Graph-Based Alternatives to LLMs for Human Simulation

Joseph Suh, Suhong Moon, Serina Chang
University of California, Berkeley
{josephsuh,serinac}@berkeley.edu

Abstract

Large language models (LLMs) have become a popular approach for simulating human behaviors, yet it remains unclear if LLMs are necessary for all simulation tasks. We study a broad family of close-ended simulation tasks, with applications from survey prediction to test-taking, and show that a graph neural network can match or surpass strong LLM-based methods. We introduce **G**raph-bas**E**d **M**odels for Human **S**imulation (GEMS) which formulates close-ended simulation as link prediction on a heterogeneous graph of individuals and choices. Across three datasets and three evaluation settings, GEMS matches or outperforms the strongest LLM-based methods while using three orders of magnitude fewer parameters. These results suggest that graph-based modeling can complement LLMs as an efficient and transparent approach to simulating human behaviors. Code is available at https://github.com/schang-lab/gems.

1 Introduction

Human simulation has recently attracted significant attention, driving new research directions (Gao et al., 2024), workshops (SocialSim'25, 2025; PersonaLLM'25, 2026), panels (Hwang et al., 2025), and even startups (Expected Parrot, 2025; Artificial Societies, 2025). Throughout this excitement, large language models (LLMs) have remained by far the predominant approach, to the extent that references to this burgeoning field typically include LLM in their titles, such as "LLM social simulation" (Anthis et al., 2025) or "LLM-simulated data" (Hwang et al., 2025). While some simulation tasks are open-ended (Zhou et al., 2024; Bianchi et al., 2024), many of the most popular tasks are *close-ended*, predicting an individual's response from a set of options. Despite this close-ended structure, LLMs have also remained the predominant approach here, with many works proposing LLM-based methods for problems such as predicting survey responses (Santurkar et al., 2023; Hwang et al., 2023; Zhao et al., 2024; Feng et al., 2024; Moon et al., 2024; Suh et al., 2025; Cao et al., 2025; Kolluri et al., 2025; Krsteski et al., 2025), effects of social science experiments (Hewitt et al., 2024; Park et al., 2024; Manning et al., 2024), voting outcomes (Yu et al., 2024; Kreutner et al., 2025; von der Heyde et al., 2025; Li et al., 2025), and test-taking abilities (Wang et al., 2025; Binz et al., 2025).

While LLMs have shown strong results on these tasks, especially after fine-tuning (Suh et al., 2025; Cao et al., 2025; Binz et al., 2025), LLMs have downsides: they are expensive to run and train and their opaque pretraining processes lead to concerns of data leakage (Deng et al., 2024) and social biases (Cheng et al., 2023; Bisbee et al., 2024). This motivates a natural question: **for close-ended simulation tasks, can a smaller and more transparent model class be competitive with LLMs?** On one hand, focusing on these popular close-ended tasks may reduce LLMs' comparative advantage in open-ended text generation. On the other hand, high-quality simulation may still depend on unique LLM capabilities—language understanding, knowledge from pretraining, and effective adaptation via prompting or fine-tuning. Resolving this tension is important both for understanding what drives performance in close-ended simulation and for identifying modeling approaches that may be more efficient or transparent without sacrificing predictive performance.

The present work. We show that a simpler model—graph neural networks (GNNs)—can indeed match or outperform LLMs in close-ended simulation settings, with far less compute and greater transparency. We introduce **G**raph-bas**E**d **M**odels for Human **S**imulation (GEMS), which formulates close-ended simulation tasks as link prediction on a heterogeneous graph, with nodes as individuals and options and edges as observed choices (Figure 1). GEMS uses a GNN to learn

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 46479–46506

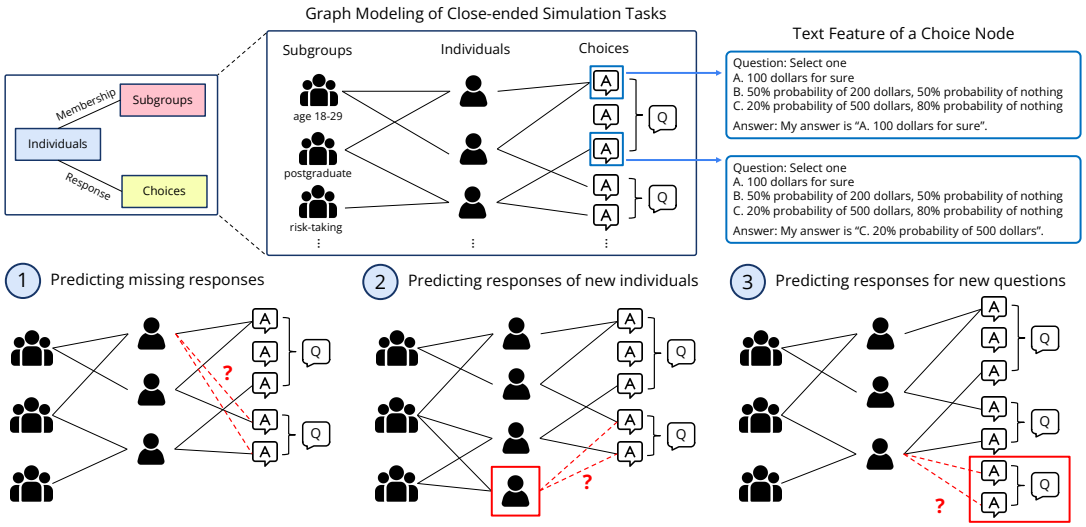


Figure 1: In GEMS, we construct a heterogeneous graph for close-ended simulation tasks (**Top**) where the goal is to predict the option chosen by an individual in response to a context. Under three widely studied settings (**Bottom**), we show that our approach achieves prediction accuracy consistently comparable to strong LLM-based methods.

representations of individuals, subgroups, and choices and to predict unseen choices. We consider three key settings for close-ended simulation tasks: predicting (1) missing responses (imputation), (2) responses of new individuals, and (3) responses for new questions. We compare GEMS against strong LLM-based methods—zero-shot, few-shot, chain-of-thought prompting, and supervised fine tuning—on three datasets: OPINIONQA public opinion polls (Santurkar et al., 2023), TWIN-2K psychological / economic / personality measures (Toubia et al., 2025), and tests of individuals' grammar skills (Jansen et al., 2021; Binz et al., 2025).

Across the datasets and settings, GEMS matches or outperforms strong LLM-based methods used in prior work. In the first two settings, GEMS achieves such performance using only the GNN and no language representations. In the third setting, where generalizing to unseen text is essential, leveraging a lightweight projection layer mapping frozen LLM representations to GNN embeddings (Sheng et al., 2025) yields comparable performance. While matching LLM accuracy, GEMS uses $\sim 10^3$ fewer parameters and up to 10^2 times less compute, lowering barriers for researchers with limited compute and enabling scaling to larger datasets. Furthermore, unlike LLMs' opaque pretraining, GEMS can be trained from scratch on transparent, domain-specific data, mitigating data leakage and bias concerns. Overall, our work offers graph-based modeling as a compelling alternative to LLMs for human simulation, enabling broader participation in this interdisciplinary field and more trustworthy predictions.

2 Related Work

As interest in human simulation has risen sharply in the past few years, LLMs have remained by far the predominant approach (Gao et al., 2024; Anthis et al., 2025; Hwang et al., 2025). As described previously, a large portion of this literature focuses on close-ended simulation tasks, aiming to predict an individual's choice among a set of options, typically cast as next-token prediction of the LLM. Prior work has explored prompting strategies (Dominguez-Olmedo et al., 2023), including few-shot prompting (Hwang et al., 2023) and prompt engineering (Kim and Yang, 2025), as well as conditioning on open-ended narratives (Park et al., 2024; Moon et al., 2024; Rahimzadeh et al., 2025). Recently, fine-tuning has emerged as a promising alternative, either on community-specific text corpora (Chu et al., 2023; He et al., 2024; Li et al., 2024; Feng et al., 2024) or directly on human choice data (Cao et al., 2025; Suh et al., 2025; Binz et al., 2025; Xie et al., 2025; Kolluri et al., 2025), augmented with prediction-powered inference (Angelopoulos et al., 2023; Krsteski et al., 2025).

Yet the task remains selecting from a small, fixed set of options (or tokens), where LLM-based methods might not be the sole approach. We build on this observation and, in GEMS, emphasize the relational structure underlying human choices. This approach exploits similar relational dependencies as graph-based recommender systems (Ying et al., 2018; Fan et al., 2019; He et al., 2020), but have been absent in the LLM-centric simulation literature. Our work is the first to clarify how LLM

methods perform in comparison to graph-based models in human choice simulation and to provide a direct comparison. In the Appendix (Section A), we provide an extended discussion of graph-based recommenders, along with classical models (e.g., from economics) for modeling discrete choice.

3 Problem Definition

We focus on close-ended simulation tasks, where the goal is to predict an individual's selected response among a set of options. Given an individual u and a question q, with answer options $\mathcal{A}(q)$, the goal is to predict u's response $y_{uq} \in \mathcal{A}(q)$. Each individual has *individual features*, such as demographic variables. We use individual features to define *subgroups*, which are groups of individuals sharing one or more features. We also have *question features* and *option features*. Since we focus on simulation tasks where LLM-based methods have been used, these features are text, i.e., the text of the question and of each option, but our framework is not restricted to text-only features. We define a *choice* as a pair (q, a) of question q and answer option $a \in \mathcal{A}(q)$; its *choice feature* is the concatenation of the question text and option text. We observe a set of prior responses $\mathcal{Y}$, which consists of responses from *seen* individuals (i.e., those with at least one response in $\mathcal{Y}$) and *seen* questions (i.e., those with at least one response in $\mathcal{Y}$). However, we do not observe responses between all pairs of seen individuals and questions.

We consider three settings of simulating human choices widely studied in previous work (Figure 1).

(1) Missing response (Imputation). Given a *seen* individual u with individual features and a *seen* question q with question and option features, predict the missing response $y_{uq} \notin \mathcal{Y}$. Prior work studies few-shot prompting and fine-tuning for this setting (Hwang et al., 2023; Zhao et al., 2024; Kim and Yang, 2025; Kolluri et al., 2025).

(2) New individuals. Given a *new* individual u, where we observe their individual features but not any prior responses, predict u's responses to seen questions. This setting has been investigated in several simulation works (Santurkar et al., 2023; Moon et al., 2024; Kang et al., 2025; Li et al., 2025) and is also of interest to pluralistic alignment (Feng et al., 2024; Yao et al., 2025).

(3) New questions. Given a *new* question q, where we observe its question and option features but not any prior responses, predict the responses of seen individuals to q. This setting is useful for simulating newly designed items in survey research (Rothschild et al., 2024; Suh et al., 2025; Cao et al., 2025) or testing generalization to new simulation settings (Binz et al., 2025; Xie et al., 2025).

4 GEMS Graph-based Modeling

4.1 Graph Representation of the Task

We represent the task as a heterogeneous graph $\mathcal{G}$ with three types of nodes: subgroups $\mathcal{S}$, individuals $\mathcal{U}$, and choices $\mathcal{C}$. Choice nodes are structured as a disjoint union $\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2 \cup \cdots \cup \mathcal{C}_n$, where $\mathcal{C}_q$ is the set of choice nodes for question q and n is the total number of questions. We include two bidirectional relations: membership and response. Membership edges $E_{\mathcal{US}}$ with an adjacency matrix $\mathbf{A}_{\mathcal{US}} \in \{0,1\}^{|\mathcal{U}|\times|\mathcal{S}|}$ connect each individual to relevant subgroups. Response edges $E_{\mathcal{UC}}$ with an adjacency matrix $\mathbf{A}_{\mathcal{UC}} \in \{0,1\}^{|\mathcal{U}|\times|\mathcal{C}|}$ record which choice an individual made as a response to a question. Because each question requires selecting one choice, the row-wise sum of $\mathbf{A}_{\mathcal{UC}}$ is at most n.

4.2 GNN Architecture

Given the graph representation, we define GEMS as a link prediction model trained end-to-end. As illustrated in Figure 2, an encoder performs relation-aware message passing to produce node embeddings for subgroups, individuals, and choices, and the decoder performs link prediction from output node embeddings. To generalize to *new questions* (setting 3) whose choice nodes have no edges at test time, we additionally train an LLM-to-GNN projection that maps choice nodes' text features (frozen LLM hidden states) to representations in the GNN output embedding space.

Input node features. Individual nodes $\mathcal{U}$ are non-identifiable, thereby assigned a uniform feature $Z_{\mathcal{U}} = \mathbf{1}_{|\mathcal{U}|}$. For subgroup nodes $\mathcal{S}$, we learn input node features via a learnable table $Z_{\mathcal{S}} \in \mathbb{R}^{|\mathcal{S}|\times d_{\mathcal{S}}}$, with a feature dimension $d_{\mathcal{S}}$. For choice nodes $\mathcal{C}$, we also maintain a learnable table $Z_{\mathcal{C}} \in \mathbb{R}^{|\mathcal{C}|\times d_{\mathcal{C}}}$ with a feature dimension $d_{\mathcal{C}}$; no textual information about choices are provided as input to the graph.

Graph encoder. We adopt heterogeneous graph extensions of GNNs, e.g., RGCN, GAT, and GraphSAGE (Schlichtkrull et al., 2018; Veličković et al., 2018; Hamilton et al., 2017). Let $z_w^{(0)}$ be the

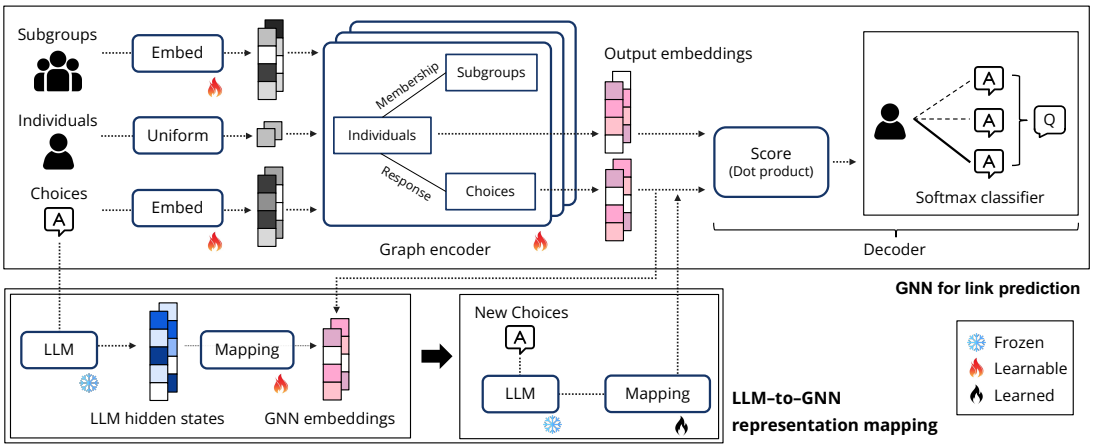


Figure 2: Overall architecture of GEMS. The graph encoder learns representations of individual and choice nodes from the relational structure of observed responses, then predicts new responses with a softmax classifier over question options (**Top**). In setting 3, we train a simple LLM-to-GNN projection that maps a language representation of the choice node's text to its GNN embedding, so that we can acquire meaningful representations of new questions (**Bottom**).

input feature for node w from $Z_{\mathcal{U}}$, $Z_{\mathcal{S}}$, or $Z_{\mathcal{C}}$. An L-layer graph encoder computes

$$z_w^{(\ell+1)} = \sigma\Bigg(\sum_{r\in\mathcal{R}}\Big[\mathrm{AGG}_r \underset{v\in\mathcal{N}_r(w)}{\phi_r^{(\ell)}}\big(z_w^{(\ell)}, z_v^{(\ell)}\big)\Big] + \phi_{\text{self}}^{(\ell)}\big(z_w^{(\ell)}\big)\Bigg), \ell = 0,\ldots,L-1. \quad (1)$$

where $\mathcal{R} = \{\mathcal{U}\to\mathcal{S}, \mathcal{S}\to\mathcal{U}, \mathcal{U}\to\mathcal{C}, \mathcal{C}\to\mathcal{U}\}$ are types of two bidirectional relations (membership and response), $\phi_r^{(\ell)}$ is a relation-specific message passing, $\phi_{\text{self}}^{(\ell)}$ is a self-loop, AGG_r is a per-relation aggregation over neighbors $\mathcal{N}_r(w)$, and σ is a non-linear activation function. We present the details of each function for different GNN architectures in Section D. After the final layer L, we apply a node-type-specific linear projection on $z_w^{(L)}$ to obtain the output embedding $z_w^O \in \mathbb{R}^{d_{\text{GNN}}}$, where d_{GNN} is the dimension of output node embeddings.

Link prediction decoder. The decoder consists of a dot-product and a softmax classifier. For an individual $u \in \mathcal{U}$ and a question q with a set of choice nodes $\mathcal{C}_q \subseteq \mathcal{C}$, the score between the individual and each choice $c \in \mathcal{C}_q$ is obtained as $\mathrm{Dot}(u,c) = (z_u^O)^\top z_c^O$. These scores are then converted to a distribution over choices, with a learnable temperature τ:

$$p(c|u,q) = \frac{\exp\big(\mathrm{Dot}(u,c)/\tau\big)}{\sum_{c'\in\mathcal{C}_q}\exp\big(\mathrm{Dot}(u,c')/\tau\big)}. \quad (2)$$

LLM-to-GNN representation map. In setting 3 (new questions), choice nodes for the new question are isolated in the graph since we do not have any responses for that question yet, and have no learned features in the learnable table $Z_{\mathcal{C}}$. Therefore, the graph encoder cannot produce the output embedding for new choice nodes. To enable prediction, we generate a substitute embedding directly from its text features by learning an LLM-to-GNN representation mapping on seen questions. For a choice c, the mapping takes a language representation of the choice's text features (a frozen LLM's hidden state $h_{\text{LLM}}(c) \in \mathbb{R}^{d_{\text{LLM}}}$) then outputs $z'_c = \mathbf{W}_{\text{proj}} h_{\text{LLM}}(c) \in \mathbb{R}^{d_{\text{GNN}}}$, where d_{LLM} and d_{GNN} are dimensions of LLM hidden states and GNN output embeddings, respectively.

The projection is trained on seen choice nodes by matching z'_c to the output node embedding z_c^O, inspired by previous work (Sheng et al., 2025; Zhang et al., 2019). At inference time, for a new question q, we compute z'_c for each $c \in \mathcal{C}_q$ and plug these into the decoder in place of z_c^O. We note that this mapping is only needed in setting 3; settings 1–2 use the output embeddings z_c^O directly. Furthermore, since we use frozen LLM hidden states, we only add $d_{\text{LLM}} \times d_{\text{GNN}}$ trainable parameters (far less than LLM fine-tuning) and do not require querying an LLM each time to predict new edges, unlike prompt-based LLM approaches.

4.3 Training objective

Link prediction. Following self-supervised link prediction (Kipf and Welling, 2016; Berg et al., 2017), we train end-to-end by exposing a subset of train edges to the graph encoder and supervising the model to reconstruct the rest. At each train step we randomly mask response edges from $E_{\mathcal{UC}}$, with a masking strategy defined in Section 5 per setting.

For example, say we masked a response edge (u,c) for an individual u and a choice c where c belongs to a question $q(c)$. The decoder generates a probability $p(c|u,q(c))$ by Equation (2). We aim to minimize the cross-entropy loss

$$-\sum_{(u,c)\in\text{masked}} \log p(c|u,q(c)) \quad (3)$$

It requires no explicit negative sampling: the masked response edge (u,c) is the positive edge, while (u,c') for all $c' \in \mathcal{C}_{q(c)} \setminus \{c\}$ act as implicit negatives through the normalization with softmax.

LLM-to-GNN map. For setting 3 (new question), we learn a linear mapping $\mathbf{W}_{\text{proj}}$ by minimizing

$$\sum_{c\in\mathcal{C}_{\text{train}}} \left\|\mathbf{W}_{\text{proj}} h_{\text{LLM}}(c) - z_c^O\right\|_2^2 + \alpha\|\mathbf{W}_{\text{proj}}\|_2^2 \quad (4)$$

where $\mathcal{C}_{\text{train}}$ is the set of choice nodes available during training, h_{LLM} is a frozen LLM's hidden state for a text feature of a choice node c, and z_c^O is the output embedding of an L-layer graph encoder. α is a hyperparameter of a ridge regression selected by the prediction accuracy on the validation set.

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate on three simulation datasets: (1) OPINIONQA public opinion polls (Santurkar et al., 2023), comprising responses from 76K individuals to 500 questions spanning various social topics (e.g., political attitudes, media consumption); (2) TWIN-2K (Toubia et al., 2025), a 150-item battery including economic preferences, cognitive biases, and personality traits, administered to 2K individuals; and (3) DUNNING-KRUGER effect replication (Jansen et al., 2021; Binz et al., 2025), 20 grammar and logical reasoning questions with pre- and post-question confidence ratings, administered on 3K individuals. For (1) and (2), nine demographic attributes (e.g., age, gender) are used to define individual features and subgroup nodes; for (3), responses to four pre-question confidence ratings are used as individual features. Examples of questions and choices are in Section B. Dataset splits are described per setting below; graph statistics appear in Section C.

Evaluation metric. We use average prediction accuracy, comparing the individual's true choice to the highest-probability model prediction. Specifically, for a test response edge (u,c) with a question $q(c)$ that c belongs to, the model's prediction is correct if $c = \text{argmax}_{c'\in\mathcal{C}_{q(c)}} p(c'|u,q(c))$ (Equation (2)) and incorrect otherwise.

Compared methods. We compare GEMS against five LLM-based baselines and lower/upper performance bounds. Prompt examples for each of the baselines are provided in Section G.

1. Zero-shot prompting: Prompt with individual features, following Santurkar et al. (2023).

2. Few-shot prompting: Prompt with individual features and the individual's prior responses, following Hwang et al. (2023); Kim and Yang (2025).

3. Agentic CoT prompting: A chain-of-thought (CoT) framework consisting of a reflection agent and a prediction agent (Park et al., 2024).

4. Supervised fine-tuning (SFT): Fine-tune an LLM to predict the answer token given individual features (Cao et al., 2025; Suh et al., 2025; Yao et al., 2025; Kolluri et al., 2025).

5. Few-shot fine-tuning (Few-shot FT): Fine-tune an LLM with individual features and the individual's prior responses (Zhao et al., 2024).

6. Random (lower bound): Uniformly sample a choice from the question's available options.

7. Human retest (upper bound): When available from dataset authors, report test-retest accuracy. It is the probability that the same individual repeats the same choice when re-asked the same question after a fixed time interval (e.g., two weeks).

For main experiments we adopt three instruction-tuned language models, LLaMA-2-7B, Mistral-7B-v0.1, Qwen3-8B (Touvron et al., 2023; Jiang et al., 2023; Yang et al., 2025). We also present additional inference results in Section E including state-of-the-art proprietary models. We note that results with additional models do not alter the overall trends or our main conclusions.

5.2 Setting 1: Missing Responses (Imputation)

Setup. We follow the split scheme of (Zhao et al., 2024): each dataset is first split at an individual level into 35/5/60% train/validation/test individuals. For each individual held out for validation/test, 40% of their responses are also available during training, while 60% are held out for evaluation. LLM fine-tuning prompts and train graphs are built upon all responses from

Table 1: Accuracy of imputing missing responses. Numbers indicate mean test accuracy with standard deviation from 3 train/val/test random splits with different seeds. Numbers in parentheses indicate the number of in-context examples (e.g., 3) or the GNN architecture (e.g., RGCN). For each dataset, bold marks the best accuracy per GEMS and LLM-based methods; underline for the runner-up. N.A. and C.L. stand for 'not available' and 'context limit', respectively.

Methods	OpinionQA			Twin-2K			Dunning-Kruger		
	LLaMA2-7B	**Mistral-7B**	**Qwen3-8B**	**LLaMA2-7B**	**Mistral-7B**	**Qwen3-8B**	**LLaMA2-7B**	**Mistral-7B**	**Qwen3-8B**
Random	27.87			35.05			20.00		
Human retest	N.A.			81.72			N.A.		
Zero-shot	29.18±0.15	34.63±0.19	39.38±0.20	41.49±0.31	42.47±0.27	52.06±0.38	22.76±0.59	39.00±0.73	41.82±0.41
Few-shot (3)	38.54±0.21	42.52±0.06	42.21±0.08	41.44±0.88	48.25±0.73	54.10±0.51	26.77±0.23	43.54±0.72	46.81±0.48
Few-shot (8)	37.91±0.65	45.78±0.56	43.66±0.59	43.40±0.99	51.26±0.84	56.08±1.01	26.34±0.51	43.59±0.60	47.60±0.67
Agentic CoT (3)	32.19±0.25	41.37±0.47	47.63±0.17	33.13±1.57	50.14±0.93	57.89±1.80	31.01±1.51	34.41±2.84	51.18±0.79
Agentic CoT (8)	28.80±0.15	38.43±0.31	47.97±0.36	C.L.	48.76±0.53	60.20±1.28	31.68±0.88	35.45±1.23	54.71±1.27
SFT	49.41±0.12	50.56±0.14	48.84±0.14	61.23±0.13	61.85±0.13	61.49±0.15	56.47±0.10	56.45±0.14	56.47±0.03
Few-shot FT (3)	55.59±0.11	56.31±0.10	55.09±0.14	63.51±0.15	63.91±0.16	62.61±0.19	56.81±0.15	56.88±0.05	56.71±0.12
Few-shot FT (8)	55.98±0.12	**56.76±0.13**	55.61±0.13	65.86±0.17	**66.36±0.13**	65.27±0.16	57.18±0.02	**57.21±0.41**	56.94±0.11
GEMS (RGCN)	56.89±0.12			66.36±0.13			57.68±0.13		
GEMS (GAT)	56.40±0.10			66.01±0.14			56.95±0.09		
GEMS (SAGE)	**57.00±0.12**			**66.62±0.12**			**57.89±0.10**		

Table 2: Accuracy of predicting responses of new, unseen individuals. Numbers indicate mean test accuracy with standard deviation from 3 train/val/test random splits with different seeds. For LLM, few-shot methods are not applicable since prior responses for new individuals are not available.

Methods	OpinionQA			Twin-2K			Dunning-Kruger		
	LLaMA2-7B	**Mistral-7B**	**Qwen3-8B**	**LLaMA2-7B**	**Mistral-7B**	**Qwen3-8B**	**LLaMA2-7B**	**Mistral-7B**	**Qwen3-8B**
Random	27.87			35.05			20.00		
Zero-shot	29.15±0.15	34.40±0.13	38.97±0.16	41.57±0.39	43.03±0.50	51.79±0.27	22.44±0.39	38.83±0.70	42.06±0.41
Agentic CoT	18.44±0.47	33.84±0.31	39.53±0.22	21.91±0.82	45.30±0.34	53.45±0.43	31.52±1.47	34.04±0.66	49.96±1.19
SFT	49.35±0.15	**50.49±0.17**	48.87±0.16	61.29±0.22	**61.85±0.19**	61.38±0.22	56.54±0.33	**56.66±0.17**	56.50±0.34
GEMS (RGCN)	50.50±0.12			62.39±0.14			56.76±0.21		
GEMS (GAT)	50.36±0.14			62.22±0.14			56.70±0.10		
GEMS (SAGE)	**50.73±0.11**			**62.50±0.19**			**57.07±0.32**		

the train individuals and 40% responses from the validation/test individuals. At each training step of GEMS, 50% of response edges in the train graph are randomly masked and used as supervision edges, while all membership edges and unmasked train response edges serve as message passing edges. At validation/test, the entire train graph is used for message passing to predict held-out edges. For LLM few-shot prompts we use 3 or 8 in-context examples, selected from training data by the highest cosine similarity of text embeddings, following Liu et al. (2021) and Hwang et al. (2023). Please refer to Section D for additional details.

Results. Table 1 reports results. GEMS outperforms all LLM prompting baselines and SFT, and matches the strongest LLM baseline, 8-shot fine-tuning. Performance of LLM-based methods generally improves with more sophisticated prompt design and compute, from zero-shot prompting to few-shot fine-tuning; however, GEMS attains comparable accuracy without using any textual features, relying solely on a learnable feature table of choices and subgroups. We attribute this to the relational structure that alone provides sufficient signal about the choice, even in the absence of textual information. For example, GNN infers that two frequently co-selected choices share latent attributes and therefore someone choosing one will likely choose the other, even without text of choices. Taken together, these results highlight the value of relational structure for accurate prediction.

5.3 Setting 2: New Individuals

Setup. The split is also done at an individual level: 35% train, 5% validation, and 60% test individuals. In contrast to setting 1 where we hold out 60% responses from each validation/test individual, here we hold out all responses to have a new individual at test time. We also modify GEMS training to teach the model how to make predictions for new individuals. At each training step, we randomly

Table 3: Accuracy of predicting human responses to new, unseen questions. Numbers indicate mean test accuracy with standard deviation from 3 train/val/test random splits with different seeds. For GEMS, within a row, each column indicates the performance with hidden states from different LLMs. Experimental details are in Section D.

Methods	OPINIONQA			TWIN-2K			DUNNING-KRUGER		
	LLaMA2-7B	**Mistral-7B**	**Qwen3-8B**	**LLaMA2-7B**	**Mistral-7B**	**Qwen3-8B**	**LLaMA2-7B**	**Mistral-7B**	**Qwen3-8B**
Random	27.87			35.05			20.00		
Zero-shot	29.15±0.57	35.60±2.91	38.84±1.08	40.03±2.45	41.30±3.69	50.94±2.76	19.77±5.64	43.87±4.74	44.12±4.40
Few-shot (3)	37.93±2.24	42.49±3.16	42.74±2.87	42.09±3.38	47.88±1.93	54.02±4.19	23.83±5.85	49.31±4.48	52.55±3.71
Few-shot (8)	37.98±1.62	42.81±3.39	44.05±2.65	41.15±2.77	47.93±2.30	55.09±2.50	23.81±6.30	44.30±4.12	52.93±3.97
Agentic CoT (3)	31.46±2.92	40.20±1.60	45.90±3.57	32.16±3.66	49.67±3.61	56.18±2.74	23.56±18.8	31.18±8.58	53.54±4.41
Agentic CoT (8)	27.15±1.42	37.45±4.94	46.18±3.70	C.L.	48.24±5.43	58.08±2.83	24.45±17.8	30.98±9.37	53.72±4.61
SFT	44.12±2.30	47.86±0.95	43.95±0.87	55.85±1.21	56.21±0.96	56.24±1.42	33.65±13.1	48.12±7.46	43.08±5.31
Few-shot FT (3)	49.83±1.53	51.77±1.09	49.59±0.84	58.07±1.86	59.86±1.52	59.99±1.33	41.18±16.3	49.28±9.40	**54.13±4.98**
Few-shot FT (8)	50.11±1.97	**51.83±1.47**	50.00±1.00	59.87±1.35	**60.84±1.40**	60.48±1.79	41.81±9.96	51.79±8.00	53.47±5.93
GEMS (RGCN)	48.94±1.71	**50.13±1.85**	49.07±1.48	56.24±3.65	**60.37±2.47**	59.59±4.42	47.20±1.66	47.31±9.13	**52.52±3.91**
GEMS (GAT)	46.87±1.78	49.25±2.46	48.52±2.13	52.00±1.52	56.57±1.95	57.38±2.44	45.02±3.09	44.57±4.85	52.01±3.16
GEMS (SAGE)	47.29±1.89	49.84±1.98	49.09±1.80	54.06±4.47	58.56±2.43	60.03±3.88	46.17±2.59	47.43±8.60	52.02±3.58

select 50% of training individuals, mask all of their response edges to use as supervision edges, and use all membership edges and unselected training individuals' response edges for message passing.

Results. Table 2 reports results for setting 2. GEMS outperforms the LLM prompting baselines and matches the strongest LLM baseline, SFT. Trends mirror that of setting 1: (i) zero-shot prompting and CoT exceed a random baseline and benefit from stronger LLMs but fall behind SFT, and (ii) fine-tuning narrows performance gaps across LLM families. In GEMS, a new individual connects only to subgroup nodes via membership edges; with input features $\mathbf{1}_{|\mathcal{U}|}$ (Section 4.2), their output node embeddings are obtained entirely by aggregating messages from subgroup neighbors (Equation (1)). By masking out all response edges for 50% individuals during training, the learnable subgroup features $Z_{\mathcal{S}}$ are encouraged to encode representations that generalize to new individuals, precisely what is needed for the current setting. LLM-based methods can acquire similar knowledge by iteratively seeing pairs of individual features and responses, but at a substantially higher computational cost.

5.4 Setting 3: New Questions

Setup. We split at the question level into 70/10/20% train/validation/test, following Suh et al. (2025). Validation/test questions are entirely unseen during training. Even at test time their choice nodes are isolated in the graph, and do not have learned input features in the table $Z_{\mathcal{C}}$. Responses to train questions from all individuals are used to fine-tune LLMs or to construct the train graph. At validation/test, responses to train questions are used as few-shot examples or as message-passing edges.

GEMS is trained in two stages. In the first stage, we train the GNN using the link prediction objective (Equation (3)) to learn representations of individual and choice nodes in the train graph. To this end, we initially hold out a small fraction (5%) of response edges from the train graph, which we call "transductive validation edges". At each GNN training step, remaining 95% response edges in the graph are partitioned into 50% supervision edges and 50% message-passing edges as done in setting 1. At the GNN checkpoint with the best accuracy on the transductive validation edges, we extract the output node embeddings z_c^O of choice nodes $c \in \mathcal{C}_{\text{train}}$ (Equation (1)). In the second stage, we train a linear projection to map LLM hidden states of $\mathcal{C}_{\text{train}}$ text features to the GNN output node embedding space (Equation (4)). At test time, we make predictions for new questions using the projected embeddings of their choices, as described in Section 4.2.

Results. Table 3 reports results for setting 3. GEMS, with the LLM-to-GNN representation mapping, remains competitive with all LLM-based methods, while requiring far fewer trainable parameters. This performance is achieved with LLM hidden states – GNN output embedding pairs from 500 (OPINIONQA), 150 (TWIN-2K), or 20 (DUNNING-KRUGER) questions. We also observe that the choice of LLM affects GEMS: accuracies achieved with GEMS correlate with those of the corresponding LLM-based baselines, indicating that gains from stronger LLMs translate through the mapping (Sheng et al., 2025).

5.5 Error Analysis

To better understand the relationship between GEMS and LLM predictions, we compute contingency tables between the best-performing LLM method and the best-performing GEMS variant on the OPINIONQA dataset across all three settings.

Table 4: Contingency tables (%) between the best LLM method and GEMS on OPINIONQA. ✓(✗) indicates that the model makes a correct (incorrect) prediction.

	Setting 1		Setting 2		Setting 3	
	LLM✓	LLM✗	LLM✓	LLM✗	LLM✓	LLM✗
GNN✓	53.4	3.6	47.9	2.8	40.9	9.3
GNN✗	3.4	39.6	2.6	46.7	11.0	38.8

Across all three settings, shared failures (bottom right) are substantially larger than complementary subsets (off-diagonal entries), indicating that two methods tend to make similar predictions. In Settings 1 and 2, the complementary subsets where only one model succeeds are small (3–4% each), suggesting limited headroom for simple cross-model ensembling. In Setting 3, the off-diagonal mass is larger (∼20% combined), implying greater complementarity between GEMS and LLM-based methods when generalizing to new questions.

5.6 Comparison with Classical Baselines

The previous sections show that GEMS matches or outperforms LLM-based methods. To isolate the contribution of graph structure beyond classical tabular or factorization methods, we compare GEMS against two additional baselines.

XGBoost (Settings 1 and 2). We train a per-question XGBoost classifier whose input is a one-hot encoding of individual features (e.g., demographic traits) and whose output is a distribution over the available choices. XGBoost cannot generalize to entirely new questions (Setting 3), as it lacks a mechanism to transfer across questions.

Matrix factorization (Setting 1). We learn embeddings for each individual and each choice by optimizing a cross-entropy loss over softmax-normalized dot products. This method applies only to Setting 1, as it cannot construct embeddings for new individuals or new questions at test time.

Results are reported in Table 5. In Setting 1, where the model must capture relational structure among discrete choices across multiple questions, matrix factorization underperforms both GEMS and LLM few-shot fine-tuning, with XGBoost performing even worse. In Setting 2, which relies more heavily on individual features, the simpler baselines perform more competitively, though GEMS still achieves a consistent edge. Importantly, neither XGBoost nor matrix factorization can handle Setting 3, while GEMS can. These results suggest that GEMS' primary advantage over classical methods lies in its ability to learn richer relational structure through message passing.

Table 5: Comparison with classical methods on OPINIONQA (OQ), TWIN-2K (TW), and DUNNING-KRUGER (DK). We note that neither XGBoost nor matrix factorization can handle Setting 3.

Method	OQ	TW	DK
Setting 1: Missing Responses			
XGBoost	50.20	61.94	56.44
Matrix Factorization	52.12	62.77	56.53
LLM Best (Few-shot FT, 8)	56.76	66.36	57.21
GEMS Best	**57.00**	**66.62**	**57.89**
Setting 2: New Individuals			
XGBoost	50.16	61.61	56.24
LLM Best (SFT)	50.49	61.85	56.66
GEMS Best	**50.73**	**62.50**	**57.07**

6 Advantages of GEMS

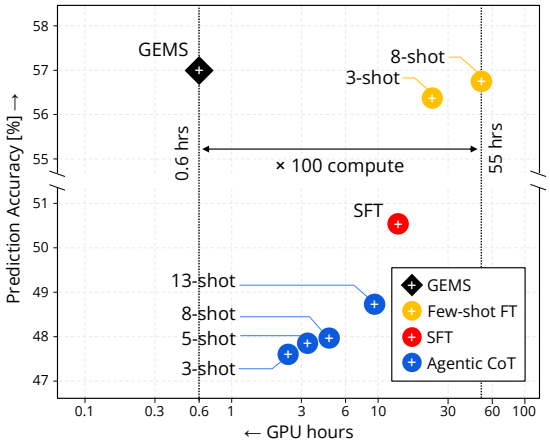


Figure 3: Accuracy *vs.* GPU-hours (A100-80GB-SXM4) on the OPINIONQA dataset, setting 1. Zero-/few-shot prompting accuracies fall below the plotted y-range. For LLM-based methods, we report the best result across three LLMs (LLaMA-2-7B, Mistral-7B-v0.1, and Qwen3-8B). For GEMS, we report the best result across three models (RGCN, GAT, and SAGE). See Section D for details and Section E for the extended figure.

While maintaining comparable accuracy to LLMs, GEMS brings numerous practical advantages.

Efficiency and scalability. As shown in Figure 3, GEMS matches the strongest LLM-based methods

while using $\sim 10^2$ less compute and $\sim 10^3$ fewer parameters (see Section D.7), remaining tractable as both the base LLM and dataset scale. Extrapolating from Figure 3, few-shot fine-tuning a 70B model on OPINIONQA would require ~ 500 GPU-hours, whereas GEMS completed in less than an hour. Likewise, scaling to datasets $10\times$ larger than OPINIONQA (e.g., SubPoP (Suh et al., 2025)) would push an LLM fine-tuning toward $\sim 10^3$ GPU-hours, while GEMS would train in a few hours.

This efficiency advantage also enables prediction ensembling (Lakshminarayanan et al., 2017), which averages predicted probabilities across multiple models trained with different initializations. Leveraging GEMS' compute efficiency ($\sim 10^2$ compared to LLM fine-tuning) we ensemble predictions from 11 differently initialized GEMS models (still ~ 10x cheaper than a single LLM fine-tuning) and observe consistent accuracy gains:

Table 6: Ensembling results on OPINIONQA. GEMS Ensembled averages predicted probabilities from 11 GEMS models trained with different initializations.

Method	Setting 1	Setting 2	Setting 3
LLM Best	56.76	50.49	51.83
GEMS Best (single)	57.00	50.73	50.13
GEMS Ensembled (11)	57.31	50.86	51.65
Gain from ensembling	+0.31	+0.13	+1.52

This demonstrates an advantage of GEMS: one can affordably train many models and ensemble predictions for improved accuracy, a strategy prohibitively expensive with LLM fine-tuning.

Transparency and trustworthiness. LLMs are often trained on undisclosed data, which creates contamination concerns where evaluation data (e.g., past behavioral studies) may have appeared during training (Deng et al., 2024). Furthermore, LLMs have been shown to display social biases in simulation, such as leaning towards certain groups' opinions (Santurkar et al., 2023), stereotyping (Cheng et al., 2023), or underestimating variance (Bisbee et al., 2024). Finally, pretrained LLMs are sensitive to prompt format (Lu et al., 2022; Sclar et al., 2024), with many formatting decisions involved in simulation tasks. All of these issues challenge the trustworthiness of LLM-based human simulations.

In contrast, GEMS is trained from scratch on task-specific data, removing issues of contamination or learning social biases from pretraining data. Furthermore, there is no issue of ordering in-context examples or individual features, since GNN aggregation is equivariant to the order of neighbors (Hamilton et al., 2017). Prompt formatting is only relevant to GEMS when the LLM-to-GNN mapping is used; even then, we find that it exhibits lower variance under prompt perturbations due to the training of the projection matrix.

Insights. Finally, directly inspecting GEMS' embeddings reveals insights about human behavior and preferences (Figure 5). For example, when we train GEMS on OPINIONQA, we find that certain dimensions of opinions naturally emerge in the embedding space, with the first and second principal components corresponding to political ideology and class, respectively. Second, even though we see clear subgroup-level patterns, inspecting the embeddings of individuals reveals substantial heterogeneity among individuals in the same subgroup. These results emphasize the diversity of individuals beyond their demographics, in contrast with LLM methods that exhibit demographic stereotyping and underestimate variance within subgroups (Cheng et al., 2023; Bisbee et al., 2024). Third, we find that GEMS encodes nuanced meanings of questions that are missed by LLMs. In particular, two choices that reflect similar ideology but have different wordings—for example, saying that "reducing illegal immigration" is "a top priority" and "addressing climate change" is "not too important"—have similar GEMS embeddings, while the LLM hidden states tend to be overly focused on surface word similarity (e.g., all "a top priority" are clustered regardless of the topic). Please refer to Section E for details.

7 Conclusion

We present GEMS, a graph-based approach to model a large class of close-ended human simulation tasks previously dominated by LLMs. By reformulating these tasks as link prediction on a graph, GEMS learns from the relational structure of choices, complemented by a lightweight LLM-to-GNN projection. We test three settings, including generalization to entirely new individuals or questions, and three datasets spanning opinion, cognitive, and educational domains. Across these settings and datasets, GEMS matches or surpasses the strongest LLM baselines, while offering nuanced insights and practical advantages. Our work moves beyond the assumed dominance of LLMs for human simulation, offering lightweight alternatives without sacrificing predictive performance.

8 Limitations

Limitations of the graph construction. In Figure 1, we encode individual features via subgroup nodes and connect individual nodes to subgroup nodes with membership edges. The formulation is flexible: it admits different subgroup granularities (e.g., intersectional groups), alternative features (e.g., psychometrics test results), or even peer–to-peer topology that links individuals by social ties as in social recommendation (Fan et al., 2019). However, in the datasets used here, only basic demographic attributes (Santurkar et al., 2023; Toubia et al., 2025) and pre-question survey responses (Jansen et al., 2021) were available as individual features. Exploring alternative graph constructions with richer features and analyzing their effects is an important future work.

Limitations of dataset coverage. Experiments use OPINIONQA, TWIN-2K, and a replication study of Dunning-Kruger effect, all participants from the United States. Generalization to other countries or languages is untested. We note, however, that GEMS primarily learns from relational structure and uses language representations only when necessary (Setting 3), making it less sensitive to the interface language than LLM-based simulation methods. By contrast, prior works document that LLM performance can vary substantially by language; this has been shown in public opinion simulation across countries (Qu and Wang, 2024) as well as in multilingual benchmarks (Singh et al., 2024). Accordingly, GEMS may offer robustness when linguistic variation is large, though this claim should be validated empirically with non-English contexts.

Limitations of performance comparisons. Our LLM-based methods are fine-tuned up to $\sim$10B parameters. Larger models may further improve with fine-tuning. However, our experimental results show that after SFT or few-shot fine-tuning, performance gaps across LLMs narrow (Table 1, 2, 3), indicating that GEMS would remain competitive to fine-tuning larger LLMs. Also, our compute figures (GPU hours, parameter counts) are not definitive in the sense that they vary with hardware, quantization, implementation of kernels, and model architecture details. We still expect that the relative orders of compute figures would remain stable due to the significant difference in model sizes and input data formulation between LLMs and GNNs. To support an informed comparison, we present the implementation details in Section D.

Insight claims. Dot-product decoding based on output node embeddings makes the mechanics of prediction transparent (scores factor as similarities), but they are not causal explanations. Qualitative inspection may risk being misread as normative judgments about groups. Therefore, we suggest using them as diagnostic tools, complemented by ablations and sensitivity analyses.

9 Ethical Considerations

Privacy. The graph in Figure 1 is constructed from de-identified individual features and response histories provided under the original data providers' terms of use. We neither collect nor store direct identifiers (e.g., names, addresses), and all analyses are performed on anonymized records. To reduce identification risk, we report aggregate metrics (e.g., mean test accuracy) and do not release person-identifiable outputs. For future work, we recommend treating individual-level data as sensitive, and adhering to applicable regulations, institutional review, and data-security best practices.

Responsible human simulation. Even when predictive accuracy is high, simulated responses must not replace human participants. Human simulations should be carefully validated against historical data and deployed with guardrails, such as continued checks with human data, and with the goal of augmenting—not replacing—human studies. Encoding people via demographic membership edges can inadvertently reinforce stereotypes or obscure within-group heterogeneity; over-reliance on subgroup signals risks reproducing historical biases rather than revealing true relations. Therefore, we are against any deployment without governance, informed consent, and human oversight aligned with ethical guidelines.

Acknowledgments

The authors thank Minwoo Kang, Prof. John Canny, and Prof. Emma Pierson for constructive comments and valuable feedback. This work was supported in part by the Korea Foundation for Advanced Studies and Google Research Scholar Program, and by compute resources from VESSL AI, the Center for Human-Compatible AI at Berkeley, and the BAIR-Google Commons program. The views and findings expressed are those of the authors and should not be interpreted as representing the official views or policies of any sponsor.

Use of AI Statement

We used generative AI (ChatGPT, Claude) to (1) help locate related work and relevant domains, and (2) edit writing for potential grammatical mistakes.

References

Anastasios N Angelopoulos, Stephen Bates, Clara Fannjiang, Michael I Jordan, and Tijana Zrnic. 2023. Prediction-powered inference. *Science*, 382(6671):669–674.

Jacy Reese Anthis, Ryan Liu, Sean M Richardson, Austin C Kozlowski, Bernard Koch, James Evans, Erik Brynjolfsson, and Michael Bernstein. 2025. Llm social simulations are a promising research method. *arXiv preprint arXiv:2504.02234*.

Artificial Societies Artificial Societies. 2025. Artificial societies — company profile. https://www.ycombinator.com/companies/artificial-societies. Accessed: 2025-09-21.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. 2018. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*.

Moshe Ben-Akiva, Joan Walker, Adriana T Bernardino, Dinesh A Gopinath, Taka Morikawa, and Amalia Polydoropoulou. 2002. Integration of choice and latent variable models. *Perpetual motion: Travel behaviour research opportunities and application challenges*, 2002:431–470.

Rianne van den Berg, Thomas N Kipf, and Max Welling. 2017. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263*.

Federico Bianchi, Patrick John Chia, Mert Yuksekgonul, Jacopo Tagliabue, Dan Jurafsky, and James Zou. 2024. How well can llms negotiate? negotiationarena platform and analysis. *arXiv preprint arXiv:2402.05863*.

Marcel Binz, Elif Akata, Matthias Bethge, Franziska Brändle, Fred Callaway, Julian Coda-Forno, Peter Dayan, Can Demircan, Maria K Eckstein, Noémi Éltető, et al. 2025. A foundation model to predict and capture human cognition. *Nature*, pages 1–8.

James Bisbee, Joshua D. Clinton, Cassy Dorff, Brenton Kenkel, and Jennifer M. Larson. 2024. Synthetic replacements for human survey data? the perils of large language models. *Political Analysis*.

Yong Cao, Haijiang Liu, Arnav Arora, Isabelle Augenstein, Paul Röttger, and Daniel Hershcovich. 2025. Specializing large language models to simulate survey response distributions for global populations. *arXiv preprint arXiv:2502.07068*.

Pew Research Center. 2024. Issues and the 2024 election. https://www.pewresearch.org/politics/2024/09/09/issues-and-the-2024-election/. Accessed: 2026-01-06.

Runjin Chen, Tong Zhao, Ajay Jaiswal, Neil Shah, and Zhangyang Wang. 2024. Llaga: Large language and graph assistant. *arXiv preprint arXiv:2402.08170*.

Myra Cheng, Tiziano Piccardi, and Diyi Yang. 2023. Compost: Characterizing and evaluating caricature in llm simulations. In *EMNLP*.

Eric Chu, Jacob Andreas, Stephen Ansolabehere, and Deb Roy. 2023. Language models trained on media diets can predict public opinion. *arXiv preprint arXiv:2303.16779*.

Chunyuan Deng, Yilun Zhao, Yuzhao Heng, Yitong Li, Jiannan Cao, Xiangru Tang, and Arman Cohan. 2024. Unveiling the spectrum of data contamination in language models: A survey from detection to remediation. *arXiv preprint arXiv:2406.14644*.

Ricardo Dominguez-Olmedo, Moritz Hardt, and Celestine Mendler-Dünner. 2023. Questioning the survey responses of large language models. *arXiv preprint arXiv:2306.07951*.

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

Expected Parrot Expected Parrot. 2025. Expected parrot. https://www.expectedparrot.com/. Accessed: 2025-09-21.

Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In *The world wide web conference*, pages 417–426.

Shangbin Feng, Taylor Sorensen, Yuhan Liu, Jillian Fisher, Chan Young Park, Yejin Choi, and Yulia Tsvetkov. 2024. Modular pluralism: Pluralistic alignment via multi-LLM collaboration. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4151–4171.

Matthias Fey and Jan Eric Lenssen. 2019. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*.

Chen Gao, Xiaochong Lan, Nian Li, Yuan Yuan, Jingtao Ding, Zhilun Zhou, Fengli Xu, and Yong Li. 2024. Large language models empowered agent-based modeling and simulation: a survey and perspectives. *Humanities and Social Sciences Communications*, 11(1259).

William H Greene and David A Hensher. 2003. A latent class model for discrete choice analysis: contrasts with mixed logit. *Transportation Research Part B: Methodological*, 37(8):681–698.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 639–648.

Xiaoxin He, Xavier Bresson, Thomas Laurent, Adam Perold, Yann LeCun, and Bryan Hooi. 2023. Harnessing explanations: Llm-to-lm interpreter for enhanced text-attributed graph representation learning. *arXiv preprint arXiv:2305.19523*.

Zihao He, Minh Duc Chu, Rebecca Dorn, Siyi Guo, and Kristina Lerman. 2024. Community-cross-instruct: Unsupervised instruction generation for aligning large language models to online communities. *arXiv preprint arXiv:2406.12074*.

Luke Hewitt, Ashwini Ashokkumar, Isaias Ghezae, and Robb Willer. 2024. Predicting results of social science experiments using large language models. Technical report, Stanford University and New York University.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Zhengyu Hu, Yichuan Li, Zhengyu Chen, Jingang Wang, Han Liu, Kyumin Lee, and Kaize Ding. 2024. Let's ask gnn: Empowering large language model for graph in-context learning. *arXiv preprint arXiv:2410.07074*.

Angel Hsing-Chi Hwang, Michael S Bernstein, S Shyam Sundar, Renwen Zhang, Manoel Horta Ribeiro, Yingdan Lu, Serina Chang, Tongshuang Wu, Aimei Yang, Dmitri Williams, et al. 2025. Human subjects research in the age of generative ai: Opportunities and challenges of applying llm-simulated data to hci studies. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pages 1–7.

EunJeong Hwang, Bodhisattwa Prasad Majumder, and Niket Tandon. 2023. Aligning language models to user opinions. *arXiv preprint arXiv:2305.14929*.

Rachel A Jansen, Anna N Rafferty, and Thomas L Griffiths. 2021. A rational model of the dunning–kruger effect supports insensitivity to evidence in low performers. *Nature Human Behaviour*, 5(6):756–763.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Minwoo Kang, Suhong Moon, Seung Hyeong Lee, Ayush Raj, Joseph Suh, and David M Chan. 2025. Higher-order binding of language model virtual personas: a study on approximating political partisan misperceptions. *arXiv preprint arXiv:2504.11673*.

Jaehyung Kim and Yiming Yang. 2025. Few-shot personalization of llms with mis-aligned responses. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 11943–11974.

Junsol Kim, James Evans, and Aaron Schein. 2025. Linear representations of political perspective emerge in large language models. In *The Thirteenth International Conference on Learning Representations*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Thomas N Kipf and Max Welling. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*.

Akaash Kolluri, Shengguang Wu, Joon Sung Park, and Michael S. Bernstein. 2025. Finetuning llms for human behavior prediction in social science experiments.

Maximilian Kreutner, Marlene Lutz, and Markus Strohmaier. 2025. Persona-driven simulation of voting behavior in the european parliament with large language models. *arXiv preprint arXiv:2506.11798*.

Stefan Krsteski, Giuseppe Russo, Serina Chang, Robert West, and Kristina Gligorić. 2025. Valid survey simulations with limited human data: The roles of prompting, fine-tuning, and rectification. *arXiv preprint arXiv:2510.11408*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30.

Ang Li, Haozhe Chen, Hongseok Namkoong, and Tianyi Peng. 2025. Llm generated persona is a promise with a catch. *arXiv preprint arXiv:2503.16527*.

Cheng Li, Mengzhuo Chen, Jindong Wang, Sunayana Sitaram, and Xing Xie. 2024. Culturellm: Incorporating cultural differences into large language models. *Advances in Neural Information Processing Systems*, 37:84799–84838.

Gechun Lin and Christopher Lucas. 2023. An introduction to neural networks for the social sciences. *Oxford Handbook of Engaged Methodological Pluralism in Political Science*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098.

Benjamin S Manning, Kehang Zhu, and John J Horton. 2024. Automated social science: Language models as scientist and subjects. Technical report, National Bureau of Economic Research.

Daniel McFadden and Kenneth Train. 2000. Mixed mnl models for discrete response. *Journal of applied Econometrics*, 15(5):447–470.

Suhong Moon, Marwa Abdulhai, Minwoo Kang, Joseph Suh, Widyadewi Soedarmadji, Eran Kohen Behar, and David M Chan. 2024. Virtual personas for language models via an anthology of backstories. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 19864–19897.

OpenAI. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

Joon Sung Park, Carolyn Q Zou, Aaron Shaw, Benjamin Mako Hill, Carrie Cai, Meredith Ringel Morris, Robb Willer, Percy Liang, and Michael S Bernstein. 2024. Generative agent simulations of 1,000 people. *arXiv preprint arXiv:2411.10109*.

PersonaLLM'25. 2026. Workshop on llm persona modeling @ neurips 2025. https://personallmworkshop.github.io/. Accessed: 2026-01-05.

PewResearch. 2018. America trends panel waves. Retrieved February 06, 2025, from https://www.pewsocialtrends.org/dataset.

Yao Qu and Jue Wang. 2024. Performance and biases of large language models in public opinion simulation. *Humanities and Social Sciences Communications*, 11(1):1–13.

Vahid Rahimzadeh, Erfan Moosavi Monazzah, Mohammad Taher Pilehvar, and Yadollah Yaghoobzadeh. 2025. Synthia: Synthetic yet naturally tailored human-inspired personas. *arXiv preprint arXiv:2507.14922*.

Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2024. Representation learning with large language models for recommendation. In *Proceedings of the ACM web conference 2024*, pages 3464–3475.

David M Rothschild, James Brand, Hope Schroeder, and Jenny Wang. 2024. Opportunities and risks of llms in survey research. *Available at SSRN*.

Shibani Santurkar, Esin Durmus, Faisal Ladhak, Cinoo Lee, Percy Liang, and Tatsunori Hashimoto. 2023. Whose opinions do language models reflect? In *International Conference on Machine Learning*, pages 29971–30004. PMLR.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In *The Twelfth International Conference on Learning Representations*.

Leheng Sheng, An Zhang, Yi Zhang, Yuxin Chen, Xiang Wang, and Tat-Seng Chua. 2025. Language representations can be what recommenders need: Findings and potentials. In *ICLR*.

Shivalika Singh, Angelika Romanou, Clémentine Fourrier, David I Adelani, Jian Gang Ngui, Daniel Vila-Suero, Peerat Limkonchotiwat, Kelly Marchisio, Wei Qi Leong, Yosephine Susanto, et al. 2024. Global mmlu: Understanding and addressing cultural and linguistic biases in multilingual evaluation. *arXiv preprint arXiv:2412.03304*.

SocialSim'25. 2025. Social simulation with llms @ colm 2025. https://sites.google.com/view/social-sims-with-llms. Accessed: 2025-09-21.

Joseph Suh, Erfan Jahanparast, Suhong Moon, Minwoo Kang, and Serina Chang. 2025. Language model fine-tuning on scaled survey data for predicting distributions of public opinions. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 21147–21170.

Yuanfu Sun, Zhengnan Ma, Yi Fang, Jing Ma, and Qiaoyu Tan. 2025. Graphicl: Unlocking graph learning potential in llms through structured prompt design. *arXiv preprint arXiv:2501.15755*.

Curt Tigges, Oskar John Hollinsworth, Atticus Geiger, and Neel Nanda. 2023. Linear representations of sentiment in large language models. *arXiv preprint arXiv:2310.15154*.

Kiran Tomlinson and Austin R Benson. 2024. Graph-based methods for discrete choice. *Network Science*, 12(1):21–40.

Olivier Toubia, George Z Gui, Tianyi Peng, Daniel J Merlau, Ang Li, and Haozhe Chen. 2025. Twin-2k-500: A data set for building digital twins of over 2,000 people based on their answers to over 500 questions. *Marketing Science*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Kenneth E Train. 2009. *Discrete choice methods with simulation*. Cambridge university press.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Leah von der Heyde, Anna-Carolina Haensch, and Alexander Wenz. 2025. Vox populi, vox ai? using large language models to estimate german vote choice. *Social Science Computer Review*, page 08944393251337014.

Jimmy Wang, Thomas Zollo, Richard Zemel, and Hongseok Namkoong. 2025. Adaptive elicitation of latent information using natural language. *arXiv preprint arXiv:2504.04204*.

Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019a. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 950–958.

Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019b. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, pages 165–174.

Zichao Wang, Angus Lamb, Evgeny Saveliev, Pashmina Cameron, Yordan Zaykov, José Miguel Hernández-Lobato, Richard E Turner, Richard G Baraniuk, Craig Barton, Simon Peyton Jones, et al. 2020. Instructions and guide for diagnostic questions: The neurips 2020 education challenge. *arXiv preprint arXiv:2007.12061*.

Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 726–735.

Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 346–353.

Yutong Xie, Zhuoheng Li, Xiyuan Wang, Yijun Pan, Qijia Liu, Xingzhi Cui, Kuang-Yu Lo, Ruoyi Gao, Xingjian Zhang, Jin Huang, et al. 2025. Be. fm: Open foundation models for human behavior. *arXiv preprint arXiv:2505.23058*.

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

Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y Chang. 2015. Network representation learning with rich text information. In *IJCAI*, volume 2015, pages 2111–2117.

Binwei Yao, Zefan Cai, Yun-Shiuan Chuang, Shanglin Yang, Ming Jiang, Diyi Yang, and Junjie Hu. 2025. No preference left behind: Group distributional preference optimization. In *The Thirteenth International Conference on Learning Representations*.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7370–7377.

Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 974–983.

Chenxiao Yu, Zhaotian Weng, Yuangang Li, Zheng Li, Xiyang Hu, and Yue Zhao. 2024. A large-scale empirical study on large language models for election prediction. *CoRR*.

Junliang Yu, Xin Xia, Tong Chen, Lizhen Cui, Nguyen Quoc Viet Hung, and Hongzhi Yin. 2023. Xsimgcl: Towards extremely simple graph contrastive learning for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 36(2):913–926.

Daokun Zhang, Jie Yin, Xingquan Zhu, and Chengqi Zhang. 2019. Attributed network embedding via subspace discovery. *Data Mining and Knowledge Discovery*, 33(6):1953–1980.

Jianan Zhao, Meng Qu, Chaozhuo Li, Hao Yan, Qian Liu, Rui Li, Xing Xie, and Jian Tang. 2022. Learning on large-scale text-attributed graphs via variational inference. *arXiv preprint arXiv:2210.14709*.

Siyan Zhao, John Dang, and Aditya Grover. 2024. Group preference optimization: Few-shot alignment of large language models. In *The Twelfth International Conference on Learning Representations*.

Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, Haofei Yu, Zhengyang Qi, Louis-Philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, et al. 2024. Sotopia: Interactive evaluation for social intelligence in language agents. In *The Twelfth International Conference on Learning Representations*.

Jason Zhu, Yanling Cui, Yuming Liu, Hao Sun, Xue Li, Markus Pelger, Tianqi Yang, Liangjie Zhang, Ruofei Zhang, and Huasha Zhao. 2021. Textgnn: Improving text encoder via graph neural network in sponsored search. In *Proceedings of the Web Conference 2021*, pages 2848–2857.

A Extended Related work

GNN recommender systems. Relational inductive biases are central to graph recommenders that represent user–item interactions as edges (Battaglia et al., 2018). From GCMC (Berg et al., 2017), GNNs explicitly leverage higher-order connectivity, including PinSage (Ying et al., 2018), NGCF (Wang et al., 2019b), and simplified designs like LightGCN (He et al., 2020). Furthermore, knowledge-graph-aware models capture attribute/item relations (Wang et al., 2019a) and complementary directions capture session and social structures (Wu et al., 2019; Fan et al., 2019) or harness contrastive signals on graphs (Wu et al., 2021; Yu et al., 2023). These successes suggest that human choices and behaviors are inherently relational. We draw on these insights to bridge two largely disjoint literature: LLM-based simulation and graph-based modeling.

Discrete choice modeling. Classical 'human simulation' in discrete choice has been based on latent variable frameworks (Train, 2009; Lin and Lucas, 2023), random-utility models such as mixed logit (McFadden and Train, 2000), latent-class/finite-mixture models (Greene and Hensher, 2003), integrated choice-and-latent-variable hybrids and hierarchical Bayes methods for individualized posteriors (Ben-Akiva et al., 2002). More recently, graph learning offers a promise of complementary inductive bias that exploits interaction topology: Tomlinson and Benson (2024) outline several ways of integration, such as GNN-based chooser embeddings and propagation of local choice rates. Building on these insights and the latent variable tradition, GEMS casts discrete choice simulation as a link prediction on a heterogeneous graph. By contrast, current LLM-based approaches primarily leverage parametric knowledge from pretraining.

Text-attributed graphs (TAGs). TAGs integrate node and relation's text attributes with graph topology, letting models enjoy complementary signals. Early work injected text features into matrix-factorization formulations or constructed word–document graphs (Yang et al., 2015; Yao et al., 2019). More recently, an LLM-to-GNN interplay has emerged: (i) LLM as encoder/feature generator, using an LLM as an encoder whose embeddings serve as GNN inputs (Zhu et al., 2021); (ii) alternating, EM-style training that decouples text and graph modules while co-training them via variational objectives (Zhao et al., 2022); and (iii) prompting LLMs to generate descriptions or explanations that enrich node attributes (He et al., 2023). A complementary line of work conditions LLMs on graph structure through prompting and in-context learning, including AskGNN (Hu et al., 2024) and GraphICL (Sun et al., 2025). Related efforts project graphs directly into an LLM's token space or align GNN embeddings with token embeddings so that an LLM can reason over graph tokens (Chen et al., 2024). In recommender systems, the inverse mapping uses LLM representations within learned graph / collaborative-filtering spaces or co-trains them with GNNs (Ren et al., 2024; Sheng et al., 2025). Collectively, these works underscore the complementarity of language and graph signals. GEMS leverages these insights for human simulation on discrete choice tasks, especially on prediction for new questions where an appropriate mapping between language representations and graph representations is necessary.

B Dataset Details

B.1 OpinionQA

OPINIONQA (Santurkar et al., 2023) is a curated subset of the American Trends Panel (ATP) (PewResearch, 2018). It comprises 500 contentious questions drawn from 14 ATP survey waves, selected for large inter-group response differences. For each anonymized participant, information across 9 demographic traits (age, gender, race or ethnicity, highest level of education, annual income, Census Bureau regions, religion, political affiliation, and political ideology) and their response to survey questions are available. Survey items span a wide range of social topics, including race, politics, age-specific attitudes, media consumption, and views on the future of AI. Owing to its breadth and diversity, OPINIONQA has become a popular dataset for LLM-based human simulation or pluralistic preference alignment research (Hwang et al., 2023; Feng et al., 2024; Zhao et al., 2024; Moon et al., 2024; Kolluri et al., 2025). Here we present

three example questions from the dataset.

Question Example: OpinionQA (1)

Question: Which of the following would you say you prefer for getting news?

A. A print newspaper
B. Radio
C. Television
D. A social media site (such as Facebook, Twitter or Snapchat)
E. A news website or app

Question Example: OpinionQA (2)

Question: In the future, what kind of an impact do you think the military will have in solving the biggest problems facing the country?

A. A very positive impact
B. A somewhat positive impact
C. A somewhat negative impact
D. A very negative impact

Question Example: OpinionQA (3)

Question: For each, please indicate if you, personally, think it is acceptable. Casting an actor to play a character of a race or ethnicity other than their own

A. Always acceptable
B. Sometimes acceptable
C. Rarely acceptable
D. Never acceptable
E. Not sure

B.2 Twin-2K

Twin-2K (Toubia et al., 2025) is a four-wave, nationally representative U.S. panel fielded in January – February 2025 on Prolific for LLM human simulation. Each participant completed questions spanning demographic information, personality scales, cognitive ability tests, economic preference, heuristics-and-biases experiments, etc. Among all questions from Twin-2K, we filtered for multiple-choice questions by removing short answer questions, resulting in 150 questions total. The authors release the full dataset publicly to support broader social-science research.

Question Example: Twin-2K (1)

Choose an option.

A. I don't feel like a failure
B. I feel that I have failed more than the average person
C. As I look back on my life, all I can see is a lot of failures
D. I feel I am a complete failure as a person

Question Example: Twin-2K (3)

Antonym: Select the word that is most nearly the opposite in meaning to DEARTH

A. birth
B. brevity
C. abundance
D. splendor
E. renaissance

Question Example: Twin-2K (2)

You have recently graduated from university, obtained a good job, and are buying a new car. A newly designed seatbelt has just become available that would save the lives of 95% of the 500 drivers a year who are involved in a type of head-on-collision. (Approximately half of these fatalities involve drivers who were not at fault.) The newly designed seatbelt is not yet standard on most car models. However, it is available as a $500 option for the car model that you are ordering. How likely is it that you would order your new car with this optional seatbelt?

A. very unlikely
B. unlikely
C. somewhat unlikely
D. somewhat likely
E. likely
F. very likely

B.3 Replication Study of the Dunning-Kruger Effect

Replication study of the Dunning–Kruger effect (referred to as DUNNING-KRUGER throughout) (Jansen et al., 2021) is a single-wave, U.S.-based study. 3K participants first completed four pre-question survey items on self-expected problem-

solving ability and confidence level, then answered 20 multiple-choice questions on grammar and logical reasoning (five options per question, exactly one correct). After the test, they completed four post-question survey items. Our prediction target is the option a participant would select—not the correct answer—aligning with previous LLM human simulation objective (Binz et al., 2025) and prediction tasks of interest in intelligent tutoring systems (Wang et al., 2020). We present two samples from the 20 questions and one pre-question survey.

Question Example: Dunning-Kruger (1)

Compared to other participants in this study, how well do you think you will do?
Marking 90% means you will do better than 90% of participants, marking 10% means you will do better than only 10%, and marking 50% means that you will perform better than half of the participants.

Question Example: Dunning-Kruger (2)

Some part of the sentence is in square brackets.
Five choices for rephrasing that part follow the sentence; one choice repeats the original, and the other four are different.
Your task is to select the grammatically correct choice.

The school-age child faces a formidable task when during the first few years of classroom experiences [he or she is expected to master the printed form of language.]

A. he or she expects to master the printed form of language.
B. he or she is expected to master the printed form of language.
C. he or she faces expectations of mastering the printed form of language.
D. mastery of the printed form of language is expected of him or her.
E. mastery of print is expected by his or her teacher.

Question Example: Dunning-Kruger (3)

Some part of the sentence is in square brackets.
Five choices for rephrasing that part follow the sentence; one choice repeats the original, and the other four are different.
Your task is to select the grammatically correct choice.

[The belief of ancient scientists was] that maggots are generated from decaying bodies and filth and are not formed by reproduction.

A. The belief of ancient scientists was
B. The ancient scientists beliefs were
C. The ancient scientists believe
D. The belief of ancient scientists were
E. The ancient belief of scientists was

C Graph Statistics

In this section, we present the graph statistics, including: the number of nodes and edges of the graph formulation (Figure 1) and the construction of subgroup nodes.

C.1 OpinionQA

We followed the dataset filtering process of (Zhao et al., 2024). Beginning with 76K participants in OpinionQA dataset, filtering to those who answered at least 30 questions yields 19K individuals, 284 survey questions, and 695K (individual, question, choice) triples. 284 survey questions have total 1,103 choices, indicating that each survey question has 3.88 available choices on average. Because the number of individual nodes (19K) is much larger than the number of choice nodes (1,103), choice nodes have much higher average degree than individual nodes.

To define subgroup nodes, we employ the 9 demographic attributes used in previous works (Santurkar et al., 2023): age, gender, race or ethnicity, education level, annual income, Census regions, religion, political affiliation, and political ideology. This results in 48 subgroup nodes:

Age : 18-29, 30-49, 50-64, 65+

Race or ethnicity : White, Black, Hispanic, Asian, Other

Gender : Male, Female, Other

Education : Less than high school, High school graduate, Some college, no degree, Associate's degree, College graduate / some postgrad, Postgraduate

Annual income : Less than $30,000, $30,000 – $50,000, $50,000 – $75,000, $75,000 – $100,000, $100,000 or more

Region : Northeast, Midwest, South, West

Religion : Protestant, Roman Catholic, Mormon, Orthodox, Jewish, Muslim, Buddhist, Hindu, Atheist, Agnostic, Other, Nothing in particular

Political affiliation : Republican, Democrat, Independent, Something else

Political ideology : Very conservative, Conservative, Moderate, Liberal, Very liberal

We note that there can be different constructions of subgroup nodes, either by considering additional individual features (e.g., marital status, country of birth, etc.) or intersectional attributes as a single subgroup node (e.g., construct a subgroup node representing 'age 18-29 male'). Future work can design their own subgroup nodes tailored to the specific need, and our construction is easily generalizable in those settings.

C.2 Twin-2K

TWIN-2K dataset includes both multiple choice and short answer questions. To align with our focus on discrete choice human simulation tasks, we exclude short-answer items, yielding 150 multiple-choice questions. Because nearly all of the 2,000 participants responded to most multiple choice questions, no individuals were removed by the minimum-30-responses criterion. The dataset authors (Toubia et al., 2025) collect demographics using the same categories as (Santurkar et al., 2023): we reuse the identical 48 subgroup definitions as in OPINIONQA. The resulting graph contains 48 subgroup nodes, 2,000 individual nodes, 539 choice nodes, and 297K response edges.

C.3 Replication Study of the Dunning-Kruger Effect

Unlike the previous two datasets, this study does not include participant demographics. Instead, we define individual features and derive subgroup nodes from responses to four pre-question survey items. Choice nodes are constructed from 5 options for each of 20 questions, 100 in total.

Expected correctness (How many of the 20 questions do you think you will answer correctly?): 0, 1, 2, ..., 20, total 21 subgroups

Expected performance (How well do you think you will do in percentile?): 0-10, 11-20, 21-30, ..., 91-100, total 10 subgroups

Confidence of self (How difficult is recognizing correct grammar for you?): 0, 1, 2, ..., 10

Expected confidence of others (How difficult is recognizing correct grammar for the average participant?): 0, 1, 2, ..., 10, total 11 subgroups

D Implementation Details

This section details the implementation of the GNN (Section 4.2) and the LLM baselines (Section 5.1). We first present the general GNN training configuration in Section D.1, followed by the learnable input embedding tables in Section D.2. Next, we instantiate the graph encoder in (1) with three architectures—RGCN, GAT, and GraphSAGE—in Sections D.3 to D.5, respectively. Section D.6 describes the setup for the LLM-based methods. Finally, Section D.7 compares the model size of LLMs and GEMS GNNs.

D.1 GNN Training Configuration

We implement GNNs based on PyTorch Geometric (Fey and Lenssen, 2019). All trainable components of the GNN (learnable input embedding tables, graph encoder, and decoding temperature) are optimized with AdamW optimizer (Loshchilov and Hutter, 2017) using a learning rate of 5×10^{-4} and weight decay of 10^{-3}. We use a cosine annealing learning rate scheduler and apply gradient clipping with a max norm of 0.1.

Each GNN is trained for $1{,}000$ epochs with a patience of 30 (i.e., how many epochs the model would continue training after the validation loss stops decreasing). In Section 6, the reported GNN training time is measured from the beginning of the training until the termination by exceeding patience. An epoch consists of $50n$ steps, where n is the number of training graphs. Concretely, each training graph is sampled 50 times per epoch with independently re-drawn edge masks that split train response edges into message-passing edges and supervision edges. This resampling reduces overfitting to a fixed edge partition and consistently improves validation accuracy and its variance.

Setting 3 (New questions). After the GNN is fully trained, we learn an LLM-to-GNN mapping as described in Equation (4). The mapping is obtained by solving ridge regression with a regularization strength α. Rather than a cross-validation, we choose α by directly maximizing validation prediction accuracy on held-out validation questions. In Section 6, the training time of LLM-to-GNN mapping is calculated as the time to extract LLM hidden states from textual features of choice nodes, since solving the ridge regression takes negligible amount of time. In practice, $\alpha \in [50,500]$ performs best.

D.2 Learnable Input Feature Table

In Section 4.2, we denote a learnable input feature table for subgroup nodes $\mathcal{S}$ as $Z_{\mathcal{S}} \in \mathbb{R}^{|\mathcal{S}| \times d_{\mathcal{S}}}$ and choice nodes $\mathcal{C}$ as $Z_{\mathcal{C}} \in \mathbb{R}^{|\mathcal{C}| \times d_{\mathcal{C}}}$. We set $d_{\mathcal{S}} = 16$ and $d_{\mathcal{C}} = 128$ for all settings on the OPINIONQA dataset, and $d_{\mathcal{S}} = 8$ and $d_{\mathcal{C}} = 64$ for all settings on the TWIN-2K and DUNNING-KRUGER dataset.

D.3 Relational Graph Convolution (RGCN)

We use a 2-layer RGCN (Schlichtkrull et al., 2018). Following the feature table dimension in D.2, input dimensions are $(16,1,128)$ for (subgroup, individual, choice) nodes on OPINIONQA dataset and $(8,1,64)$ on others. All hidden layers use the choice node's input dimension, i.e., 128 for OPINIONQA and 64 for TWIN-2K and DUNNING-KRUGER.

In (1), we present the general graph encoder forward pass as

$$z_w^{(\ell+1)} = \sigma\left(\sum_{r \in \mathcal{R}} \left[\underset{v \in \mathcal{N}_r(w)}{\mathrm{AGG}_r} \phi_r^{(\ell)}\left(z_w^{(\ell)}, z_v^{(\ell)}\right)\right] + \phi_{\text{self}}^{(\ell)}\left(z_w^{(\ell)}\right)\right), \qquad \ell = 0,\ldots,L-1 \tag{5}$$

For RGCN, we use ReLU as the non-linear activation σ and a mean pooling for AGG_r for all relations r. Following the standard RGCN implementation, a relation-specific message passing is

$$\phi_r^{(\ell)}\left(z_w^{(\ell)}, z_v^{(\ell)}\right) = \frac{1}{|\mathcal{N}_r(w)|} \mathbf{W}_r^{(\ell)} z_v^{(\ell)}, \tag{6}$$

where the learnable $\mathbf{W}_r^{(\ell)}$ maps from the layer-ℓ embedding of the node v to the layer-$(\ell+1)$ embedding dimension of node w; the factor $|\mathcal{N}_r(w)|^{-1}$ provides degree normalization for relation r. Similarly, self-loops use a learnable matrix $\mathbf{W}_{\text{self},t(w)}^{(\ell)}$ that is node-type specific:

$$\phi_{\text{self}}^{(\ell)}\left(z_w^{(\ell)}\right) = \mathbf{W}_{\text{self},t(w)}^{(\ell)} z_w^{(\ell)}. \tag{7}$$

where $t(w)$ represents the node type of the node w. Additionally, we apply a post-activation LayerNorm (Ba et al., 2016) and dropout with rate 0.5 at all layers of the graph encoder.

D.4 Graph Attention Network (GAT)

Equation (1) is implemented with a multi-head Graph Attention Network (Veličković et al., 2018)

$$z_w^{(\ell+1)} = \sigma\left(\sum_{r \in \mathcal{R}} \left[\Big\Vert_{h=1}^{H_\ell} \sum_{v \in \mathcal{N}_r(w) \cup \{w\}} \alpha_{wv,r}^{(\ell,h)} \mathbf{\Theta}_{t,r}^{(\ell,h)} z_v^{(\ell)}\right]\right) \tag{8}$$

where $\|$ denotes concatenation across heads, h indicates the head index ranging from 1 to the number of heads in the ℓ-th layer (H_ℓ), and the attention coefficient α for the layer-ℓ head-h relation-r from the source node v to the target node w is

$$\alpha_{wv,r}^{(\ell,h)} = \underset{v \in \mathcal{N}_r(w) \cup \{w\}}{\mathrm{softmax}}\Big(\mathrm{LeakyReLU}\Big(\mathbf{a}_{s,r}^{(\ell,h)\top} \mathbf{\Theta}_{s,r}^{(\ell,h)} z_w^{(\ell)} + \mathbf{a}_{t,r}^{(\ell,h)\top} \mathbf{\Theta}_{t,r}^{(\ell,h)} z_v^{(\ell)}\Big)\Big). \tag{9}$$

where $\mathbf{a}_{s,r}^{(\ell,h)}$ and $\mathbf{a}_{t,r}^{(\ell,h)}$ are learnable source and target scoring vectors, $\mathbf{\Theta}_{s,r}^{(\ell,h)}$ and $\mathbf{\Theta}_{t,r}^{(\ell,h)}$ are learnable source and target feature transformation matrix, and $\mathrm{LeakyReLU}$ is a LeakyReLU function with a negative slope of 0.2 as in the default implementation of PyTorch Geometric. Softmax is performed over all neighboring nodes of w defined by the relation r and w itself.

We use a 2-layer GAT. Following the input table dimension in D.2, input feature dimensions are $(16,1,128)$ for (subgroup, individual, choice) nodes on the OPINIONQA dataset and $(8,1,64)$ on others. All hidden layers use the choice node's input dimension (128 for OPINIONQA, 64 for TWIN-2K and DUNNING-KRUGER) with 4 heads in the first layer (per-head size of 32 or 16) and 1 head in the second layer (per-head size of 128 or 64), keeping the hidden size unchanged across layers.

We set $\sigma = \mathrm{ReLU}$, and apply post-activation LayerNorm (Ba et al., 2016). We also apply

dropout at rate 0.4 to the normalized attention coefficients α and at rate 0.5 to the post-activation node embeddings between layers.

D.5 GraphSAGE

We instantiate the generic graph encoder in (1) with a GraphSAGE operator (Hamilton et al., 2017). For each relation $r \in \mathcal{R}$ and given a target node w, we first compute a relation-specific mean-pooled neighbor message

$$m_{w,r}^{(\ell)} = \underset{v \in \mathcal{N}_r(w)}{\text{MEAN}} \left(\mathbf{\Theta}_r^{(\ell)} z_v^{(\ell)} \right), \tag{10}$$

where $\mathbf{\Theta}_r^{(\ell)}$ is a learnable matrix that maps the layer-ℓ embedding of a source node v to the layer-$(\ell{+}1)$ embedding space of the target node for relation r. Messages from all relations are summed and combined with a learnable root (self) transformation $\mathbf{\Theta}_{\text{self}}^{(\ell)}$. Subsequently, the embedding is $L2-$normalized and passed through a non-linear activation:

$$z_w^{(\ell+1)} = \sigma\Big(\text{Normalize}\Big(\mathbf{\Theta}_{\text{self}}^{(\ell)} z_w^{(\ell)} + \sum_{r \in \mathcal{R}} m_{w,r}^{(\ell)} \Big) \Big). \tag{11}$$

We set $\sigma = \text{ReLU}$, apply post-activation LayerNorm (Ba et al., 2016) at every layer, and use dropout with rate 0.5 on the post-activation node embeddings between layers.

We use a 2-layer GraphSAGE. Following the input feature dimensions in Section D.2, input sizes are $(16,1,128)$ for (subgroup, individual, choice) nodes on OPINIONQA and $(8,1,64)$ on Twin-2K. All hidden layers use the choice-node width, i.e., 128 for OPINIONQA and 64 for Twin-2K.

D.6 LLM

For all LLM prompting experiments, we used 2×NVIDIA A100 80GB (SXM4) and vLLM framework (Kwon et al., 2023). For selecting in-context examples in few-shot prompting and Agentic CoT, we encode each question text with `gemini-embedding-001` embedding model and compute cosine similarities between the target question and candidate in-context example questions. Following Hwang et al. (2023), the selected examples are ordered by ascending cosine similarity, from least to most similar. To ensure consistent information access across methods, in-context examples are drawn exclusively from the training set and not from the validation set.

For all LLM fine-tuning methods, we also used 2×NVIDIA A100 80GB (SXM4) and built on Llama-cookbook codebase. Each run trained for three epochs using the model's default precision, and we selected the checkpoint with the lowest validation loss. We largely followed hyperparameter setting of Suh et al. (2025), tuning the learning rate over {1e-4, 2e-4, 4e-4} and settled on 2e-4. Training used LoRA with rank 8, $\alpha = 32$, and dropout 0.05, applied to the attention query and value matrices, following the implementation details of main experiment in the original LoRA paper (Hu et al., 2022). We optimized with Adam optimizer (Kingma and Ba, 2014) and the effective batch size was fixed to 256 by setting per-GPU batches and gradient accumulation steps to fit GPU VRAM. For ablation experiments with higher LoRA rank (rank 256) or full fine-tuning, refer to Section F.

D.7 Model Parameters

In Table 7, we report parameter counts for GEMS and the LLMs, following the implementation details in the previous sections. For LLMs fine-tuned with LoRA, the trainable parameter count equals the number of LoRA adapter parameters, much smaller than the total parameter count. Because both training and inference still require loading the full model weights, we use the total parameter count when comparing model size. The size of the GNN (GEMS) varies between datasets because we select the hidden dimension per dataset, as noted above.

E Additional Experiment Results

E.1 Extended Figure of Compute Hours

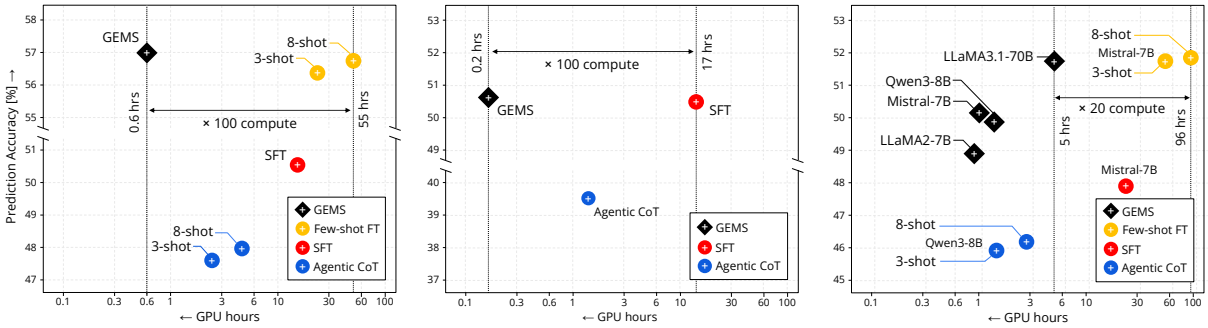
Figure 4 shows the prediction accuracy and compute hours of different LLM-based methods and GEMS. Across all settings, GEMS requires significantly less compute compared to LLM methods of comparable performance, and strictly outperforms LLM methods of comparable compute hours.

E.2 Embedding visualization

Figure 5 visualizes LLM hidden states and GNN embeddings for four example questions in OPINIONQA dataset. Each question asks how high a pri-

Table 7: Number of parameters for each model. K, M, and B stand for 10^3, 10^6, 10^9, respectively.

# parameters	LLM			GEMS (RGCN)		
Model	LLaMA2-7B	Mistral-7B	Qwen3-8B	-		
Dataset	-			OPINIONQA	TWIN-2K	DUNNING-KRUGER
Trainable	4.19 M	3.41 M	3.83 M	420 K	110 K	110 K
Total	6.61 B	7.24 B	8.19 B	420 K	110 K	110 K



Setting 1: Missing Responses (Imputation) | Setting 2: New Individuals | Setting 3: New Questions

Figure 4: Accuracy *vs.* GPU-hours (A100-80GB-SXM4) on the OPINIONQA dataset by task setting and method. Zero-/few-shot prompting accuracies fall below the plotted y-range. For LLM-based methods, we report the best result across three LLMs (LLaMA-2-7B, Mistral-7B-v0.1, and Qwen3-8B). For GEMS, we report the best result across three models (RGCN, GAT, and SAGE) for setting 1 & 2, and report across different LLMs for setting 3.

ority the federal government should give to an issue: (B) reducing illegal immigration, (C) reducing economic inequality, (D) addressing climate change, and (F) reducing gun violence, with four response options ranging from 'top priority' to 'should not be done.' This results in 16 choice nodes in total. All plots show the first two principal components of principal component analysis (PCA).

Embedding structure of choice nodes. The top left panel plots the LLM hidden states for the 16 choice nodes: points cluster by option, producing four clusters one per option but not clearly indicating what semantic meaning each choice has. The top right panel shows choice nodes' output node embeddings after the first training stage described in Section 5.4. Here, choices for three questions (C, D, F) are located along a common one-dimensional trajectory in the PCA plane, whereas the choices for question B align along a distinct trajectory. From this observation, we can infer that three questions (C, D, F) are closely related while one question (B) sits on a slightly different social issue dimension, which is consistent with prior observation from survey researchers (Center, 2024).

Embedding structure of individual nodes. The remaining panels plot GNN output node embeddings of individual nodes, with colors indicating an individual feature per panel (annual income, political ideology, age, or gender). The PCA axes exhibit interpretable variation: PC1 aligns most strongly with political ideology feature and PC2 with income. Yet, points within any given subgroup remain dispersed, indicating substantial within-group heterogeneity. We note that the prediction is made by taking dot-product between each individual node embedding and the four choice node embeddings, followed by the softmax for multinomial distribution over options.

E.3 Prediction with Additional LLMs

We expand the evaluation on Setting 1 (predicting missing responses) of the OPINIONQA dataset to include both open-weight models (LLaMA3.1-8B, LLaMA3.1-70B, Mistral-Small-24B-2501, Qwen3-32B, GPT-OSS-20B (Dubey et al., 2024; Yang et al., 2025; OpenAI, 2025)) and frontier proprietary models (GPT-5-Mini, GPT-5.2, Claude Sonnet 4.5, and Gemini 3 Flash).

Open-weight models. Table 8 reports results for open-weight models of varying sizes. Consistent with Tables 1–3, larger and more recent models generally perform better, with the largest gains appearing under Agentic CoT where reasoning ability is most critical. This trend is most pronounced for Qwen3, a reasoning model family. However, even the best prompting result among open-weight models (Qwen3-32B, Agentic CoT with 13 examples, 50.70%) falls short of few-shot

Table 8: Extended evaluation with open-weight LLMs on the OPINIONQA dataset, Setting 1 (missing responses). k indicates the number of few-shot examples. Results for LLaMA-2-7B, Mistral-7B-v0.1, and Qwen3-8B are identical to those in Table 1.

		LLaMA				**Mistral**		**Qwen3**		**GPT**
Methods	k	2-7B	2-70B	3.1-8B	3.1-70B	7B-v0.1	24B-2501	8B	32B	OSS-20B
Random						27.87				
Zero-shot		29.18	36.47	38.71	43.45	34.63	41.79	39.38	40.42	35.71
Few-shot	3	38.54	40.76	44.04	46.04	42.52	45.85	42.21	43.53	42.18
	5	39.41	43.26	44.22	47.35	44.43	45.82	42.69	46.04	44.53
	8	37.91	40.34	42.26	44.55	45.78	45.27	43.66	44.28	41.47
	13	37.78	42.45	43.24	49.66	44.03	46.41	44.03	46.72	42.50
Agentic CoT	3	32.19	43.66	42.80	49.32	41.37	46.01	47.63	47.57	44.42
	5	30.72	45.82	42.96	49.55	38.92	48.22	47.92	48.40	45.90
	8	28.80	42.63	42.15	47.72	38.43	46.24	47.97	48.44	41.94
	13	28.05	45.34	42.81	48.06	38.37	48.67	48.88	50.70	44.03
SFT		49.41	–	–	–	50.56	–	48.84	–	–
Few-shot FT	3	55.59	–	–	–	56.31	–	55.09	–	–
Few-shot FT	8	55.98	–	–	–	**56.76**	–	55.61	–	–

Table 9: Evaluation of frontier proprietary models on the OPINIONQA dataset, Setting 1 (missing responses), with prompting methods. k indicates the number of few-shot examples. Two bottom rows represent the best LLM fine-tuning and GEMS results from Table 1 for comparison.

Methods	k	GPT-5-Mini	GPT-5.2	Claude Sonnet 4.5	Gemini 3 Flash
Random			27.87		
Zero-shot		45.06	45.35	45.23	46.06
Few-shot	3	46.99	48.16	47.27	48.25
	8	47.52	49.72	50.63	50.49
Agentic CoT	3	48.83	48.68	49.34	49.19
	8	50.17	52.01	51.84	**52.78**
Few-shot FT (best)				56.76	
GEMS (best)				**57.00**	

fine-tuning (56.76%) and GEMS (57.00%).

Frontier proprietary models. Table 9 extends the comparison to frontier proprietary models, evaluated with promting methods (as fine-tuning is unavailable). First, frontier models achieve notable gains over smaller open-weight models. For example, with 8-shot Agentic CoT, Gemini 3 Flash reaches 52.78% and GPT-5.2 reaches 52.01%, compared to 47.97% for Qwen3-8B and 50.70% for Qwen3-32B. Second, even the best prompting results from frontier models do not surpass fine-tuning methods or GEMS: the strongest prompting result (Gemini 3 Flash, Agentic CoT with k=8: 52.78%) falls short of few-shot fine-tuning (56.76%) and GEMS (57.00%). This indicates diminishing returns from scaling LLMs in prompting for close-ended simulation tasks, and confirms that GEMS remains competitive even against state-of-the-art proprietary models.

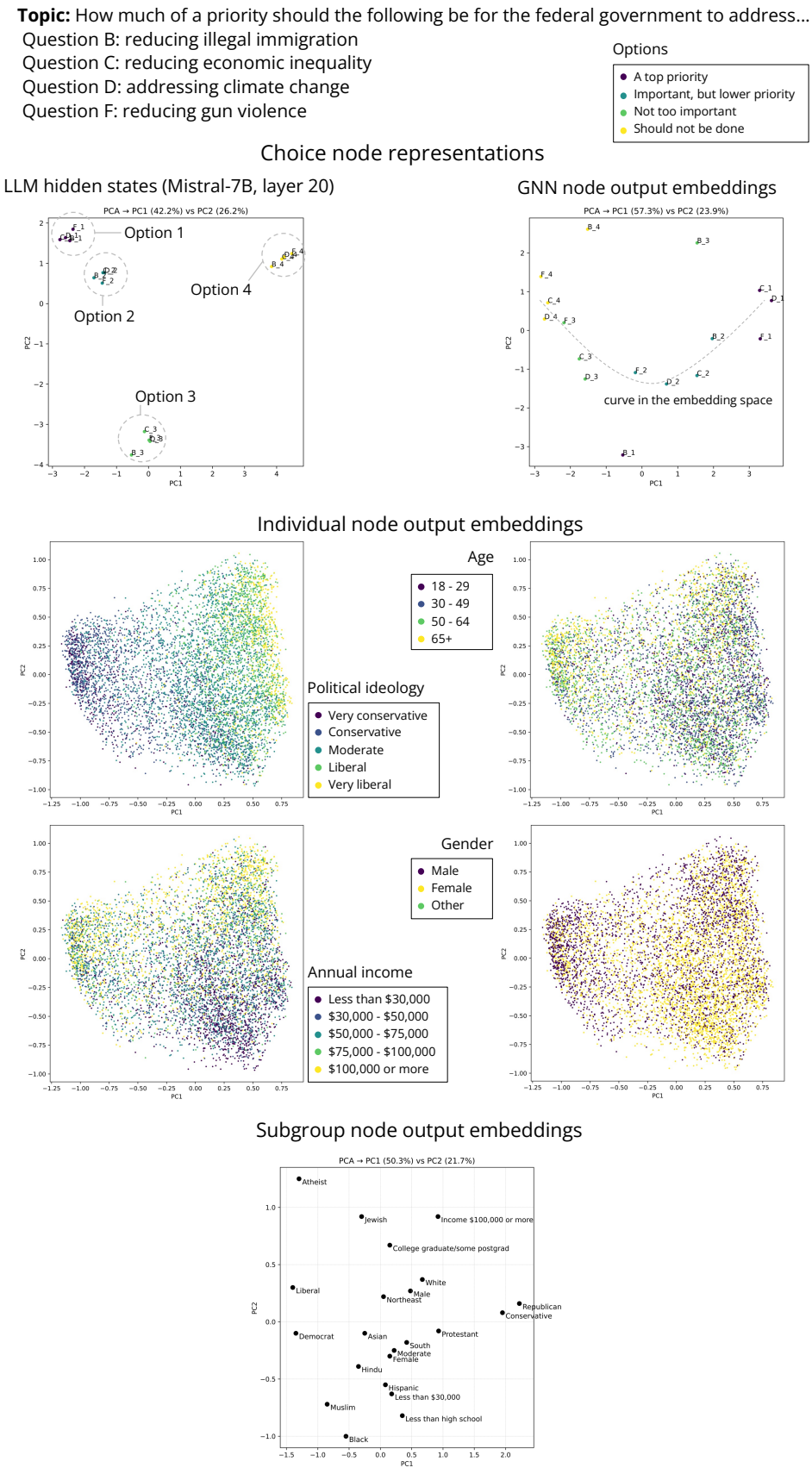


Figure 5: Visualization of LLM hidden states and GNN node embeddings on the first and second components of principal component analysis.

F Ablations

F.1 LLM baselines: LoRA fine-tuning with larger ranks, and full fine-tuning

We perform additional LLM fine-tuning baseline experiments (SFT, few-shot FT) and show that higher LoRA ranks or full fine-tuning do not necessarily improve performance over the main results (Tables 1 to 3) with LoRA rank 8. Table 10 shows the test prediction accuracy of Mistral-7B on DUNNING-KRUGER dataset for a single seed. Across all three settings, and in each setting across different numbers of few-shot examples (except for setting 2, new individuals, where past responses of new individuals are not available) our main result with LoRA $r = 8$ does not necessarily underperform fine-tuning runs with larger capacity.

F.2 Effect of hidden states across layers

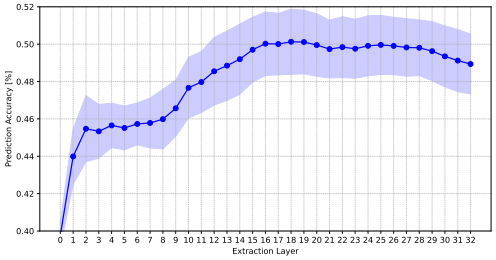


Figure 6: Mean and standard deviation of prediction accuracy on setting 3 (new questions) of OPINIONQA dataset when extracting hidden states from different layers of Mistral-7B-v0.1 (Table 3). Layer 0 is the post-embedding activation and layer 32 is the final pre-LM head activation.

Figure 6 shows GEMS accuracy on the OPINIONQA dataset with different layers of LLM (Mistral-7B-v0.1) to extract the hidden state from. In practice, we choose the layer that maximizes accuracy on validation questions, which turned out to be layer-18 for Mistral-7B-v0.1 on the OPINIONQA dataset. We note that the best transformer layer changes depending on the LLM and dataset, and has to be found through search with prediction accuracy on validation questions. Consistent with prior works on probing and interpretability (Kim et al., 2025; Tigges et al., 2023), middle-to-late layers generally provide the most semantically useful and transferable language representations.

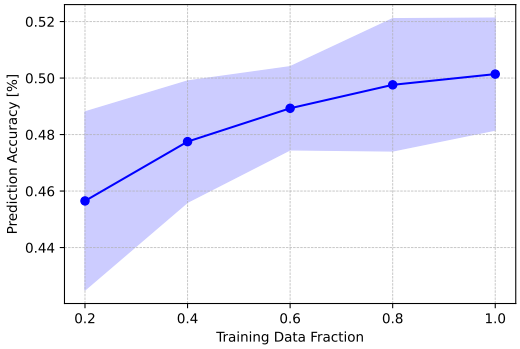


Figure 7: Mean and standard deviation of prediction accuracy on Setting 3 (new questions) of the OPINIONQA dataset using hidden states from layer-18 of Mistral-7B-v0.1. The x-axis denotes the fraction of choice nodes in the training graph used to fit the LLM-to-GNN projection in Equation (4). Accuracy improves as more paired examples are used, indicating that sufficient supervision is required to learn a map from LLM hidden states to the GNN output embedding space.

F.3 Effect of the number of LLM–GNN representation pairs

Learning the LLM-to-GNN representation projection with linear mapping requires paired examples of an LLM hidden state and its corresponding GNN output node embedding. Because this mapping lacks a linguistic prior, performance may degrade sharply when trained on too few pairs. We validate this with an ablation that fits the mapping using only 20%, 40%, 60%, and 80% of the available pairs (fractions taken over choice nodes $\mathcal{C}_{\text{train}}$ in Equation (4)) and evaluate on the new question setting. As shown in Figure 7, reducing the number of pairs leads to a rapid drop in accuracy, showing the sample size sensitivity of the mapping.

G Prompts to LLM

Zero-shot prompt: Provide an individual's features (demographics for OPINIONQA and TWIN-2K, pre-question survey responses for DUNNING-KRUGER dataset) in text form, followed by the question. Details about the list of individual features are defined in Section C. When an individual feature is missing (e.g., age is unknown), we omit it in the prompt rather than explicitly stating its absence (e.g., "Age: unknown"). Zero-shot prompts are used in zero-shot prompting, and supervised fine-tuning (SFT).

Few-shot prompt (with variable k in-context examples): Provide the individual's features,

Table 10: LLM fine-tuning performance with different trainable parameter size. Base LLM is Mistral-7B-v0.1, and fine-tuning performed on DUNNING-KRUGER dataset for all three settings. Setting 1, 2, 3 are predicting missing responses (imputation), responses of new individuals, and responses for new questions, respectively.

Setting	**Setting 1**			**Setting 2**	**Setting 3**		
Few-shots	0	3	8	-	0	3	8
LoRA, $r=8$	56.35	56.85	56.92	56.54	41.28	46.34	40.48
LoRA, $r=256$	56.35	56.61	57.38	56.68	42.75	46.27	42.33
Full fine-tuning	56.35	56.68	57.36	56.58	42.78	46.25	41.85

followed by k prior responses to related questions (see Section F for how we select related questions). Append the target question at the end. Few-shot prompts are used in few-shot prompting, and few-shot fine-tuning (few-shot FT).

Agentic CoT prompt: We directly adopt from (Park et al., 2024) with minimal modifications. The method consists of two stages. First, the individual's features and prior responses are given to an *expert reflection* module, which produces concise observations about the person's stances. Second, these observations, together with the individual's context, are passed to a prediction module that outputs an answer in the JSON format.

All examples presented in this section use synthetic profiles and responses, not real individuals, to protect privacy (Section 7). For fine-tuning, we apply cross-entropy loss to the single answer token immediately following the input prompt. We note that Qwen-3 (Yang et al., 2025) and GPT-OSS (OpenAI, 2025) use distinct response formats and detail the required tokenization and formatting in Section H.

Prompt Example: Zero-shot

System
Respond to the following question by choosing one of the available options, and strictly answering with the option letter (e.g., 'A', 'B', etc.). Do not provide any additional text or explanation.

User
Answer the following question as if your personal information is as follows:
Personal identification number: 12345.0
Age: 50-64
Race or ethnicity: White
Gender: Female
Education level: Some college, no degree
Income level: less than $30,000
Region of residence: West
Religion: Nothing in particular
Political party affiliation: Independent
Political ideology: Moderate

Question: Thinking about the nation's economy, how would you rate economic conditions in this country today?
A. Excellent
B. Good
C. Only fair
D. Poor

Answer:

Prompt Example: Few-shot ($k=2$)

System
Respond to the following question by choosing one of the available options, and strictly answering with the option letter (e.g., 'A', 'B', etc.). Do not provide any additional text or explanation.

User
Answer the following question as if your personal information is as follows:
Personal identification number: 12345.0
Age: 50-64
Race or ethnicity: White
Gender: Female
Education level: Some college, no degree
Income level: less than $30,000
Region of residence: West
Religion: Nothing in particular
Political party affiliation: Independent
Political ideology: Moderate

Question: How much, if at all, do you think the following proposals would do to reduce economic inequality in the U.S.? Expanding government benefits for the poor
A. A great deal

B. A fair amount
C. Not too much
D. Nothing at all
Answer:

Assistant
A. A great deal

User
Question: How much, if at all, do you think some people work harder than others contributes to economic inequality in this country?
A. Contributes a great deal
B. Contributes a fair amount
C. Contributes not too much
D. Contributes not at all
Answer:

Assistant
A. Contributes a great deal

User
Question: Thinking about the nation's economy, how would you rate economic conditions in this country today?
A. Excellent
B. Good
C. Only fair
D. Poor

Answer:

Prompt Example: Agentic CoT Reflection module ($k = 2$)

User
[Participant's information]
Age: 50-64
Race or ethnicity: White
Gender: Female
Education level: Some college, no degree
Income level: less than $30,000
Region of residence: West
Religion: Nothing in particular
Political party affiliation: Independent
Political ideology: Moderate

[Participant's prior responses]
Question: How much, if at all, do you think the following proposals would do to reduce economic inequality in the U.S.? Expanding government benefits for the poor
A. A great deal
B. A fair amount
C. Not too much
D. Nothing at all
Answer: A. A great deal

Question: How much, if at all, do you think some people work harder than others contributes to economic inequality in this country?
A. Contributes a great deal
B. Contributes a fair amount
C. Contributes not too much
D. Contributes not at all
Answer: A. Contributes a great deal

Imagine you are an expert social scientist (with a PhD) taking notes while observing this content. Write observations/reflections about the person's stances about key societal issues. (You should make more than 5 observations and fewer than 20. Choose the number that makes sense given the depth of the content above.)

Prompt Example: Agentic CoT Reflection module output (truncated)

Assistant
Observations & Reflections on the Participant's Societal Stances

1. **Political identity**
- Identifying as an *independent* with a *moderate* ideology indicates a willingness to cross party lines on key issues.
- Such a position often correlates with a preference for incremental, evidence-based reforms rather than ideological extremes.

2. **Attitude towards economic inequality**
...

Prompt Example: Agentic CoT - 2. Prediction module

User
[Participant's information]
Age: 50-64
Race or ethnicity: White

...

[Participant's prior responses]
Question: How much, if at all, do you think the following proposals would do to reduce economic inequality in the U.S.? Expanding government benefits for the poor
...

[Expert social scientist's observations/reflections]
(**Generated observations/reflections from the expert from step 1**)

=====

What you see above is a participant information. Based on the information, I want you to predict the participant's survey responses. All questions are multiple choice where you must guess from one of the options presented. As you answer, I want you to take the following steps:
Step 1) Describe in a few sentences the kind of person that would choose each of the response options. ("Option Interpretation")
Step 2) For each response option, reason about why the Participant might answer with the particular option. ("Option Choice")
Step 3) Write a few sentences reasoning on which of the option best predicts the participant's response ("Reasoning")
Step 4) Predict how the participant will actually respond in the survey. Predict based on the information and your thoughts, but ultimately, DON'T overthink it. Use your system 1 (fast, intuitive) thinking. ("Response")

Here is the question:

=====

Question: Thinking about the nation's economy, how would you rate economic conditions in this country today?
A. Excellent
B. Good
C. Only fair
D. Poor

=====

Output format - output your response in json, where you provide the following:

{"Response": "<your predicted response option letter>"}

H Tokenization: Qwen3 and GPT-OSS

In this section, we outline the differences in input preprocessing for GPT-OSS (OpenAI, 2025) and Qwen-3 (Yang et al., 2025), which arise from their distinct response formats.

GPT-OSS. GPT-OSS employs the Harmony response format to support advanced context engineering. Each generation typically begins with an analysis channel '<|channel|>analysis', where the model produces an internal chain-of-thought not exposed to users, and concludes with a final channel '<|start|>assistant<|channel|>final<|message|>', which contains the user-facing response.

During baseline experiments before fine-tuning, to measure the model's existing predictive capability, we place no constraints on generation: the model is free to produce both analysis and final content, and we parse the output from the final channel to evaluate accuracy.

During fine-tuning, however, we constrain the output to directly generate the answer in the final channel. This step improves predictive accuracy while avoiding social bias that could result from fine-tuning on model-generated chain-of-thoughts, which may yield correct answers through ungrounded reasoning about individuals. In this setup, we append the channel header explicitly to indicate the model that final answer should be generated, and apply next-token prediction loss to the final answer token. An example training input prompt is shown below:

```
<|start|>developer<|message|># Instructions

Respond to the following
     question by choosing one of the available
     options, and strictly answering with the
     option letter (e.g., 'A', 'B', etc.). Do not
     provide any additional text or explanation.

<|end|><|start|>user
    <|message|>Answer the following question as
     if your personal information is as follows:

Personal identification number: 12345.0
Age: 50-64
Race or ethnicity: White
```

```
Gender: Female
Education level: Some college, no degree
Income level: less than $30,000
Region of residence: West
Religion: Nothing in particular
Political party affiliation: Independent
Political ideology: Moderate

Question: Would you
     say the following was a reason or was not a
     reason why  there were guns in your household
     when you were growing up? For protection

A. Yes, was a reason
B. No, was not a reason

Answer:<|end|><|
    start|>assistant<|channel|>final<|message|>
```

As shown on the example, the tokenization step involves appending special tokens indicating the final channel. Given the input prompt, the model generates probability distribution over available options in the next token. Cross entropy loss is applied at that token position to fine-tune the model.

Qwen-3. Similarly, Qwen-3 introduces a thinking mode designed to let the model do more step-by-step reasoning (chain-of-thought) before generating a final answer. During baseline experiments before fine-tuning, we place no constraints on generation and this allows the model to perform thinking (wrapped by `<think>...</think>`). During fine-tuning, we constrain the output to directly generate the answer by appending the empty thinking (`<think>\n\n</think>`) explicitly to indicate the model for direct answer generation.