large language models in a general way, rather than being tied to a specific architecture. To this end, we evaluate five models that differ in design and training. Nonetheless, the space of contemporary LLMs is large, and additional models—particularly those with alternative training objectives or representation schemes—could further test the generality of the observed effects.

7 Acknowledgements

ChatGPT was used to take a rough draft of the text of the paper, and provide suggestions for paraphrasing and polishing. It was also used for help with conversion of table data to Latex format. All text provided by the tool was carefully checked for accuracy.

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Marco Baroni and Alessandro Lenci. 2010. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science*, 14(2):179–211.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations. *Comparing the geometry of BERT, ELMo, and GPT-2 Embeddings*, 2.

Zhaoxin Feng, Jianfei Ma, Emmanuele Chersoni, Xiaojing Zhao, and Xiaoyi Bao. 2025. Learning to look at the other side: A semantic probing study of word embeddings in LLMs with enabled bidirectional attention. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23226–23245, Vienna, Austria. Association for Computational Linguistics.

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. *Preprint*, arXiv:2407.21783.

Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

Xiaoming Hou, Jiquan Zhang, Zibin Lin, DaCheng Tao, and Shengli Zhang. 2025. Embedgrad: Gradient-based prompt optimization in embedding space for large language models. *Preprint*, arXiv:2508.03533.

Thomas K Landauer and Susan T Dumais. 1997. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM computing surveys*, 55(9):1–35.

Ben Mann, Nick Ryder, Melanie Subbiah, J Kaplan, P Dhariwal, A Neelakantan, P Shyam, G Sastry,

A Askell, S Agarwal, and 1 others. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 1(3):3.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. 2019. Knowledge enhanced contextual word representations. 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)*, pages 43–54.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019a. Language models are unsupervised multitask learners. Technical report, OpenAI. OpenAI Technical Report.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019b. Language models are unsupervised multitask learners. *OpenAI*. Accessed: 2024-11-15.

Stephen Roller and Katrin Erk. 2016. Relations such as hypernymy: Identifying and exploiting hearst patterns in distributional vectors for lexical entailment. *arXiv preprint arXiv:1605.05433*.

Timothy N Rubin, Brent Kievit-Kylar, Jon A Willits, and Michael N Jones. 2014. Organizing the space and behavior of semantic models. In *Cogsci... annual conference of the cognitive science society. cognitive science society (us). conference*, volume 2014, page 1329.

Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving hypernymy detection with an integrated path-based and distributional method. *arXiv preprint arXiv:1603.06076*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Yogarshi Vyas and Marine Carpuat. 2017. Detecting asymmetric semantic relations in context: A case-study on hypernymy detection. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 33–43, Vancouver, Canada. Association for Computational Linguistics.

Julie Weeds, David Weir, and Diana McCarthy. 2004. Characterising measures of lexical distributional similarity. In *COLING 2004: Proceedings of the 20th international conference on Computational Linguistics*, pages 1015–1021.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Hanlin Zhu, Banghua Zhu, and Jiantao Jiao. 2024. Efficient prompt caching via embedding similarity. *Preprint*, arXiv:2402.01173.

A Metaprompts and prompt templates

Condition	Metaprompt
Task-biased	Please complete the following sentence about the category label for the word that is provided. Respond as concisely as possible.
Neutral	Please complete the following sentence in a natural and fluent way in English. Respond as concisely as possible.
None	—

Table 5: Metaprompts used for the superordinate task. In the control condition, no metaprompt was prepended to the prompt template.

Condition	Metaprompt
Task-biased	Please complete the following sentence about words and whether they belong to the same category. Respond as concisely as possible.
Neutral	Please complete the following sentence in a natural and fluent way in English. Respond as concisely as possible.
None	—

Table 6: Metaprompts used for the cohyponym task. In the control condition, no metaprompt was prepended to the prompt template.

Type	Prompt template
Task-specific1	`{PROBE_DETERMINER} {PROBE} is {TARGET_DETERMINER} {TARGET}.`
Task-specific2	`{PROBE_DETERMINER} {PROBE} is a kind of {TARGET}.`
Task-specific3	`{PROBE_DETERMINER} {PROBE} is a type of {TARGET}.`
Task-specific4	`{PROBE_DETERMINER} {PROBE} belongs to the category {TARGET}.`
Task-specific5	`{PROBE_DETERMINER} {PROBE} is classified as {TARGET_DETERMINER} {TARGET}.`
Control1	`{PROBE} {TARGET}.`
Control2	`{PROBE}: {TARGET}.`
Control3	`{PROBE} -> {TARGET}.`
Control4	`{PROBE} — {TARGET}.`
Control5	`{PROBE} and {TARGET}.`

Table 7: Prompt templates used for the superordinate task. {PROBE} denotes the probe word and {TARGET} denotes the category; determiner slots are optional and omitted when not applicable.

Type	Prompt template
Task-specific1	`{PROBE_DETERMINER} {PROBE} is like {TARGET_DETERMINER} {TARGET}.`
Task-specific2	`{PROBE_DETERMINER} {PROBE} is similar to {TARGET}.`
Task-specific3	`Two words that belong to the same category are {PROBE} and {TARGET}.`
Task-specific4	`Another word that belongs to the same category as {PROBE} is {TARGET}.`
Task-specific5	`{PROBE} is the same type of thing as {TARGET}.`
Control1	`{PROBE} {TARGET}.`
Control2	`{PROBE}: {TARGET}.`
Control3	`{PROBE} -> {TARGET}.`
Control4	`{PROBE} — {TARGET}.`
Control5	`{PROBE} and {TARGET}.`

Table 8: Prompt templates used for the cohyponym task. {PROBE} denotes the probe word and {TARGET} denotes the target noun that is an instance of a category; determiner slots are optional and omitted when not applicable.

B LLM Implementation Details

GPT-2: We implemented a decoder-only Transformer language model from scratch in PyTorch, following the GPT-2 base architecture. The model uses a vocabulary size of 50,257, an embedding dimension of 768, a maximum context length of 1,024 tokens, 12 Transformer layers, and 12 self-attention heads per layer. Each Transformer block consists of masked multi-head self-attention followed by a position-wise feed-forward network, with residual connections and layer normalization applied in the standard GPT-2 configuration.

Tokenization followed the GPT-2 Byte Pair Encoding (BPE) scheme. We used the `GPT2TokenizerFast` tokenizer from the Hugging Face Transformers library, which is identical to the tokenizer used in the original GPT-2 model.

To ensure architectural correctness and facilitate controlled comparisons, we initialized our implementation by copying parameters from the pretrained GPT-2 checkpoint provided by Hugging Face (`GPT2LMHeadModel.from_pretrained("gpt2")`). Pretrained weights were transferred via an exact one-to-one mapping between corresponding modules, including token and positional embeddings, attention projection matrices, feed-forward layers, and layer normalization parameters. The language modeling head shared weights with the token embedding matrix, following the GPT-2 weight tying scheme. After weight transfer, the custom implementation was functionally equivalent to the reference GPT-2 model under identical tokenization and context length settings. All evaluations for the GPT-2 model were conducted on a MacBook equipped with an Apple M1 Max processor using CPU inference.

LLaMA-3.1 and Qwen-3: For the LLaMA-3.1 and Qwen-3 models, we used the Unsloth framework to load and evaluate pretrained decoder-only Transformer models

efficiently. Models were instantiated via `FastLanguageModel.from_pretrained`, with a maximum sequence length of 512 tokens. Models were loaded using 4-bit quantization and `bfloat16` computation.

All evaluations for the LLaMA and Qwen models were conducted using PyTorch 2.1.2 and Python 3.10 on an Ubuntu 22.04 system. Experiments were run on a single NVIDIA RTX 4090 GPU with 24 GB of memory, accompanied by 16 vCPUs (Intel Xeon Platinum 8358P at 2.60 GHz). CUDA 11.8 was used for GPU acceleration.

C Appendix: Bayesian model details

We analyzed trial-level accuracy ($acc \in \{0, 1\}$) using Bayesian mixed-effects logistic regression implemented in `bambi` (Bernoulli likelihood; logit link). We fit separate models to the embedding-similarity and sum-logprob datasets.

Reference levels and grouping variables. We enforced consistent treatment-coding references by setting factor orders as follows: `model` ∈ {gpt2, llama3.1_8b_base, llama3.1_8b_instruct, qwen3_8b_base, qwen3_8b_instruct} (reference: gpt2), `relationship` ∈ {cohyponym, superordinate} (reference: cohyponym), `condition` ∈ {easy, hard} (reference: easy), `meta_prompt_type` ∈ {neutral, none, task specific} (reference: neutral; chosen so treatment coding yields the planned contrasts none vs. neutral and task-specific vs. neutral), and `prompt_type` ∈ {control, task specific} (reference: control). We created `probe_group` as `probe_category:probe`. For logprob analyses, we created `prompt_group` from `prompt_key`, nesting task-specific prompts within relationship (control prompts use `prompt_key`; task-specific prompts use `relationship:prompt_key`).

Embedding-similarity dataset preprocessing. Because embedding similarity is invariant to prompt and metaprompt wording, rows that differed only in `prompt_type`, `meta_prompt_type`, and `prompt_group` were redundant. We therefore collapsed the embedding-similarity dataset by dropping duplicates after removing these fields, leaving one row per unique combination of model, relationship, condition, probe_group, target, comparison, measure, and accuracy.

Model specifications. For embedding similarity, we fit:

$$\begin{aligned} acc \sim\ & relationship * condition + relationship * model \\ & + condition * model + (1|probe\_group) \\ & + (1|target) + (1|comparison). \end{aligned} \tag{1}$$

For sum log probability, we fit:

$$\begin{aligned} acc \sim\ & relationship * condition + relationship * model \\ & + condition * model \\ & + meta\_prompt\_type * prompt\_type \\ & + relationship : meta\_prompt\_type \\ & + relationship : prompt\_type \\ & + (1|probe\_group) + (1|prompt\_group) \\ & + (1|target) + (1|comparison). \end{aligned} \tag{2}$$

Inference and summaries. We fit each model with 4 chains, 1000 warmup iterations, and 1000 posterior draws per chain (`target_accept=0.99`; fixed random seed), and stored pointwise log-likelihood in the returned `InferenceData`. Priors were `bambi`'s defaults for logistic mixed-effects models. We report posterior predicted cell means and planned contrasts computed from posterior draws, summarizing uncertainty with 94% credible intervals.

D Appendix: Analysis 3 Specific LLM Comparison Details

Analysis 3 serves two complementary goals. First, it tests whether the key dissociation from Analysis 1 generalizes across individual models: logprob-based accuracy shows primarily a difficulty effect with little separation between cohyponym and superordinate relations, whereas embedding-similarity accuracy exhibits a relation-specific deficit (superordinate $<$ cohyponym) that is amplified in the hard condition. This model-by-model check is important because the Analysis 1 interaction was estimated after marginalizing across model family and alignment status; if the interaction reflects a general property of LLM representations rather than an artifact of a particular architecture or training regime, it should be observable within each model when we hold prompting fixed.

Second, Analysis 3 provides a targeted set of model comparisons under a best-case prompting regime, addressing how semantic discrimination differs (i) between an older pretrained transformer (GPT-2) and contemporary 8B base models, (ii) between base and instruction-tuned variants within a family, and (iii) between the LLaMA and Qwen

families. These comparisons are not intended as an exhaustive benchmark, but rather as a structured "sanity check" on whether the broad patterns in Analyses 1–2 depend on model vintage, instruction tuning, or model family. Because prompt sensitivity (Analysis 2) can itself differ across models, all logprob-based comparisons in Analysis 3 are restricted to the task-specific prompt × task-specific metaprompt condition (matching Analysis 1), and embedding-based accuracy is computed from the corresponding static embeddings. We summarize Analysis 3 using posterior predicted accuracies for each model × relationship × difficulty × measure cell (Figure 1) and planned contrasts computed from posterior draws (Analyses 3a–3b; see contrast files described in the text below).

D.1 Replicating Analysis 1 Across Models

(2) Replication of the Analysis 1 interaction within each model (Analysis 3a). Analysis 3a asks whether the core three-way pattern from Analysis 1 generalizes across individual models when we hold prompting fixed (task-specific prompt × task-specific metaprompt) and examine each model separately. Operationally, we replicate the Analysis 1 "adjacent-cell" comparisons *within each model* and *within each measure*: (i) easy vs. hard within cohyponyms, (ii) cohyponym vs. superordinate within easy, (iii) cohyponym vs. superordinate within hard, and (iv) easy vs. hard within superordinates. We compute these four contrasts for sum-logprob and for embedding similarity (eight total per model; 40 total across the five models), using planned contrasts computed from posterior draws (Analysis 3a contrasts) and referencing the corresponding posterior predicted cell means (Figure 1).[2]

Overall structure of the within-model results. Across models, the within-model contrasts show a mixture of stable and model-specific effects. Two qualitative regularities are visible in the contrast patterns. First, relationship differences are often larger for embedding similarity than for logprob in the cells where a relationship gap is clearly expressed (i.e., HDIs excluding zero). Second, the degree to which difficulty amplifies relationship differences depends on the model family and alignment status, rather than appearing as a uniform property of all five models. As a result, the pooled interaction from Analysis 1 is best viewed as a robust aggregate pattern that is expressed strongly in some models and weakly (or differently) in others, rather than a perfectly invariant signature that reproduces with equal strength in every model.

GPT-2: measure-dependent superordinate penalty, stronger under hard difficulty. For GPT-2, the measure × relationship × difficulty dissociation is qualitatively present. Under logprob scoring, accuracy is high in the easy condition (cohyponym: 0.981; superordinate: 0.980) and decreases under hard distractors (0.958 and 0.937), indicating a modest difficulty penalty for both relationships (easy–hard: $\Delta = 0.023$ for cohyponyms, 94% HDI $[0.012, 0.037]$; $\Delta = 0.043$ for superordinates, 94% HDI $[0.013, 0.080]$). At a fixed difficulty level, GPT-2 shows little evidence of a cohyponym–superordinate gap under logprob scoring (easy: $\Delta = 0.002$, 94% HDI $[-0.018, 0.024]$; hard: $\Delta = 0.021$, 94% HDI $[-0.028, 0.087]$).

Under embedding similarity, cohyponym accuracy remains near ceiling in both difficulty conditions (0.995 vs. 0.994; $\Delta = 0.001$, 94% HDI $[-0.002, 0.005]$). By contrast, superordinate accuracy is lower than cohyponym accuracy in both easy and hard conditions, with stronger evidence in the hard condition (easy cohyponym–superordinate: $\Delta = 0.017$, 94% HDI $[-0.000, 0.041]$, $p(\Delta > 0) = 0.995$; hard: $\Delta = 0.034$, 94% HDI $[0.003, 0.077]$, $p(\Delta > 0) = 0.999$). Thus, GPT-2 reproduces the central pattern that static embedding geometry is a weaker proxy for superordinate (category-membership) structure than for within-category similarity, and that this superordinate penalty is larger under harder distractors. In contrast to embeddings, logprob accuracy remains high and is primarily modulated by difficulty rather than relationship type.

Contemporary base models: large logprob relationship gaps in easy, but divergent embedding patterns. For both contemporary *base* models, the logprob relationship contrast in the easy condition is large and credibly different from zero (LLaMA-base: $\Delta = -0.782$, 94% HDI $[-0.983, -0.497]$; Qwen-base: $\Delta = -0.790$, 94% HDI $[-0.982, -0.482]$), indicating a strong separation between cohyponym and superordinate performance under best-case prompting even before considering difficulty. However, the embedding-similarity relationship effects differ across the two bases. Qwen-base shows a strong embedding relationship contrast in both easy and hard (easy: $\Delta =$

[2] Throughout, contrasts are reported as $p(\text{cell A}) - p(\text{cell B})$ with 94% HDIs, alongside posterior mass above/below zero.

-0.551, 94% HDI $[-0.994, -0.032]$; hard: $\Delta = -0.643$, 94% HDI $[-0.947, -0.306]$), whereas LLaMA-base shows weak and non-credible embedding relationship contrasts (easy: $\Delta = -0.075$, 94% HDI $[-0.872, 0.682]$; hard: $\Delta = 0.047$, 94% HDI $[-0.391, 0.572]$). In other words, *the base models do not show a uniform embedding-geometry signature*: one base model (Qwen) expresses a strong relationship separation in embeddings, while the other (LLaMA) does not, despite both showing strong separation under logprob in easy trials.

Instruction-tuned models: attenuated easy relationship gaps for logprob, with heterogeneous embedding effects. For the instruct variants, the easy-condition logprob relationship contrasts are comparatively small and not credibly different from zero (LLaMA-instruct: $\Delta = -0.123$, 94% HDI $[-0.571, 0.352]$; Qwen-instruct: $\Delta = -0.079$, 94% HDI $[-0.561, 0.385]$). In the hard condition, LLaMA-instruct shows a strong positive relationship contrast under logprob ($\Delta = 0.614$, 94% HDI $[0.317, 0.884]$), while Qwen-instruct shows a smaller positive relationship contrast that is also credibly different from zero ($\Delta = 0.392$, 94% HDI $[0.010, 0.782]$). Embedding-similarity relationship contrasts in the instruct models are generally weak and non-credible (LLaMA-instruct: hard $\Delta = -0.002$, 94% HDI $[-0.838, 0.809]$; Qwen-instruct: easy $\Delta = -0.067$, 94% HDI $[-0.885, 0.678]$; hard $\Delta = 0.450$, 94% HDI $[-0.114, 0.988]$). Thus, *instruction tuning tends to reduce clear relationship separation in embeddings*, while relationship separation under logprob becomes more contingent on difficulty and model family.

Difficulty modulation: limited evidence for a single, invariant amplification pattern across all models. A central component of the Analysis 1 interaction is that hard trials amplify the relationship-dependent disadvantage expressed by embedding similarity. In Analysis 3a, evidence for systematic hard-condition amplification is mixed. For example, Qwen-base exhibits strong embedding relationship contrasts in both easy and hard, consistent with an embedding-dependent relationship separation that persists across difficulty. GPT-2 shows a particularly strong embedding relationship contrast in the hard condition, but the corresponding easy embedding contrast is less decisive, consistent with difficulty amplification in that model. By contrast, both LLaMA variants show weak embedding relationship contrasts in both easy and hard, suggesting that the difficulty-amplified embedding signature is not uniformly present across all model families.

Interim conclusion for Goal 1. Taken together, the within-model analyses partially support the generalization claim from Analysis 1, but they also reveal meaningful heterogeneity. Some models (notably GPT-2 and Qwen-base) express strong relationship separation under embedding similarity, most clearly in hard trials, whereas others (notably the LLaMA variants) show weak or non-credible embedding relationship separation despite clear relationship effects in logprob for some cells. Thus, the pooled three-way interaction in Analysis 1 is not driven by a single model, but neither does it reproduce as a uniform, equally strong signature in all five models under the best-case prompting regime.

LLaMA-3.1-8B-Base: strong replication of the measure-dependent superordinate penalty. For LLaMA-3.1-8B-Base, logprob accuracy is near ceiling across relationships in the easy condition (cohyponym: 0.996; superordinate: 0.993) and shows only a small decrement under hard distractors (cohyponym easy–hard: $\Delta = 0.002$, 94% HDI $[0.001, 0.003]$; superordinate easy–hard: $\Delta = 0.008$, 94% HDI $[0.002, 0.016]$). At fixed difficulty, there is little evidence of a reliable cohyponym–superordinate difference under logprob scoring (easy: $\Delta = 0.004$, 94% HDI $[-0.003, 0.011]$; hard: $\Delta = 0.010$, 94% HDI $[-0.001, 0.027]$), consistent with the aggregate finding that relationship effects are small for logprob under best-case prompting.

In contrast, embedding similarity exhibits a clear and difficulty-amplified superordinate penalty. Cohyponym embedding accuracy remains near ceiling (easy: 0.997; hard: 0.995), whereas superordinate embedding accuracy is lower even in the easy condition (0.968) and drops sharply in the hard condition (0.905). Correspondingly, the cohyponym–superordinate gap is credibly positive at both difficulty levels and increases substantially under hard distractors (easy: $\Delta = 0.030$, 94% HDI $[0.004, 0.065]$; hard: $\Delta = 0.090$, 94% HDI $[0.015, 0.185]$), and superordinates show a reliable easy–hard decrement ($\Delta = 0.063$, 94% HDI $[0.011, 0.130]$). Thus, LLaMA-base strongly reproduces the core dissociation: logprob-based semantic discrimination is uniformly high with only modest difficulty effects, while static embedding geometry supports cohyponym similarity far bet-

ter than superordinate structure, especially under harder distractors.

Qwen-3-8B-Base: the same dissociation, with an even larger superordinate embedding drop. Qwen-3-8B-Base shows the same overall pattern. Under logprob scoring, accuracy is near ceiling in easy conditions (cohyponym: 0.996; superordinate: 0.993) with a small but reliable difficulty decrement (cohyponym easy–hard: $\Delta = 0.002$, 94% HDI $[0.001, 0.003]$; superordinate easy–hard: $\Delta = 0.007$, 94% HDI $[0.001, 0.013]$). As with LLaMA-base, relationship differences under logprob are small and not credibly different from zero at either difficulty level (easy cohyponym–superordinate: $\Delta = 0.003$, 94% HDI $[-0.003, 0.010]$; hard: $\Delta = 0.008$, 94% HDI $[-0.003, 0.021]$).

Embedding similarity again diverges: cohyponym embedding accuracy is high (easy: 0.989; hard: 0.982), but superordinate embedding accuracy is markedly lower (easy: 0.938) and drops further under hard distractors (0.863). The cohyponym–superordinate gap is credibly positive at both difficulty levels (easy: $\Delta = 0.051$, 94% HDI $[0.002, 0.113]$; hard: $\Delta = 0.119$, 94% HDI $[0.020, 0.247]$), and superordinates show a reliable easy–hard decrement ($\Delta = 0.075$, 94% HDI $[0.013, 0.150]$). Thus, Qwen-base replicates the same interaction pattern, with a particularly pronounced degradation of superordinate structure under embedding similarity in the hard condition.

LLaMA-3.1-8B-Instruct: superordinate advantage under logprob, but the same embedding superordinate penalty. For LLaMA-3.1-8B-Instruct, logprob accuracy is high across conditions but now shows a reliable relationship asymmetry. Superordinate logprob accuracy is near ceiling in both easy and hard conditions (0.997 and 0.993), while cohyponym logprob accuracy is lower (0.987 and 0.979). This yields a credible superordinate advantage at both difficulty levels (easy: cohyponym vs. superordinate $\Delta = -0.010$, 94% HDI $[-0.019, -0.003]$; hard: $\Delta = -0.014$, 94% HDI $[-0.029, -0.002]$), alongside a modest difficulty decrement (cohyponym easy–hard: $\Delta = 0.008$, 94% HDI $[0.003, 0.013]$; superordinate easy–hard: $\Delta = 0.004$, 94% HDI $[0.001, 0.007]$).

Embedding similarity, however, continues to exhibit the characteristic superordinate penalty. Cohyponym embedding accuracy remains near ceiling (0.998 easy; 0.995 hard), whereas superordinate embedding accuracy is lower and drops under hard distractors (0.968 easy; 0.907 hard). The cohyponym–superordinate gap is credibly positive at both difficulty levels and larger under hard difficulty (easy: $\Delta = 0.029$, 94% HDI $[0.004, 0.065]$; hard: $\Delta = 0.088$, 94% HDI $[0.019, 0.183]$), and superordinates show a reliable easy–hard decrement ($\Delta = 0.061$, 94% HDI $[0.012, 0.127]$). Thus, LLaMA-instruct preserves the core dissociation from Analysis 1: even when next-token scoring strongly favors superordinate judgments, static embedding geometry still supports cohyponym similarity more than superordinate structure, with the misalignment amplified under hard distractors.

Qwen-3-8B-Instruct: the same pattern, with especially large embedding-based superordinate degradation. Qwen-3-8B-Instruct shows a closely parallel pattern. Under logprob scoring, superordinate accuracy is extremely high (0.998 easy; 0.994 hard) whereas cohyponym accuracy is lower (0.990 easy; 0.978 hard), producing a credible superordinate advantage in both difficulty conditions (easy: $\Delta = -0.008$, 94% HDI $[-0.015, -0.003]$; hard: $\Delta = -0.016$, 94% HDI $[-0.032, -0.004]$). Difficulty effects are present but modest (cohyponym easy–hard: $\Delta = 0.012$, 94% HDI $[0.006, 0.019]$; superordinate easy–hard: $\Delta = 0.004$, 94% HDI $[0.001, 0.008]$).

D.2 General Model Performance Comparison

Model-to-model performance differences under fixed prompting. Analysis 3b asks what we can conclude about *general* performance differences between (i) GPT-2 vs. contemporary 8B base models, (ii) base vs. instruct variants within a family, and (iii) Qwen vs. LLaMA, while holding prompting fixed (task-specific prompt × task-specific metaprompt). We report planned contrasts computed from posterior draws for each cell of the relationship × difficulty × measure design (Figure 1; contrasts in Appendix Figure X), focusing on patterns that replicate across cells rather than isolated differences.

(1) Newer base models vs. GPT-2: large gains for logprob, but not for embedding similarity. Across all logprob cells, both contemporary base models (LLaMA-base and Qwen-base) outperform GPT-2 by clear margins, with the largest gaps appearing under hard distractors. For LLaMA-base, GPT-2 is lower by about 1–1.5 points in the easy conditions (superordinate/easy: $\Delta = -0.013$, 94% HDI $[-0.025, -0.003]$; cohyponym/easy: $\Delta = -0.015$, 94% HDI $[-0.023, -0.007]$) and by roughly 3.6–4.7 points in the hard condi-

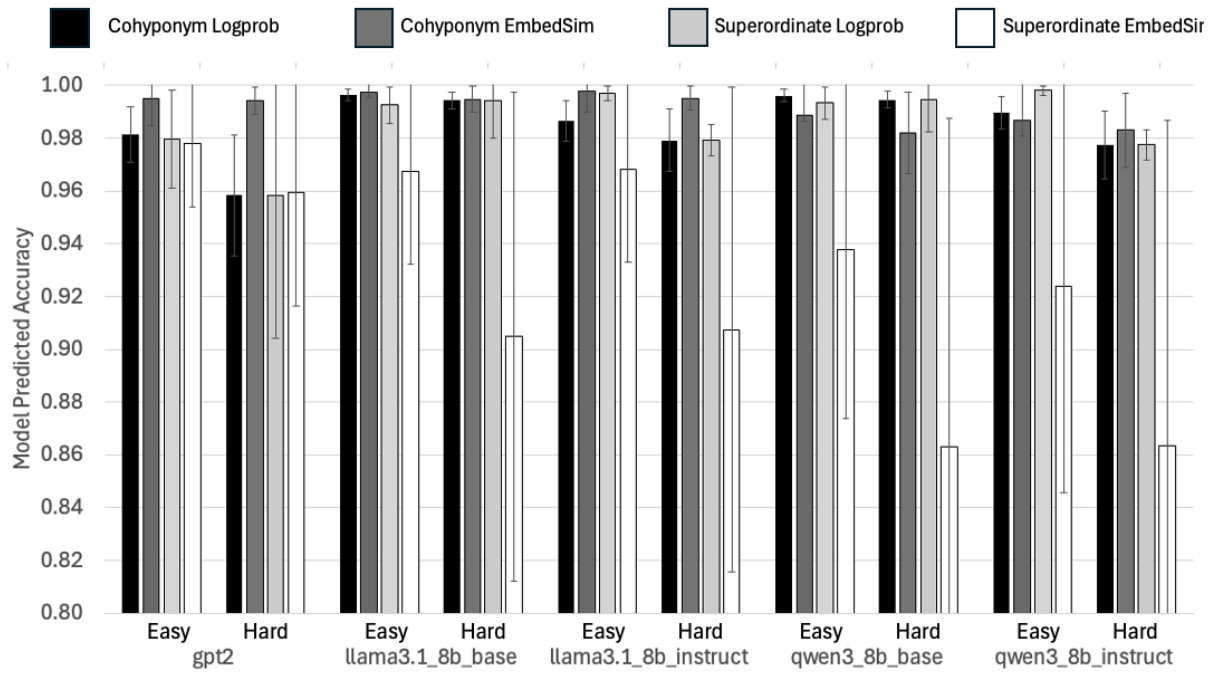

Figure 1: Bar plot showing results of model performance comparison under fixed prompting

tions (cohyponym/hard: $\Delta = -0.036$, 94% HDI $[-0.056, -0.018]$; superordinate/hard: $\Delta = -0.047$, 94% HDI $[-0.087, -0.013]$). The corresponding GPT-2 gaps to Qwen-base are nearly identical in direction and magnitude (easy: ≈1.4–1.5 points; hard: ≈3.6–5.0 points, all 94% HDIs excluding zero). Thus, under best-case prompting, the most robust model-family improvement from GPT-2 to modern 8B bases is expressed in next-token logprob discrimination, and it grows under more difficult distractors.

By contrast, embedding-similarity comparisons do *not* show a consistent "GPT-2 is worse" story. For cohyponyms, GPT-2 and LLaMA-base are essentially indistinguishable (easy: $\Delta = -0.002$, 94% HDI $[-0.005, 0.000]$; hard: $\Delta = -0.001$, 94% HDI spanning zero), whereas GPT-2 is *higher* than Qwen-base for cohyponyms (easy: $\Delta = +0.007$, 94% HDI $[0.000, 0.014]$; hard: $\Delta = +0.012$, 94% HDI $[0.002, 0.024]$). For superordinates, GPT-2 is *higher* than both bases in embedding similarity (vs. LLaMA-base hard: $\Delta = +0.055$, 94% HDI $[0.007, 0.115]$; vs. Qwen-base hard: $\Delta = +0.097$, 94% HDI $[0.024, 0.192]$; and similarly in easy). Overall, improvements from GPT-2 to contemporary bases are strong and systematic for logprob accuracy, but embedding-similarity accuracy does not track those improvements and can even reverse in sign in the superordinate cells.

(2) Base vs. instruct: instruction tuning increases superordinate logprob performance but reduces cohyponym logprob performance. Within each family, instruction tuning produces a consistent *tradeoff* in the logprob measure. For Qwen, the base model outperforms the instruct model on cohyponyms (easy: $\Delta = +0.0065$, 94% HDI $[0.0031, 0.0105]$; hard: $\Delta = +0.0171$, 94% HDI $[0.0084, 0.0271]$), but the instruct model outperforms base on superordinates (easy: base–instruct $\Delta = -0.0047$, 94% HDI $[-0.0092, -0.0013]$; hard: $\Delta = -0.0073$, 94% HDI $[-0.0141, -0.0019]$). LLaMA shows the same qualitative pattern: base exceeds instruct for cohyponyms (easy: $\Delta = +0.0096$, 94% HDI $[0.0045, 0.0151]$; hard: $\Delta = +0.0152$, 94% HDI $[0.0071, 0.0237]$), while instruct exceeds base for superordinates (easy: base–instruct $\Delta = -0.0045$, 94% HDI $[-0.0088, -0.0011]$; hard: $\Delta = -0.0094$, 94% HDI $[-0.0180, -0.0023]$). Thus, under next-token scoring and best-case prompting, instruction tuning does not behave as a uniform accuracy boost; instead, it systematically shifts performance toward superordinate (category-membership) discrimination and away from cohyponym (within-category similarity) discrimination.

For embedding similarity, base–instruct differences are generally small and often not credibly

different from zero within families, consistent with the fact that the embedding evaluation is not affected by prompt wording and appears only weakly sensitive to instruction tuning in this task. Notably, the large superordinate embedding penalties remain in both base and instruct variants, indicating that the tuning-driven logprob shift toward superordinates does not "repair" the embedding-based superordinate deficit.

(3) Qwen vs. LLaMA: near parity for logprob, but LLaMA tends to have stronger embedding performance. Within the base models, Qwen and LLaMA are extremely close on logprob in the easy conditions and differ only slightly under hard superordinates. Qwen-base shows a small advantage over LLaMA-base in the superordinate/hard logprob cell (Qwen–LLaMA: $\Delta = +0.0024$, 94% HDI $[0.0002, 0.0056]$), while the remaining logprob differences are small and not credibly different from zero. Within the instruct models, Qwen-instruct shows a very small logprob advantage over LLaMA-instruct in superordinate/easy and cohyponym/easy (superordinate/easy: $\Delta = +0.00095$, 94% HDI $[0.00018, 0.00190]$; cohyponym/easy: $\Delta = +0.00298$, 94% HDI $[0.00085, 0.00553]$), but the absolute magnitudes are tiny. Overall, *logprob-based semantic discrimination is broadly comparable between Qwen and LLaMA* under best-case prompting, with only small cell-specific differences.

Embedding similarity, however, shows a more consistent family separation: LLaMA tends to outperform Qwen across many embedding cells, especially in the superordinate conditions. For example, Qwen-base is lower than LLaMA-base in superordinate/easy embeddings (Qwen–LLaMA: $\Delta = -0.0299$, 94% HDI $[-0.0724, -0.0007]$) and is also lower in cohyponym embeddings (easy: $\Delta = -0.0089$, 94% HDI $[-0.0167, -0.0017]$; hard: $\Delta = -0.0128$, 94% HDI $[-0.0255, -0.0024]$). A similar pattern holds for instruct: Qwen-instruct is lower than LLaMA-instruct in embedding similarity for both superordinates and cohyponyms (e.g., superordinate/easy: $\Delta = -0.0443$, 94% HDI $[-0.0979, -0.0051]$; cohyponym/easy: $\Delta = -0.0110$, 94% HDI $[-0.0207, -0.0025]$). Thus, while the two families are near-parity under logprob scoring, *their static embedding geometries differ more systematically*, with LLaMA providing stronger embedding-based separability in this evaluation.

Interim conclusion for Goal 2. Analysis 3b supports three broad takeaways. First, relative to GPT-2, contemporary 8B base models show robust improvements in logprob-based semantic discrimination, particularly under hard distractors, but these improvements do not translate cleanly to embedding-based accuracy. Second, instruction tuning yields a reliable tradeoff under logprob: superordinate performance increases while cohyponym performance decreases, and this shift does not eliminate the embedding-based superordinate penalty. Third, Qwen and LLaMA are broadly comparable in logprob accuracy under best-case prompting, but LLaMA tends to show higher embedding-similarity accuracy (especially for superordinates), indicating a more favorable static embedding geometry for this task.