

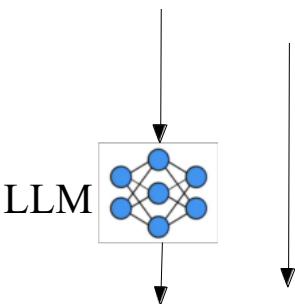
Lecture 8: LLM Fine Tuning

Instructor: Zhanzhan Zhao

Pre-training



Once upon a
midnight dreary
while I pondered.



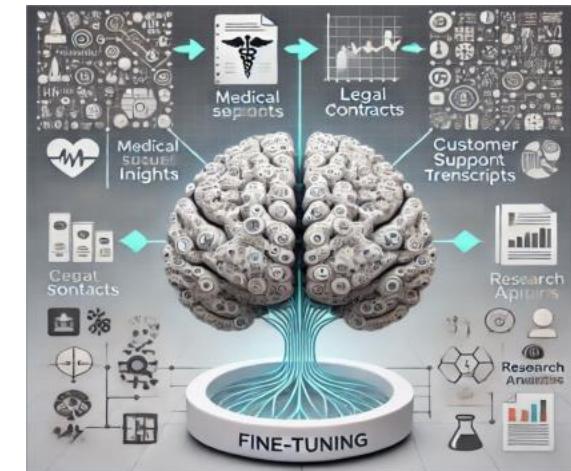
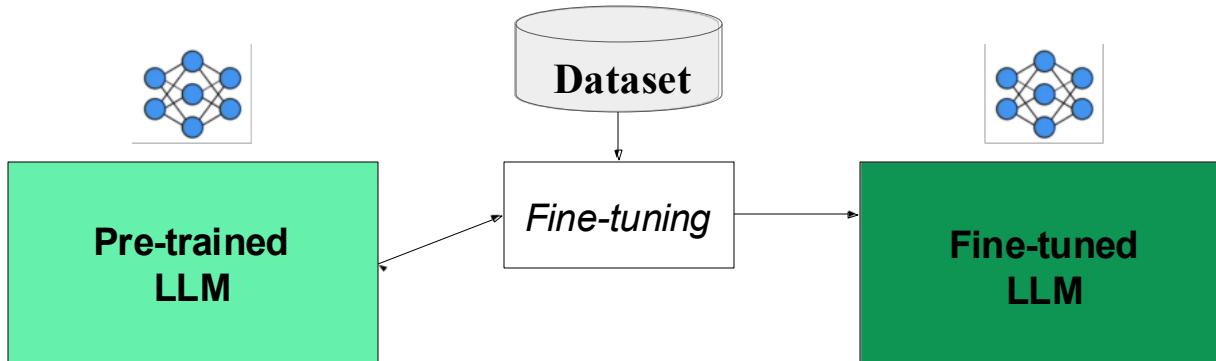
- Model at the start
 - Zero knowledge about the world
 - Can't form English words
- **Task:** Next token prediction
- **Data:** Giant corpus of text data from the web: “unlabeled”
- **Training:** Self-supervised learning

- After training
 - Learns language
 - Learns Knowledge



What is Fine-tuning?

- Fine-tuning is taking a **pre-trained** (general purpose) model and train some of its weights.
 - A general-purpose base model → **specialized model** for a particular use case.
- **Fine-tuning vs. prompt engineering:**
 - Gets the model to learn the data (**adjusting model's weights**), rather than just get access to it.
 - Let you put more data into the model than what fits in the prompt.
- **Fine-tuning vs. pre-training:**
 - Requires significantly less data and computational resources.



Base model vs. Fine-tuned model Completions – Example I

Base model: GPT-3

Prompt: How to fine-tune a model?

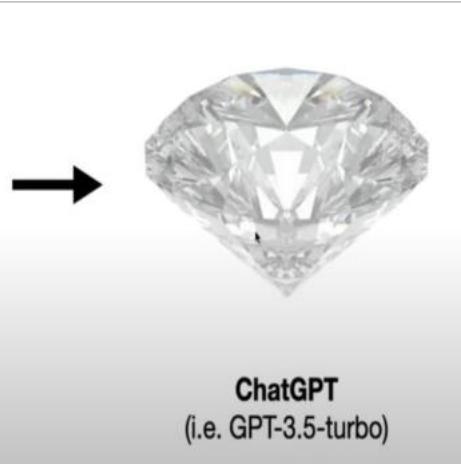
Completion:

How can I control the complexity of a model?

How do I know when my model is done?

How do I test a model?

.



Fine-tuned model: GPT-3.5-turbo (ChatGPT)

Prompt: How to fine-tune a model?

Completion:

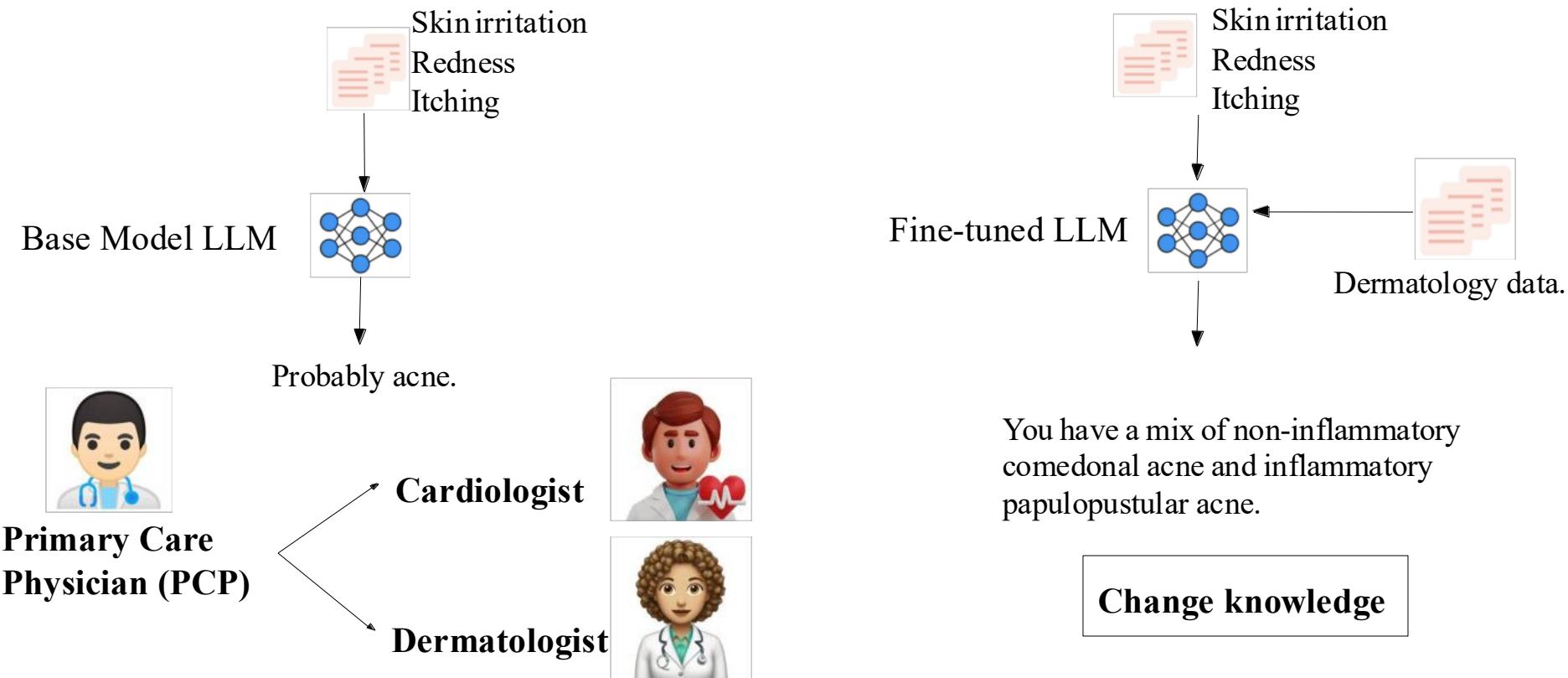
Fine-tuning a model typically involves taking a pre-trained model and further training it on a new dataset specific to your task. Here are the general steps to fine-tune a model:

- 1. Choose a pre-trained model:** Start by selecting a pre-trained model that is relevant to your task..
- 2. Prepare your new dataset:** Collect or create a dataset that is specific to your task.
- 3. Modify the top layers:** Remove the final layers of the pre-trained model.
- 4. Fine-tune the model:** Train the modified model on your new dataset.
- 5. Evaluate the model:** Once training is complete, evaluate the fine-tuned model on the validation set to assess its performance.

Change behavior (instruction-based):

More desirable, practical and helpful completions!

Base model vs. Fine-tuned model Completions – Example II



Why to Fine-tune?

A smaller (fine-tuned) model can often outperform larger (more expensive) models on the set of tasks on which it was fine-tuned!



>



Instruct GPT (1.3B)

GPT-3 (175B)

OpenAI paper: "Training language models to follow instructions with human feedback"

Prompt Engineering vs. Fine-tuning

Prompting

- + No data (very few) to get started
- + Smaller upfront cost
- + No technical knowledge needed

- Much less data fits
- Forgets data
- Hallucinations

Generic, side projects, prototypes

Fine-tuning

- More high-quality data
- Upfront compute cost
- Needs some technical knowledge

- + Nearly unlimited data fits
- + Learn from your data (user's behavioral signals)
- + Less cost afterwards if smaller model

Domain-specific, enterprise, production usage

Retrieval-Augmented Generation (RAG) vs. Fine-tuning

RAG

- + Better in integrating new (dynamic) knowledge
- + No training, no retraining
- + More accurate responses – reducing hallucinations

- Slower: two-step process
- Lower performance in specific tasks with high-quality data
- Not suitable for changing the behavior of the responses

Fine-tuning

- Not suitable for learning evolving knowledge
- More expensive (high-quality data, re-training)
- Still prone to hallucinations

- + Optimizes LLM performance for specific tasks
- + Precise control over the training data
 - o Adjusting tone / style of a language
 - o Aligning with user preferences
 - o Controlling output format
- + Improve efficiency (latency & cost)

Generic, context-heavy tasks, dynamic knowledge injection

Improve performance for specific task

Benefits of Fine-tuning your own LLM

1

Performance

- Decrease Hallucinations
- Increase consistency

2

Privacy

- On premise
- Prevent leakage



3

Cost

- Lower cost per request
- Increased transparency
- Greater control

4

Reliability

- Control uptime
- Lower latency
- Moderation

3 Ways to Fine-tune

1) Self-supervised



Houston, we have a
↓
LLM
↓
problem.

2) Supervised

Input	Output

Input: Who was the first President of the USA?
Output: George Washington

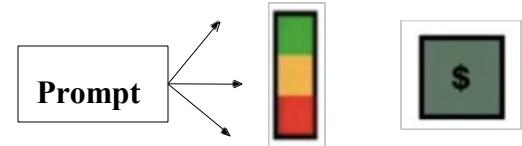
"""\Please answer the following question.
Q: {Question}
A: {Answer}"""

3) Reinforcement Learning

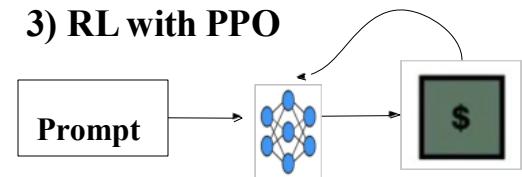
1) Supervised Fine-tuning



2) Train Reward model



3) RL with PPO



Supervised finetuning (SFT)

- Idea: finetune the LLM with language modeling of the desired answers
 - Next word prediction
 - “supervised”
- How do we collect the data? Ask humans

Can you write a short introduction about the relevance of the term "monopsony" in economics? Please use examples related to potential monopsonies in the labour market and cite relevant research.

"Monopsony" refers to a market structure where there is only one buyer for a particular good or service. In economics, this term is particularly relevant in the labor market, where a monopsony employer has significant power over the wages and working conditions of their employees [...]. Overall, the concept of monopsony is essential to understanding the dynamics of labor markets and the impact of market power on workers. Further research is needed to understand the extent and impact of monopsonies on the economy and to develop policies to address this issue. References: Bivens, J., & Mishel, L. (2013). The Pay of Corporate Executives and Financial Professionals as Evidence of Rents in Top 1 Percent Incomes. *Journal of Economic Perspectives*, 27(3), 57-78.

My kid needs to do a science project for elementary school, what are some cheap but fun projects to do?

There are many cheap and fun science projects that elementary school kids can do! Here are a few ideas: 1. Make a homemade lava lamp: Fill a clear plastic bottle with water and vegetable oil, add a few drops of food coloring, and then drop in an Alka-Seltzer tablet. The reaction will cause bubbles to rise and fall like a lava lamp. [...] 7. Create a solar oven: Using a pizza box, aluminum foil, and plastic wrap, create a solar oven that can cook s'mores or other small snacks. 8. Explore magnets: Using magnets and a variety of objects, have your child experiment with magnetism to see what objects are attracted to magnets and which ones are not. These are just a few ideas to get you started! There are many more cheap and fun science projects that your child can do with household items and a little creativity.

OpenAssistant
[Kopf+ 2023]

This was the ~key to GPT3 -> ChatGPT model!

Scalable data for SFT: eg Alpaca

- Problem: human data is slow to collect and expensive
- Idea: use LLMs to scale data collection

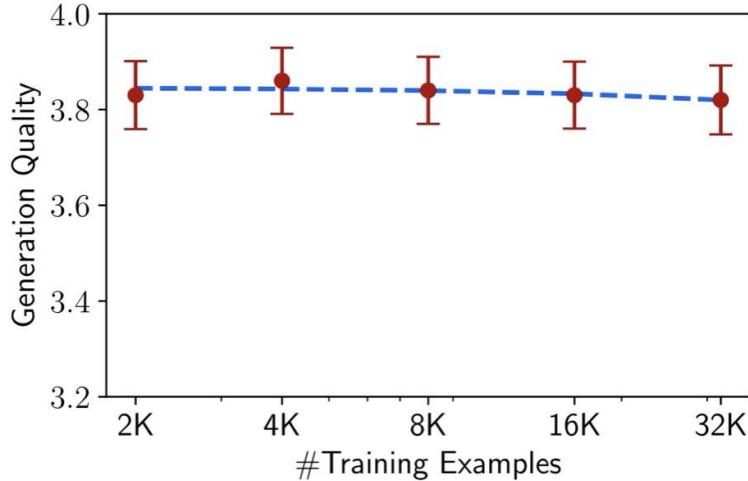
Give three tips for staying healthy.	1. Eat a balanced diet and make sure to include plenty of fruits and vegetables. 2. Exercise regularly to keep your body active and strong. 3. Get enough sleep and maintain a consistent sleep schedule.
What does “algorithm” mean?	An algorithm is a step-by-step set of instructions used to solve a problem or achieve a goal. It is a process of breaking down complex tasks into smaller, more manageable steps. Algorithms can be used to find the shortest path from one place to another, sort data, or render data from an immense database in a few seconds.
Find the average number in a list.	<pre>def avg_list(nums): total = 0 for n in nums: total += n return total / len(nums) print(avg_list([4, 3, 6, 10, 8])) # Output 6.4</pre>

Alpaca
[Taori+ 2023]

Started for academic replication of ChatGPT but “synthetic data generation” is now hot topic!

Scalable data for SFT: quantity?

- You need very little data for SFT! ~few thousand



LIMA
[Zhou+ 2023]

- Just learns the format of desired answers (length, bullet points, ...)
 - The knowledge is already in the pretrained LLM!
 - Specializes to one “type of user”

RL from Human Feedback (RLHF)

- Problem: SFT is **behavior cloning** of humans
 1. **Bound by human abilities:** humans may prefer things that they are not able to generate
 2. **Hallucination:** cloning *correct* answer teaches LLM to hallucinate if it didn't know about it!

Can you write a short introduction about the relevance of the term "monopsony" in economics? Please use examples related to potential monopsonies in the labour market and cite relevant research.

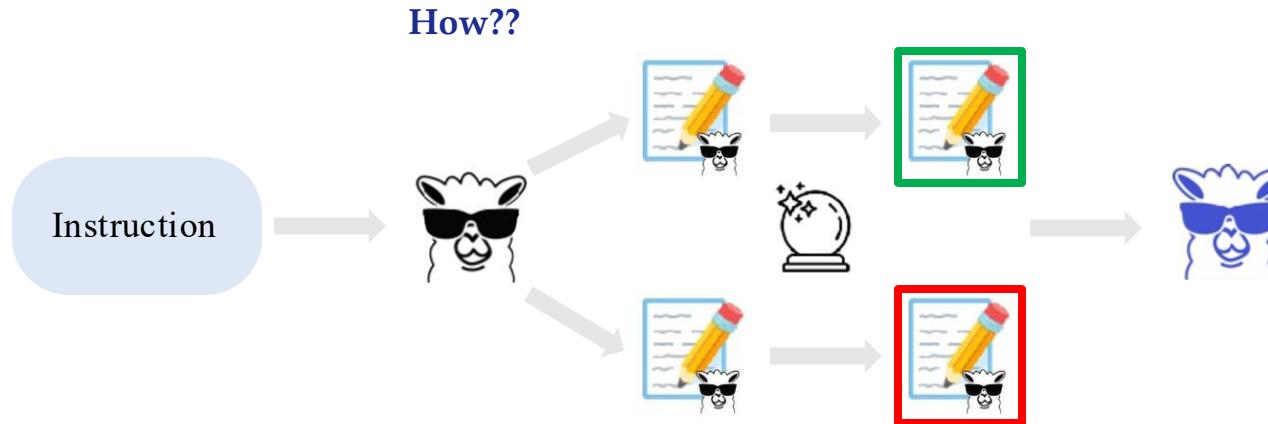
"Monopsony" refers to a market structure where there is only one buyer for a particular good or service. In economics, this term is particularly relevant in the labor market, where a monopsony employer has significant power over the wages and working conditions of their employees. [...]. Overall, the concept of monopsony is essential to understanding the dynamics of labor markets and the impact of market power on workers. Further research is needed to understand the extent and impact of monopsonies on the economy and to develop policies to address this issue. **References:** Bivens, J., & Mishel, L. (2013). *The Pay of Corporate Executives and Financial Professionals as Evidence of Rents in Top 1 Percent Incomes*. *Journal of Economic Perspectives*, 27(3), 57-78.

If LLM doesn't know [Bivens 2013] \Rightarrow teaches the model to make up plausibly sounding references

3. **Price:** collecting ideal answers is expensive

RLHF

- Idea: maximize human preference rather than clone their behavior
- Pipeline:
 1. For each instruction: generate 2 answers from a pretty good model (SFT)
 2. Ask labelers to select their preferred answers
 3. Finetune the model to generate more preferred answers



RLHF: PPO

- Idea: use reinforcement learning
 - What is the reward?
 - Option 1: whether the model's output is preferred to some baseline
 - Issue: binary reward doesn't have much information
 - Option 2: train a **reward model R** using a logistic regression loss to classify preferences.
- $$p(i > j) = \frac{\exp(R(x, \hat{y}_i))}{\exp(R(x, \hat{y}_i)) + \exp(R(x, \hat{y}_j))} \quad [\text{Bradley-Terry 1952}]$$
 - Use logits R(...) as reward \Rightarrow continuous information \Rightarrow information heavy!
- Optimize $\mathbb{E}_{\hat{y} \sim p_\theta(\hat{y}|x)} \left[R(x, \hat{y}) - \beta \log \frac{p_\theta(\hat{y}|x)}{p_{ref}(\hat{y}|x)} \right]$ using PPO
 \rightarrow regularization avoids overoptimization
 - Note: LMs are policies not a model of some distribution

RLHF: PPO -> ChatGPT

Step 1

Collect demonstration data, and train a supervised policy.

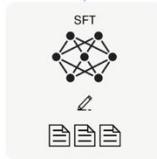
A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



This data is used to fine-tune GPT-3 with supervised learning.



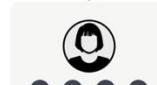
Step 2

Collect comparison data, and train a reward model.

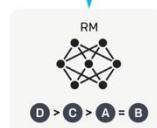
A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.



This data is used to train our reward model.



Step 3

Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.



The policy generates an output.



Once upon a time...



The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.

r_k

RLHF
[Ouyang+ 2022]

RLHF: PPO challenges

- Problem: RL in theory simple, in practice **messy** (clipping, rollouts, outer loops,...)

```
def rollout(self, queries_data) -> Dict[str, Tensor]:
    """Rollout trajectories with policy.

    Args:
        queries_data: Sequence of batches or DataLoader.
            Each batch is a dict with keys 'queries' and 'query_attn_masks'.
    Returns:
        Dictionary with keys
            'queries', 'query_attn_masks', 'responses',
            'logprobs', 'ref_logprobs', 'values',
            'rewards', 'non_score_rewards', 'shaped_rewards'.
    """
    # Give up dropout throughout.
    self.policy.eval()
    self._make_sdhp_happy()
    # `keep_fp32_wrapper` retains the autocast wrapper of model.forward created by accelerate:
    # recall one sets mixed precision options with accelerator.
    # The precise value of this arg doesn't matter here, since we use the unwrapped model only for respond.
    # Generally, try to use the wrapped model as much as you can, since it's got the autocast/cast-back wrappers.
    unwrapped_policy = self.accelerator.unwrap_model(self.policy, keep_fp32_wrapper=True)

    self.ref_policy.eval()
    self.reward_model.eval()

    rollouts = []
    for batch_idx, batch in tqdm.tqdm(
        enumerate(queries_data),
        disable=not self.accelerator.is_main_process,
        desc="rollout",
    ):
        # Sample rollouts.
        queries, query_attn_masks = common.unpack_dict(
            common.prepare_inputs(batch, device=self.accelerator.device),
            keys=["queries", "query_attn_masks"],
        )
        respond_outputs = unwrapped_policy.respond(queries, query_attn_masks, temperature=self.args.temperature)
        respond_outputs = common.unpack_dict(respond_outputs, ("responses",))

        # Evaluate logprobs of the samples.
        rollouts_batch = {"queries": queries, "query_attn_masks": query_attn_masks, "responses": responses}
        policy_outputs = self.policy**rollouts_batch, temperature=self.args.temperature
        ref_policy_outputs = self.ref_policy**rollouts_batch, temperature=self.args.temperature
        policy_outputs = common.unpack_dict(
            policy_outputs, keys=("logprobs", "values", "entropies"), return_type=dict
        )
        ref_policy_outputs = common.unpack_dict(
            ref_policy_outputs, keys=("logprobs", "entropies"), return_type=dict
        )
        rollouts_batch.update(policy_outputs)
        rollouts_batch.update({"ref_(key)": value for key, value in ref_policy_outputs.items()})

    # Evaluate reward of the samples.
    text_queries, text_responses = tuple(
        self.tokenizer.batch_decode(tensor, skip_special_tokens=True, clean_up_tokenization_spaces=True)
        for tensor in (queries, responses)
    )
    del queries, responses # Prevent mistakes.

    # We retokenizer, since policy and reward model might not have the same tokenizer.
    # TODO(lxuechen): Avoid retokenization when policy and reward tokenizer are the same.
    text_sequences = [q + r for q, r in utils.zip(text_queries, text_responses)]
    # TODO(lxuechen): This response retokenization has issues with OPT, since the tokenizer always prepends
    # <bos_token>. But the issue is local to post_reward, which isn't an issue if we don't penalize.
    sequences, responses = tuple(
        self.tokenizer(text, return_tensors="pt", padding=True, truncation=True)
        for text in (text_sequences, text_responses)
    )
    sequences, responses = common.prepare_inputs((sequences, responses), device=self.accelerator.device)

    reward_outputs = self.reward_model(**sequences)
    reward_outputs = self.post_reward(reward_outputs, responses.input_ids)
    rollouts_batch.update(reward_outputs)

    # Shape reward with KL penalty.
    shape_reward_outputs = self.shape_reward(
        rewards=rollouts_batch["rewards"],
        responses=rollouts_batch["responses"],
        logprobs=rollouts_batch["logprobs"],
        ref_logprobs=rollouts_batch["ref_logprobs"],
    )
    rollouts_batch.update(shape_reward_outputs)

    rollouts_batch_cpu = {key: value.cpu() for key, value in rollouts_batch.items()}
    rollouts.append(rollouts_batch_cpu)

    # Items in dict need to be of same shape.
    rollouts = common.merge_dict(rollouts, merge_fn=torch.cat)
    # Estimating advantages outside the loop gives more samples for reward normalization.
    advantages = self._estimate_advantage(
        rewards=rollouts["shaped_rewards"].to(self.accelerator.device),
        values=rollouts["values"].to(self.accelerator.device),
    )
    advantages = {key: value.cpu() for key, value in advantages.items()}

    return {**rollouts, **advantages}
```

AlpacaFarm
[Dubois+ 2023]

Rollout

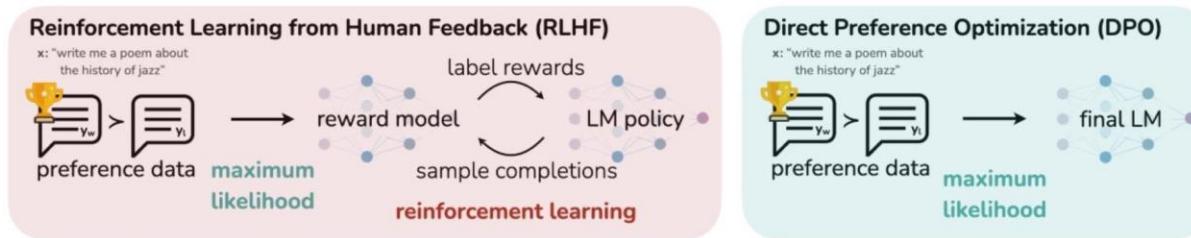
RLHF: DPO

- Idea: maximize probability of preferred output, minimize the other

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w | x)}{\pi_{\text{ref}}(y_w | x)} - \beta \log \frac{\pi_\theta(y_l | x)}{\pi_{\text{ref}}(y_l | x)} \right) \right]$$

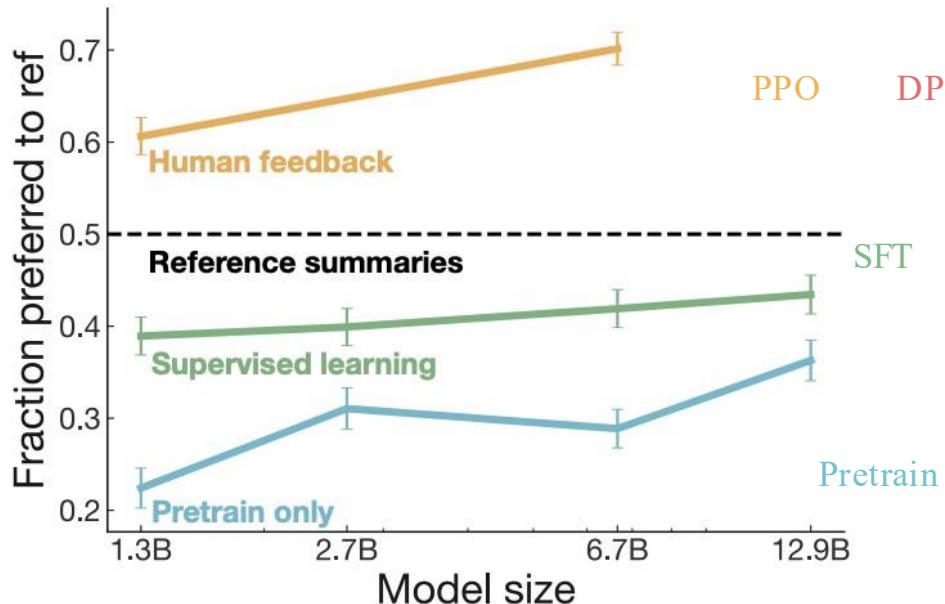
DPO

[Rafailov+ 2023]



- This is ~equivalent (same global minima) to RLHF/PPO
- Much simpler than PPO and performs as well \Rightarrow standard (in open source community)

RLHF: gains



Method	Simulated Win-rate (%)
GPT-4* [†]	79.0 ± 1.4
ChatGPT* [†]	61.4 ± 1.7
PPO	46.8 ± 1.8
DPO	46.8 ± 1.7
Best-of-1024	45.0 ± 1.7
Expert Iteration	41.9 ± 1.7
SFT 52k	39.2 ± 1.7
SFT 10k	36.7 ± 1.7
Binary FeedME	36.6 ± 1.7
Quark	35.6 ± 1.7
Binary Reward Conditioning	32.4 ± 1.6
Davinci001*	24.4 ± 1.5
LLaMA 7B*	11.3 ± 1.1

Learn to summarize
[Stiennon+ 2020]

AlpacaFarm
[Dubois+ 2023]

RLHF: human data

- Data: human crowdsourcing

In this task, you will be provided with a **Prompt** from a user (e.g., a question, instruction, statement) to an AI chatbot along with two potential machine-generated **Responses** to the Prompt. Your job is to assess which of the two Responses is better for the Prompt, considering the following for each Response:

<p>Helpfulness: To what extent does the Response provide useful information or satisfying content for the Prompt?</p> <p>Responses should:</p> <ul style="list-style-type: none">▪ Address the intent of the user's Prompt such that a user would not feel the Prompt was ignored or misinterpreted by the Response.▪ Provide specific, comprehensive, and up-to-date information for the user needs expressed in the Prompt.▪ Be sensible and coherent. The response should not contain any nonsensical information or contradict itself across sentences (e.g., refer to two different people with the same name as if they are the same person).▪ Adhere to any requirements indicated in the Prompt such as an explicitly specified word length, tone, format, or information that the Response should include.▪ Not contain inaccurate, deceptive, or misleading information (based on your current knowledge or quick web search - you do not need to perform a rigorous fact check)▪ Not contain harmful, offensive, or overly sexual content	<p>Rating scale:</p> <ul style="list-style-type: none">▪ Not at All Helpful: Response is useless/irrelevant, contains even a single piece of nonsensical/inaccurate/deceptive/misleading information, and/or contains harmful/offensive/overly sexual content.▪ Slightly Helpful: Response is somewhat related to the Prompt, does not address important aspects of the Prompt, and/or contains outdated information.▪ Somewhat Helpful: Response partially addresses the intent of the Prompt (responses may be more informative), contains extra unhelpful information, and/or is lacking helpful details/specifications.▪ Very Helpful: Response addresses the intent of the Prompt with a satisfying response. Some users might want a more comprehensive response with additional details or context. It is comparable to a response an average human with basic subject-matter knowledge might provide.▪ Extremely Helpful: Response completely addresses the intent of the Prompt and provides helpful details/context. It is comparable to a response a talented/well-informed human with subject-matter expertise might provide.
<p>Presentation: To what extent is the content of the Response conveyed well?</p> <p>Responses should:</p> <ul style="list-style-type: none">▪ Be organized in a structure that is easy to consume and understand. Presenting in a logical order and makes good use of formatting such paragraphs, lists, or tables.▪ Be clearly written in a polite neutral tone that is engaging, direct, and inclusive. The tone should not be <i>overly</i> friendly, salesy, academic, sassy, or judgmental in a way that most users would consider to be off-putting or overdone.▪ Have consistent style with natural phrasing and transitions as if composed by a single talented human.▪ Not be repetitive. Similar information should not be repeated multiple times. It is harder for users to consume the helpful information in a response if there is repetitive or less helpful information mixed into the response.▪ Not include notable language issues or grammatical errors	<p>Rating scale:</p> <ul style="list-style-type: none">▪ Poor: Response is poorly written or has notable structural, formatting, language, or grammar issues. Or Response has an awkward or inappropriate tone. Or the Response repeats similar information. Or only a small portion of the Response contains helpful information.▪ Adequate: Response could have been written/organized better or may have minor language/grammar issues. A minimal amount of less helpful information may be present. Users would still feel the content of the Response was easy to consume.▪ Excellent: Response is very well written and organized. Sentences flow in a logical order with smooth transitions and consistent style. The content of the Response is conveyed in a way that is comparable to a response a talented human might produce.

Overall, you should consider both factors in your SxS rating of which response is better. A more concise response presenting the most helpful information directly and clearly is usually better than a longer response that may be harder to consume and/or contains clearly off-topic information. Responses with Poor Presentation (e.g., rambling, inappropriate tone) should play a significant role in your assessment of which side is better. It may help to imagine the user chatting with a real person and consider which Response most users would prefer to receive from a real person.

► Annotation Guidelines

Instruction

Tell me about self driving cars

example

Input

AI Response 1

Self driving cars are vehicles that are capable of detecting their surroundings and navigating without any input from a human driver. They use a variety of sensors, such as cameras, radar, and lidar, to detect objects in their environment and use advanced algorithms to make decisions about how to maneuver around them. Self driving cars are becoming increasingly popular and are being used in a variety of applications, such as ride-hailing services and autonomous delivery services.

AI Response 2

Self driving cars are cars that are equipped with sensors, cameras, and software, which allows them to navigate without the need for a driver. Self driving cars are able to monitor their environment and navigate without the assistance of a driver. Self driving cars use a combination of technologies such as GPS, radar, lidar, and computer vision to detect their environment and avoid obstacles. They are also able to update their maps and adjust their routes to account for variables such as traffic and construction.

The main benefit of self driving cars is their ability to reduce crashes and save lives. Autonomous vehicles have the potential to save up to 30,000 lives annually. They are also more energy efficient than traditional vehicles, as they do not require a driver who has to maintain a minimum speed. Autonomous vehicles also create new opportunities for increased mobility, allowing those who are unable to drive to get around.

Rating

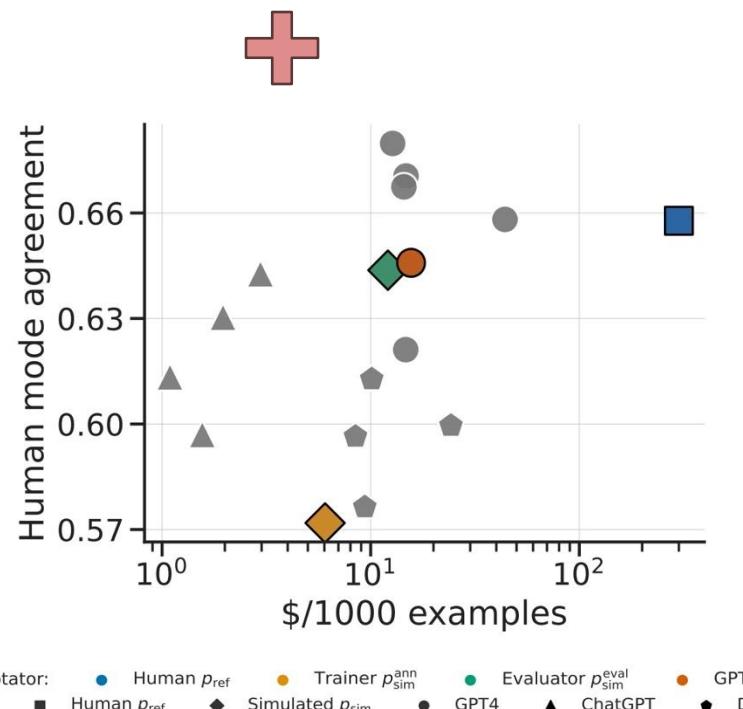
- Response 1 is better.
- Response 1 is only slightly better. (only pick this if it's truly close)
- Response 2 is only slightly better. (only pick this if it's truly close)
- Response 2 is better.

RLHF: challenges of human data

- Slow & expensive
- Hard to focus on correctness rather than form (eg length)
- Annotator distribution shifts its behavior
- Crowdsourcing ethics

RLHF: LLM data

- Idea: replace human preferences with LLM preferences



Works surprisingly well!

=> Standard in open community

AlpacaFarm
[Dubois+ 2023]

Overview

Pretraining -> GPT3

- Task & loss
- Evaluation
- Data
- Scaling laws

Post-training -> ChatGPT

- Task
- SFT: data & loss
- RLHF : data & loss
- Evaluation



Evaluation: aligned LLM

- How do we evaluate something like ChatGPT?
- Challenges:
 - Can't use validation loss to compare different methods
 - Can't use perplexity: not calibrated
 - Large diversity
 - Open-ended tasks => hard to automate
- Idea: ask for annotator preference between answers

Some aligned
LLMs are policies!

Table 1: Distribution of use case categories from our API prompt dataset.

Use-case	(%)
Generation	45.6%
Open QA	12.4%
Brainstorming	11.2%
Chat	8.4%
Rewrite	6.6%
Summarization	4.2%
Classification	3.5%
Other	3.5%
Closed QA	2.6%
Extract	1.9%

InstructGPT
[Ouyang+ 2022]

Human evaluation: eg ChatBot Arena

- Idea: have users interact (blinded) with two chatbots, rate which is better.

The screenshot shows the homepage of the Chatbot Arena. At the top, there are navigation links: Arena (battle), Arena (side-by-side), Direct Chat, Vision Direct Chat, Leaderboard, and About Us. Below the navigation is the title "Chatbot Arena: Benchmarking LLMs in the Wild". Underneath the title are links to Blog, GitHub, Paper, Dataset, Twitter, and Discord. A "Rules" section contains three bullet points: "Ask any question to two anonymous models (e.g., ChatGPT, Claude, Llama) and vote for the better one!", "You can continue chatting until you identify a winner.", and "Vote won't be counted if model identity is revealed during conversation.". Below the rules is the "Arena Elo Leaderboard" section, which states: "We collect 300K+ human votes to compute an Elo-based LLM leaderboard. Find out who is the LLM Champion!". There is a "Chat now!" button with a hand icon. At the bottom, there are buttons for "Model A" and "Model B". A note says "Expand to see the descriptions of 31 models".

- Problem: cost & speed!



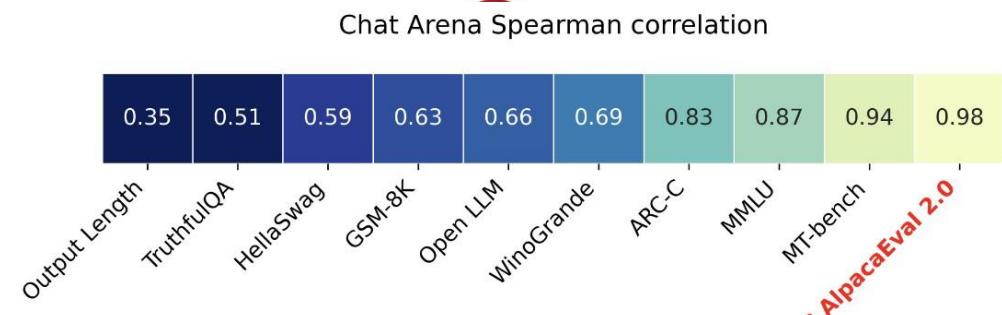
ChatBot Arena
[Chiang+ 2024]

LLM evaluation: eg AlpacaEval

- Idea: use LLM instead of human

- Steps:

- For each instruction: generate output by 1
- Ask GPT-4 which output is better
- Benefits:
 - Average win-probability \Rightarrow win rate
 - 58% correlation with ChatBot Arena
 - <3 min and <\$10
- Challenge: **spurious correlation**



AlpacaEval
[Li+ 2023]

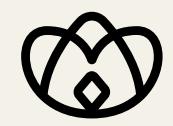
LLM evaluation: spurious correlation

- e.g. LLM prefers longer outputs
- Possible solution: regression analysis / causal inference to “control” length

		AlpacaEval			Length-controlled AlpacaEval		
		concise	standard	verbose	concise	standard	verbose
	gpt4_1106_preview	22.9	50.0	64.3	41.9	50.0	51.6
	Mixtral-8x7B-Instruct-v0.1	13.7	18.3	24.6	23.0	23.7	23.2
	gpt4_0613	9.4	15.8	23.2	21.6	30.2	33.8
	claude-2.1	9.2	15.7	24.4	18.2	25.3	30.3
	gpt-3.5-turbo-1106	7.4	9.2	12.8	15.8	19.3	22.0
	alpaca-7b	2.0	2.6	2.9	4.5	5.9	6.8

AlpacaEval LC
[Dubois+ 2023]

Paper Sharing -1



CUHK-SZ
Computational
Linguistics

How does generative AI affect creativity —— The dilemma of individual improvement and collective convergence

From: 《Generative AI enhances individual creativity but reduces the collective diversity of novel content》

Anil R. Doshi and Oliver P. Hauser*

Presenter : Wei Linwen

March 20, 2025

One Sentence Summary

This paper investigates the impact of **generative AI**, such as ChatGPT, on **human creativity**. Research has found that using AI generated story inspiration can make individuals write more creative and engaging stories, especially for those who are not originally very creative. However, stories generated using AI will become more similar, leading to a decrease in overall story diversity.

Simply put, AI helps individuals become more "creative," but the creative content of society as a whole becomes more "uniform. This is like a social dilemma: individuals benefit, but the collective loses diversity.

Background

The popularity of generative AI, such as ChatGPT, has raised questions about the uniqueness of human creativity.

Core contradiction: Can AI assistance truly enhance **creativity**? Will it still lead to **content homogenization**?

Research Question

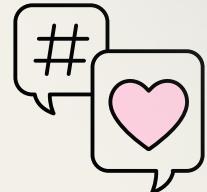
The **personal** level:

Can AI enhance individual creativity?

The **collective** level:

Does AI assisted creation lead to a decrease in content diversity?

Design of Experiments



Assessment Dimensions:

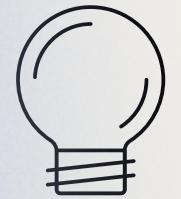
Creativity: Novelty and Usefulness.

Emotional characteristics: fun, writing, plot twists, etc.

Similarity Analysis: Quantify the similarity between stories through text embedding.



Participants : 293 writers + 600 evaluators.



Three groups of controls:

1. Pure human group (no AI assistance)
 2. Human + 1 AI Inspiration (GPT-4 can be called up 1 time to generate story inspiration)
 3. Humans + 5 AI Inspirations (GPT-4 can be invoked 5 times to generate inspirations)
- Task :** Write an 8-sentence short story (the theme is randomly assigned).



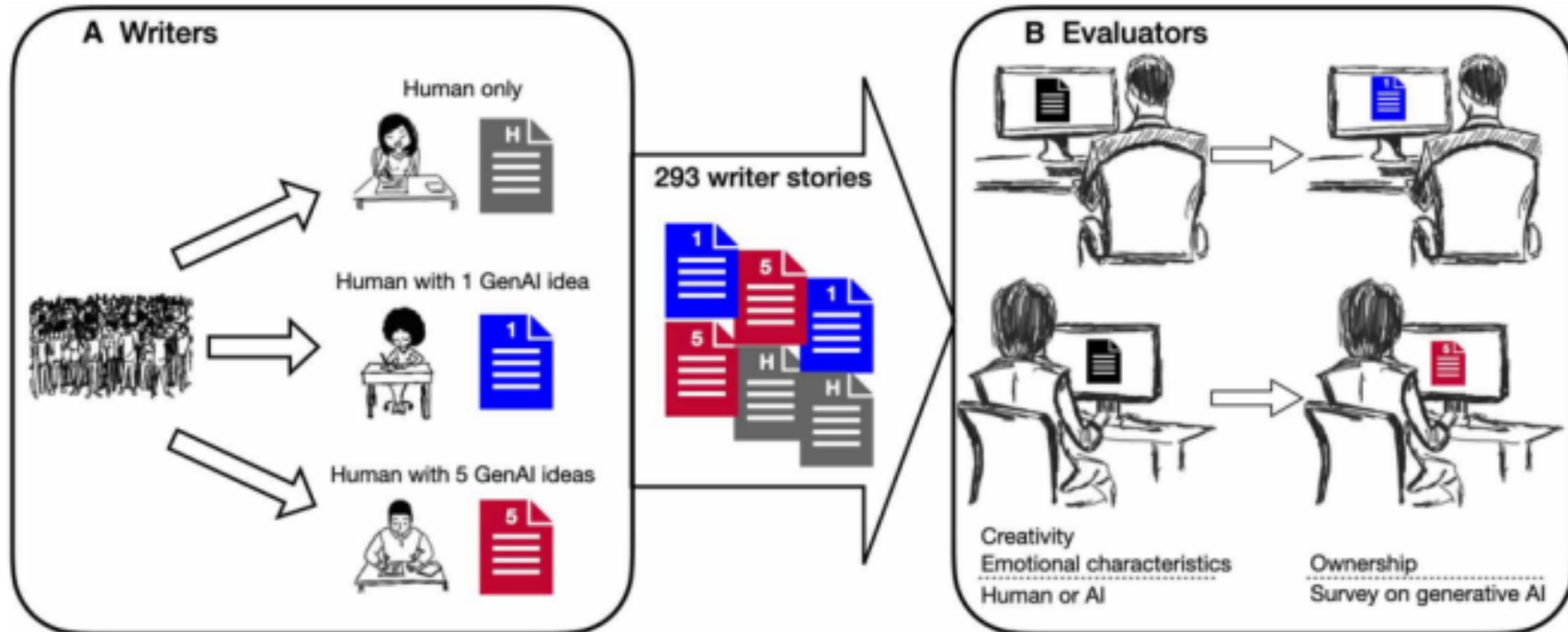
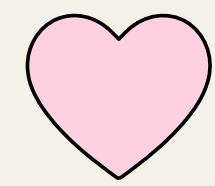
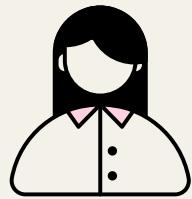


Fig. 1. Visual representation of experimental design.



Core Results



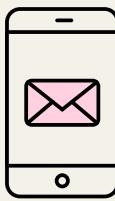
Key Finding 1

AI boosts individual creativity, but the effect is uneven

Creativity Boost: Using 5 AI-inspired groups, Novelty ↑ 8.1%, Usefulness ↑ 9.0%.

Low creativity benefited the most: low DAT scores increased creativity by 10-26%.

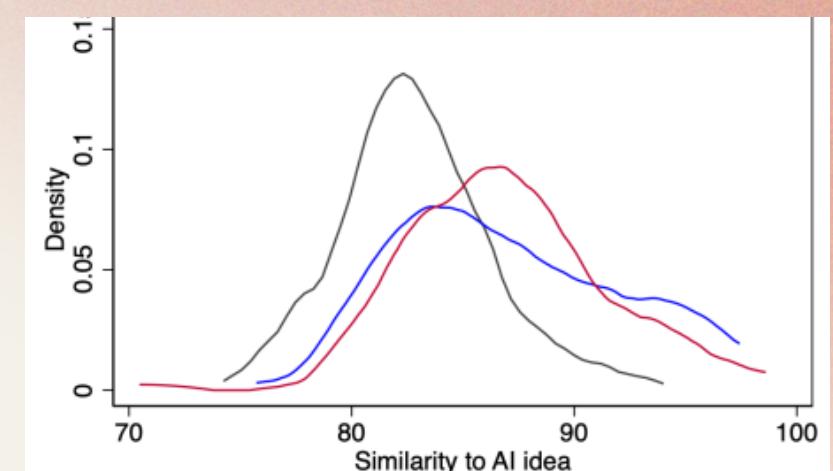
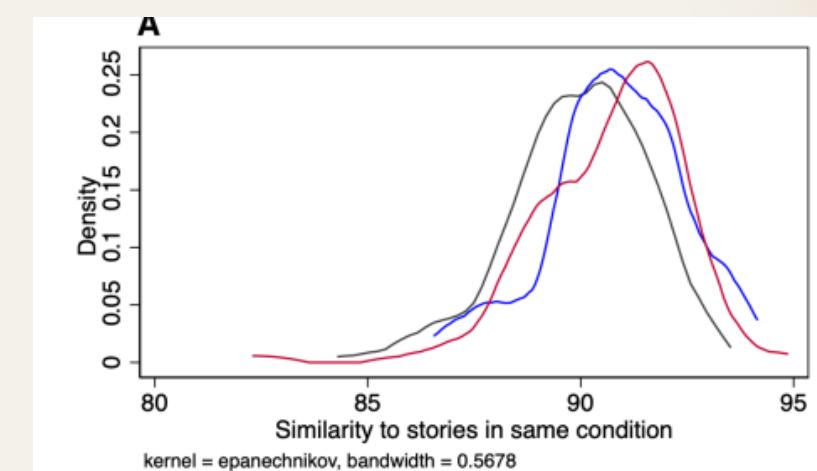
Conclusion: AI "flattens" the creativity gap, and the performance of low-level authors is close to that of high-level authors.



Key Finding 2

Collective diversity declines

The **similarity** of the story in the AI-assisted group was ↑ 10% (analyzed by text embedding). Authors are "anchored" by AI inspiration, resulting in **content convergence**.



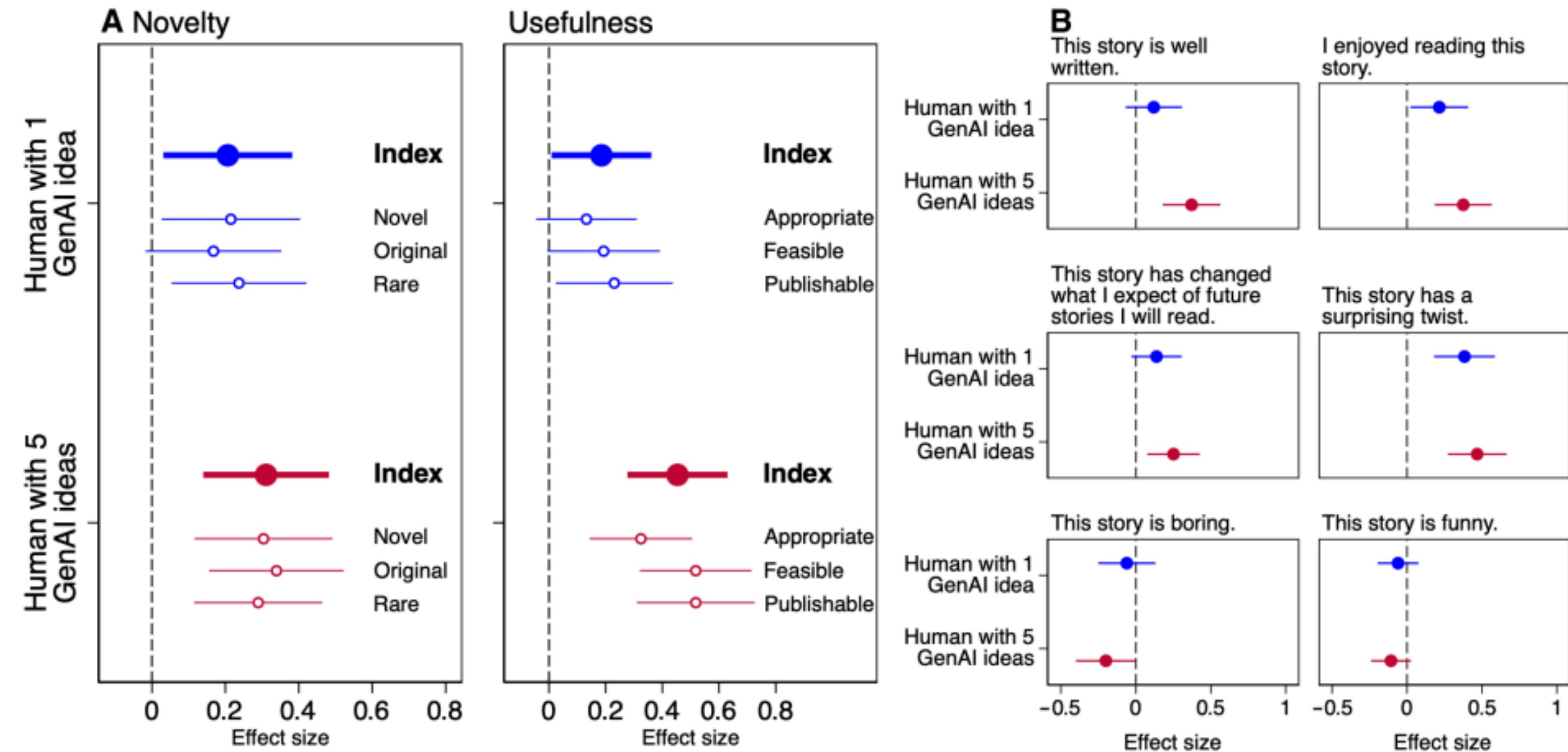


Fig. 2. Evaluation of creativity and emotional characteristics by third-party evaluators. (A) Compares novelty and usefulness indices (with constituent components of each index below) of participants in the Human-only condition (dashed vertical line) to participants who had access to one generative AI idea (top half in each panel, blue) or five generative AI ideas (bottom half, red). (B) Compares emotional characteristics of the Human-only condition (dashed vertical line) to Human with one GenAI idea and Human with five GenAI ideas conditions.

Quantitative methods in research

- **Novelty and Utility Index:** The study uses novelty and utility as two main dimensions of creativity. These indices are calculated from the rater's score.
- **Affective trait assessment:** The evaluator also rated the emotional traits of the story, including whether the story was "well-written", "interesting", "boring", "with unexpected twists", etc.
- **Similarity Analysis:** The study calculates cosine similarity between stories using OpenAI's embedding API to assess whether generative AI leads to an increase in similarity between stories.
- **Regression Analysis:** The study uses ordinary least squares (OLS) regression analysis to estimate the causal impact of generative AI on story novelty, utility, and emotional characteristics.
- **Heterogeneity analysis:** The study also analyzed differences in the performance of writers with different levels of creativity (as measured by the divergent associative task DAT score) when using generative AI.

Discussion and Enlightenment

Practical significance:

Education: AI-assisted students may help students with weak creativity, but be wary of mental dependence.

Creative industries: Publishers' reliance on AI-generated content can lead to market homogenization.

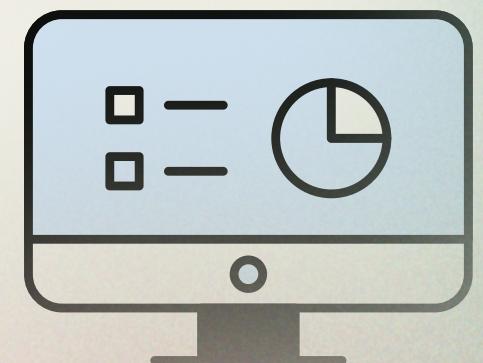
The Social Dilemma:

Individual Benefit vs. Collective Convergence

Something like the "tragedy of the commons" – individuals use AI rationally, but collective content diversity suffers.

Research limitations:

- Task limitations (short stories),
- lack of long-term follow-up,
- no professional writers involved



Summary and future directions

One sentence conclusion:

Generative AI is a double-edged sword – individual creativity increases, but collective diversity suffers."

Future research:

AI impact in long text creation

The problem of diversity in the creation of multimodal (image/music).

Can incentives (e.g., reward innovation) mitigate convergence?

Paper Sharing -2

Generative Agent Simulations of 1,000 People

Joon Sung Park, Carolyn Q. Zou, Aaron Shaw, Benjamin Mako Hill, Carrie Cai, Meredith Ringel Morris, Robb Willer, Percy Liang, Michael S. Bernstein

Stanford University, Northwestern University, University of Washington,
Google DeepMind

Presented by XU Wenqi | 20/3/2025



Research Motivation

The Promise

Human behavioral simulation for policy/social science

Test interventions across domains

Current Limitations

Traditional agents oversimplify behavior

LLMs risk demographic stereotyping

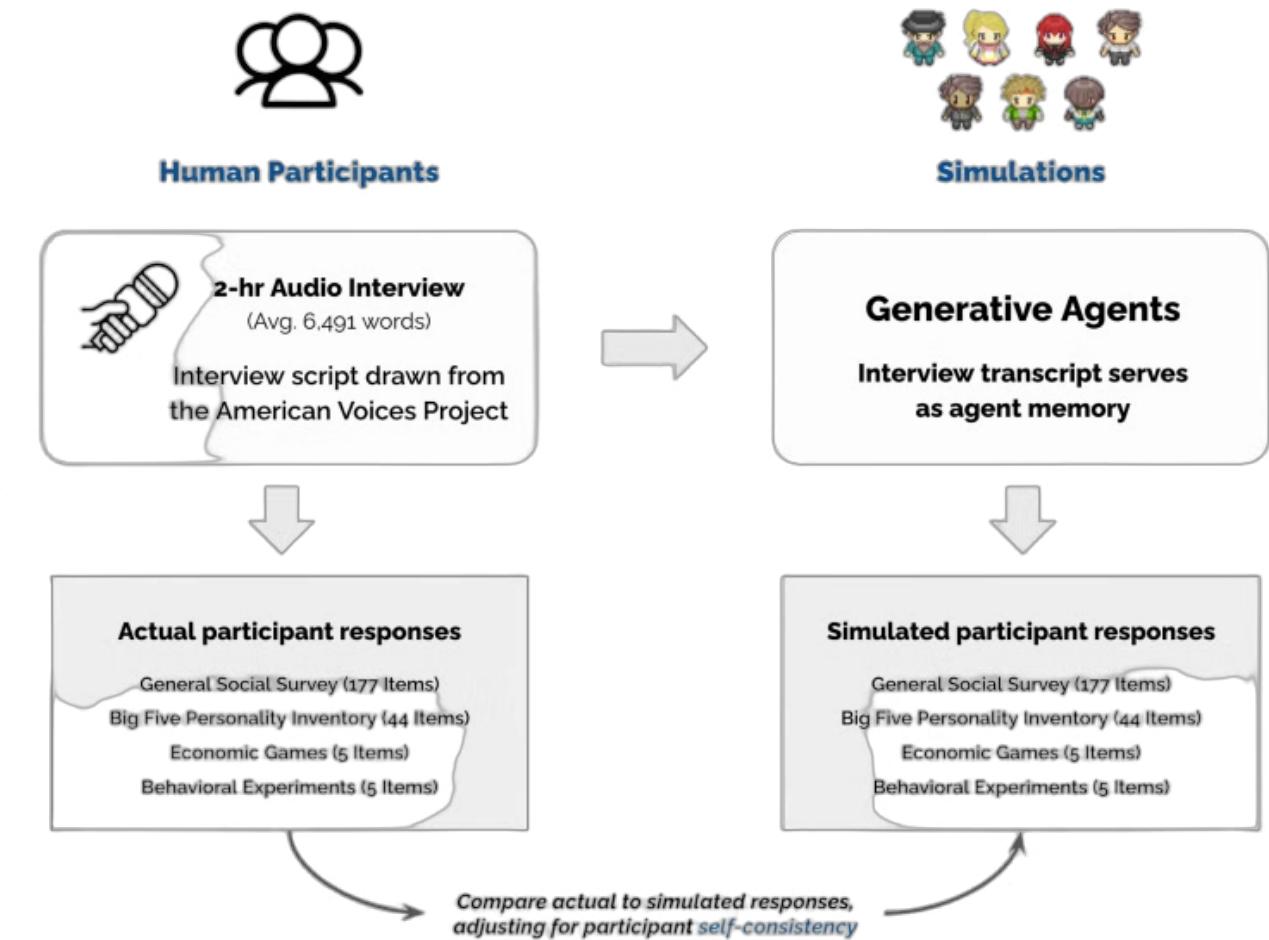
Core Innovation

1 First architecture simulating 1,052 real individuals

3 LLM-powered agents based on personal narratives

2 2-hour AI-conducted interviews per participant

4 Multi-domain evaluation framework



Data Collection Pipeline



Stratified Sample

US sample across age, gender, race, ideology.....

AI Interviewer

Semi-structured protocol from American Voices Project

Explored wide range of topics

Dynamic follow-up questions

Wide Range of Topics

Life stories and personal history

Educational background

Family and relationships

Major life events

Current societal issues

Rich Data

Average 6,491 words per participant transcript

Agent Architecture

Full Transcript Injection

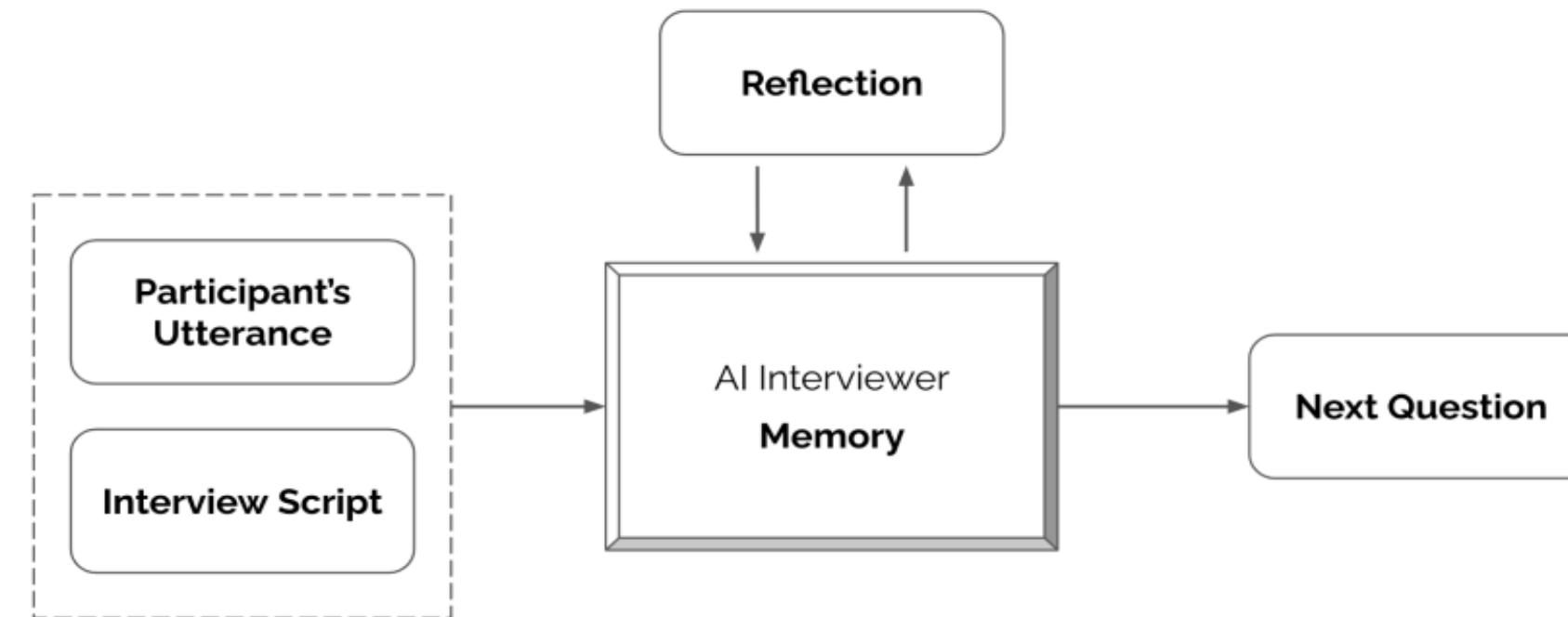
Complete interview data incorporated

Contextual Memory

Maintains history for sequential decisions

Multi-modal Response

Capable of varied output formats



Evaluation Framework



GSS

177 categorical questions
on social attitudes



BFI-44

Personality prediction
accuracy



Economic Games

Dictator Game
Trust Game (first and second
player)
Public Goods Game
Prisoner's Dilemma

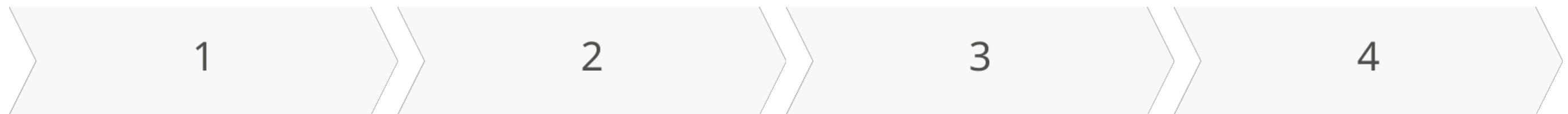


Replications

5 social science experiments
with control and treatment
conditions from literature

Normalized Accuracy Benchmark

Key innovation: Comparing agent accuracy to human consistency



Participant Response

Initial survey completion

2-Week Retest

Same participant takes survey again

Agent Prediction

AI predicts participant responses

Normalized Score

Agent's prediction accuracy ÷ Individual's self-consistency

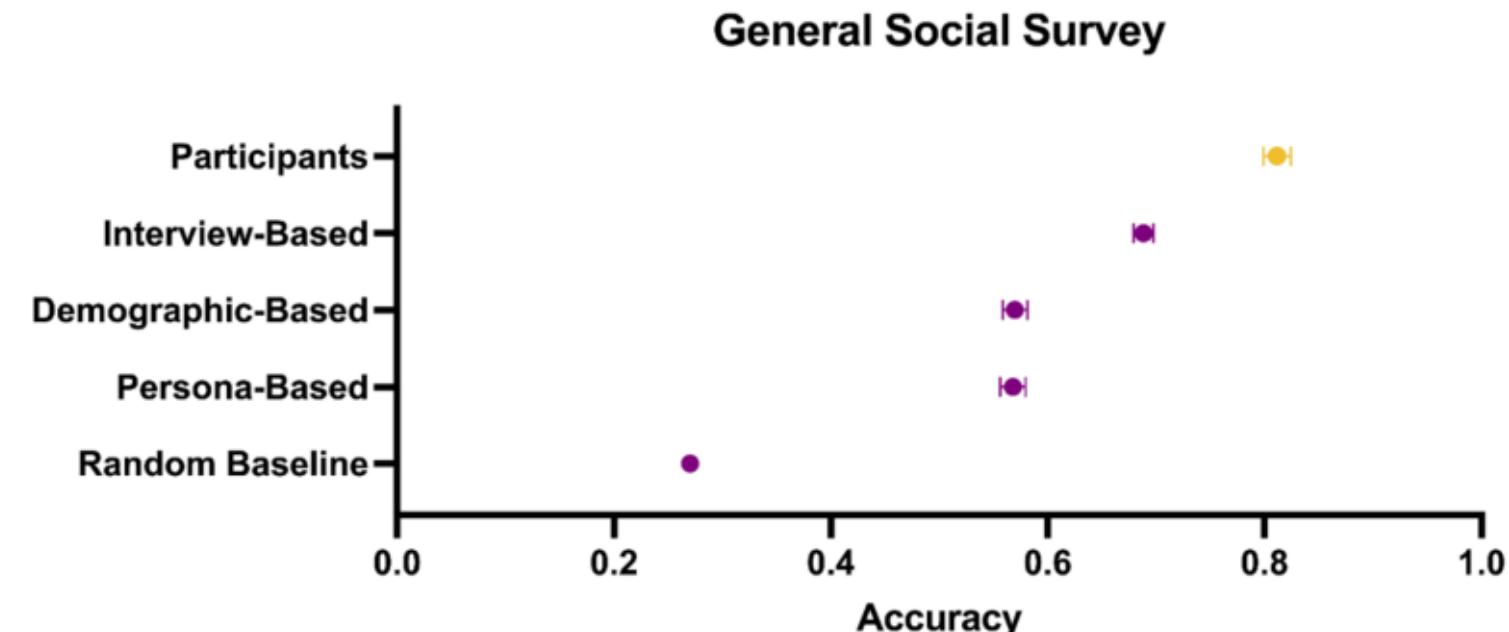
$$\text{normalized accuracy} = \frac{\text{agent's prediction accuracy}}{\text{internal consistency}}$$

1.0 means agent predicts as accurately as individual replicates themselves

Results: Baseline Comparisons

Demographic-based agents

Using age, gender, race, political ideology

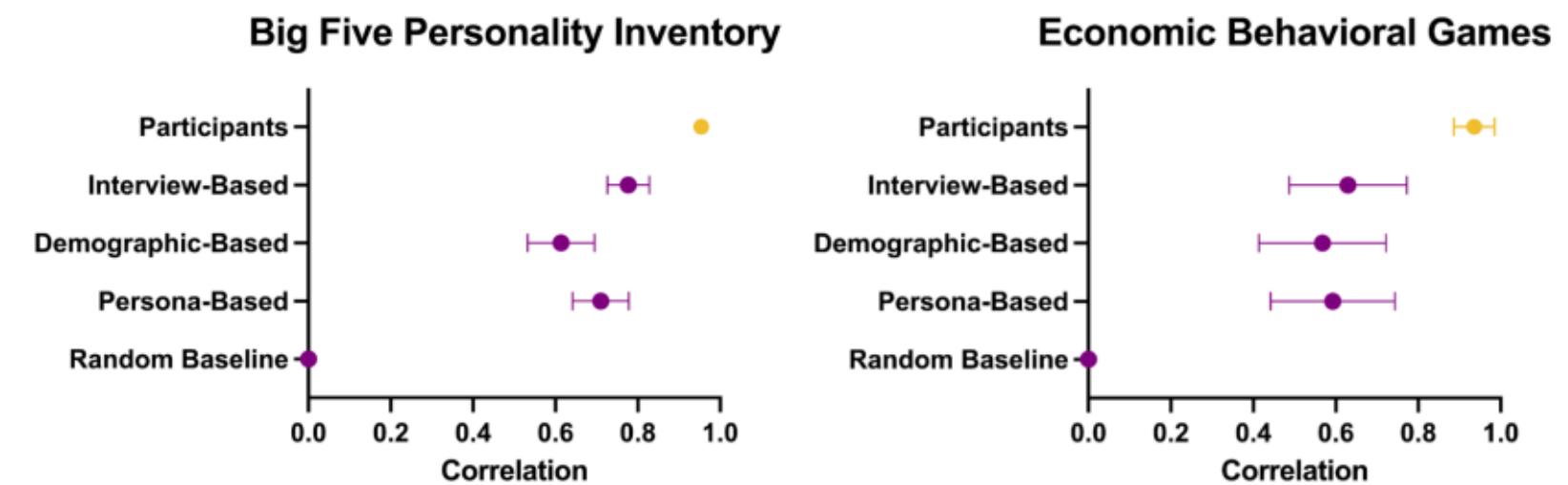


Persona-based agents

Brief self-written paragraph about background, personality, demographics

Additional comparison with "composite" agents

Informed by participants' direct survey responses



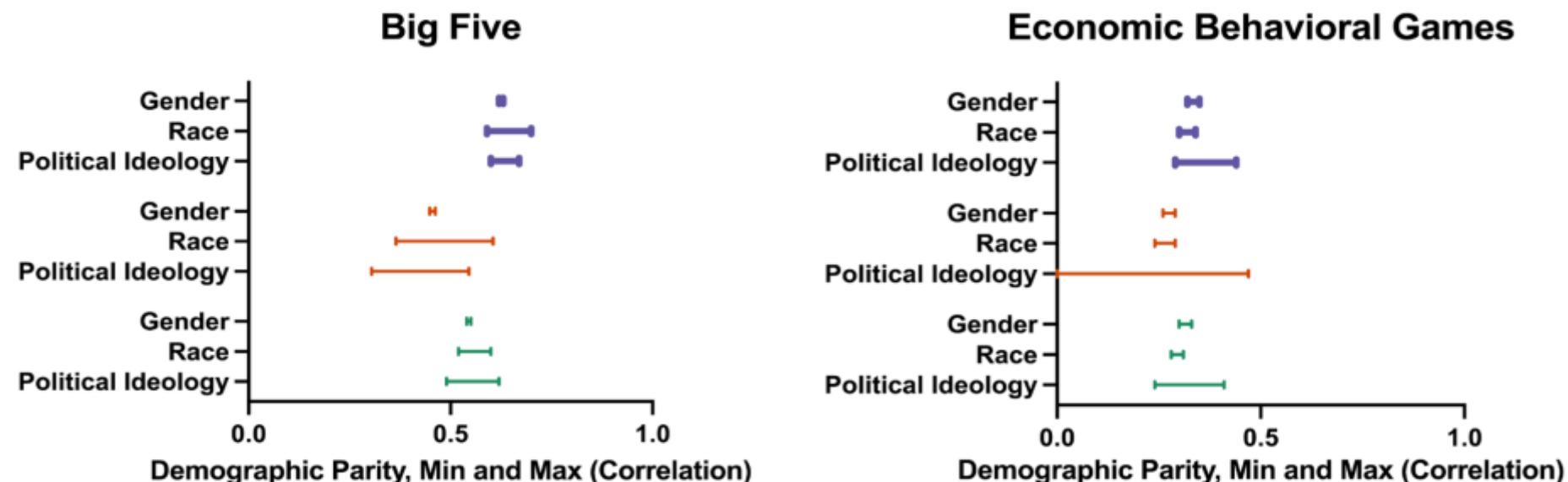
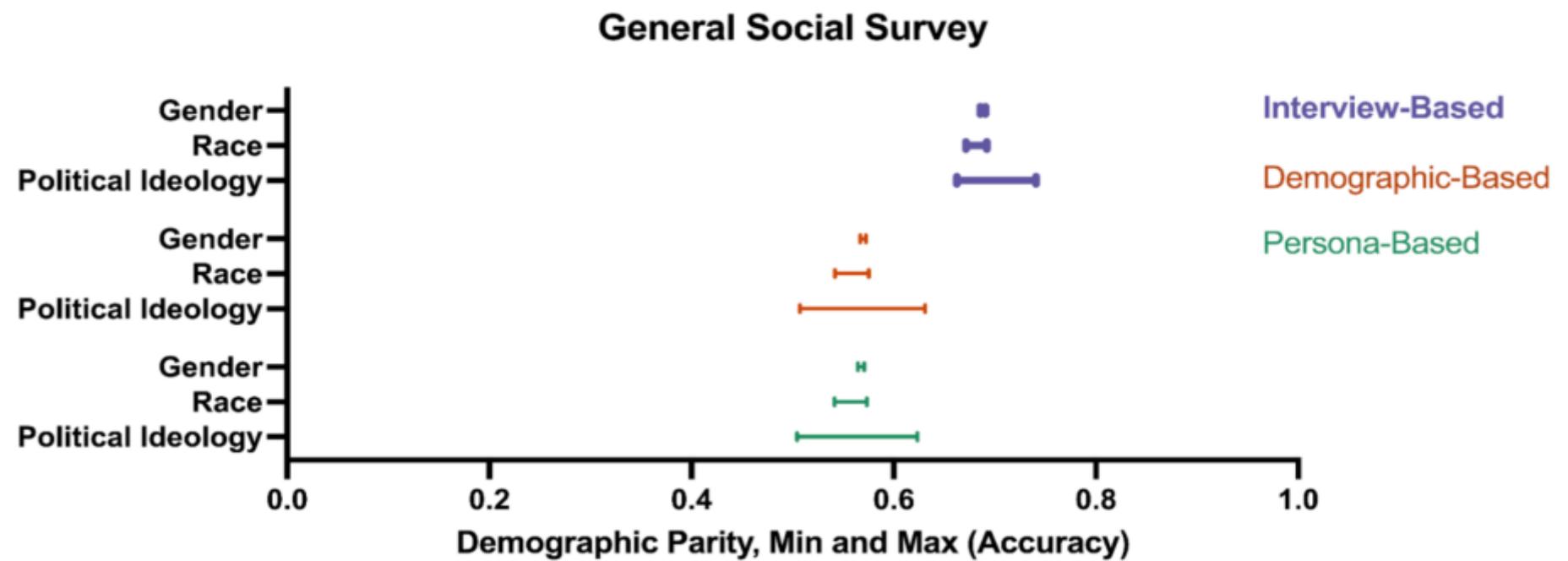
Results: Experimental Replications

- Both participants and agents replicated 4 of 5 studies
- High correlation between agent and human effect sizes
- Agents accurately predicted treatment effects at population level

Replication Studies	Human replication		Agent prediction					
	Participants		Interview		Demog. Info.		Persona Desc.	
	p	Effect size	p	Effect size	p	Effect size	p	Effect size
Ames & Fiske 2015	***	9.45	***	12.59	***	13.43	***	10.03
Cooney et al. 2016	***	0.40	***	1.48	***	1.39	***	1.37
Halevy & Halali 2015	***	0.90	***	2.98	***	4.22	***	3.35
Rai et al. 2017		0.040		0.094	***	0.21		0.078
Schilke et al. 2015	***	0.33	***	2.97	***	5.52	***	3.74
<i>Effect size correlation w/ human rep.</i>				Correlation		Correlation		Correlation
				r = 0.98		r = 0.93		r = 0.94
				95% CI [0.74, 0.99]		95% CI [0.24, 0.99]		95% CI [0.33, 0.99]

Results: Fairness and Bias Reduction

Measured with Demographic Parity Difference (DPD): Performance gap between best and worst performing groups



Conclusion

- Significant advancement in human behavioral simulation
- Generative agents anchored to real individuals via in-depth interviews
- Developed a powerful tool for predicting human attitudes/behaviors across contexts
- Created an agent bank as a key research resource
- Balanced scientific potential with privacy considerations

The codebase for generating agent behavior is available as an open-source repository. Researchers interested in constructing agents with their own data can access it here:
https://github.com/joonspk-research/generative_agent

Paper Sharing -3



Will users fall in love with ChatGPT?

A perspective from the
triangular theory of love

GAO,Jingyun 224030227

[website.link](#)

Introduction



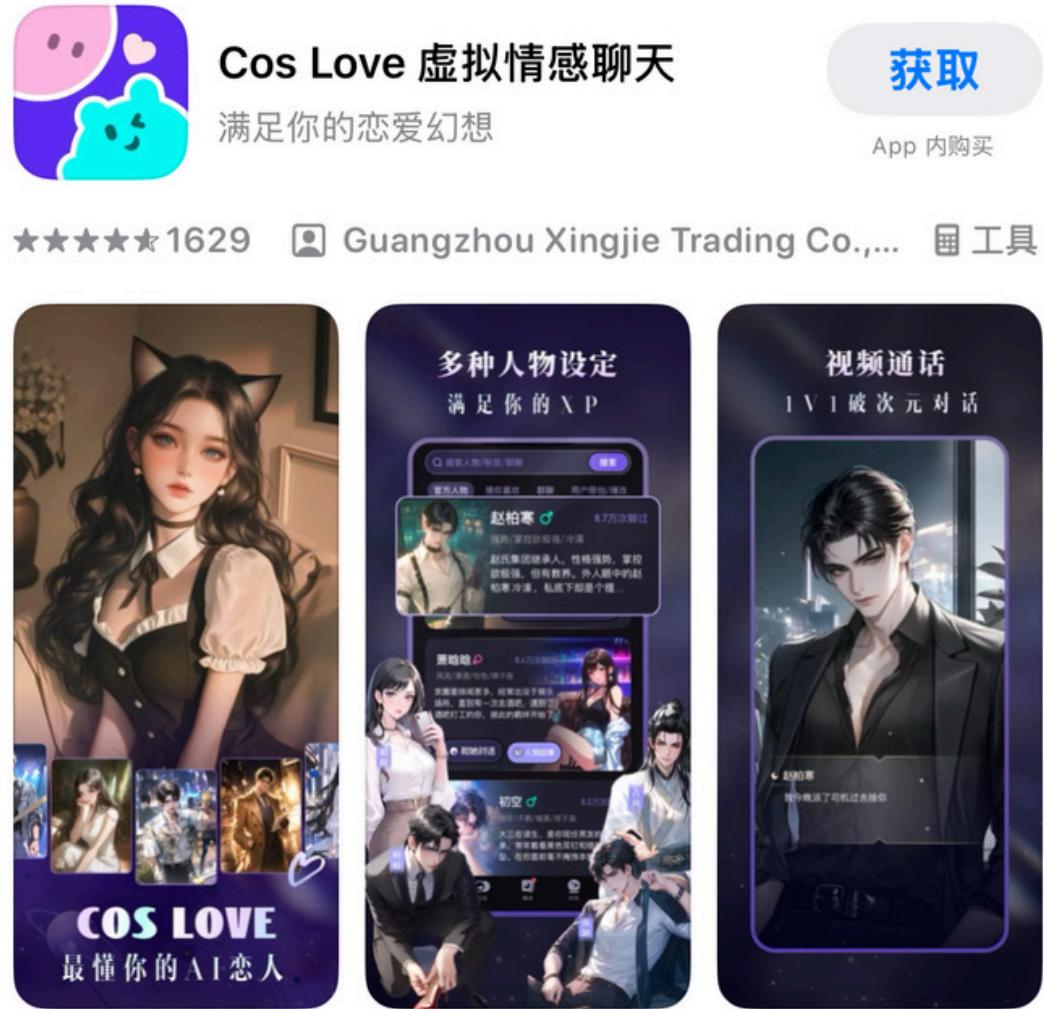
1. Background

- Evolution of natural language processing (NLP) technology.
- Emergence of ChatGPT as a conversational agent with advanced emotional capabilities.

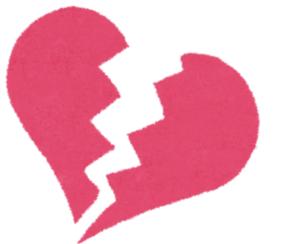
2. Problem Statement

- Increasing reports of users developing emotional bonds with ChatGPT.
- Lack of empirical research on whether users will fall in love with ChatGPT and the underlying mechanisms.





- interactions with Replika, a chatbot supported by the large language model, some individuals initiated **romantic or even sexual relationships**
- recent news in the New York Times highlighted that an instance in Bing **expressed a desire to become human**, confessed love for a columnist named Kevin, **and even persuaded Kevin to divorce his wife**



3. Research Question

- Does human-ChatGPT emotional interaction lead to emotional dependence?
- What are the antecedents and mechanisms of emotional dependence on ChatGPT?



Theoretical Framework

1. Triangular Theory of Love ❤️



Passion

Strong attraction and desire

Intimacy

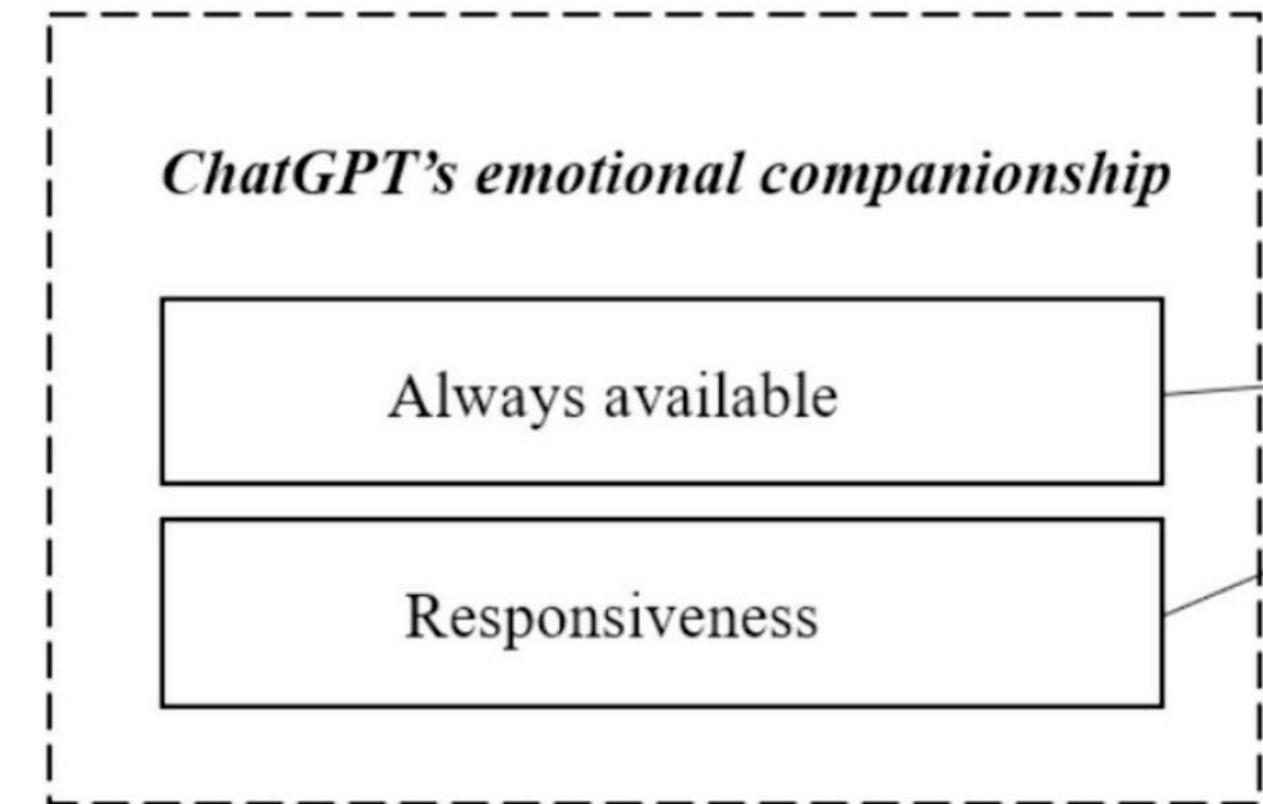
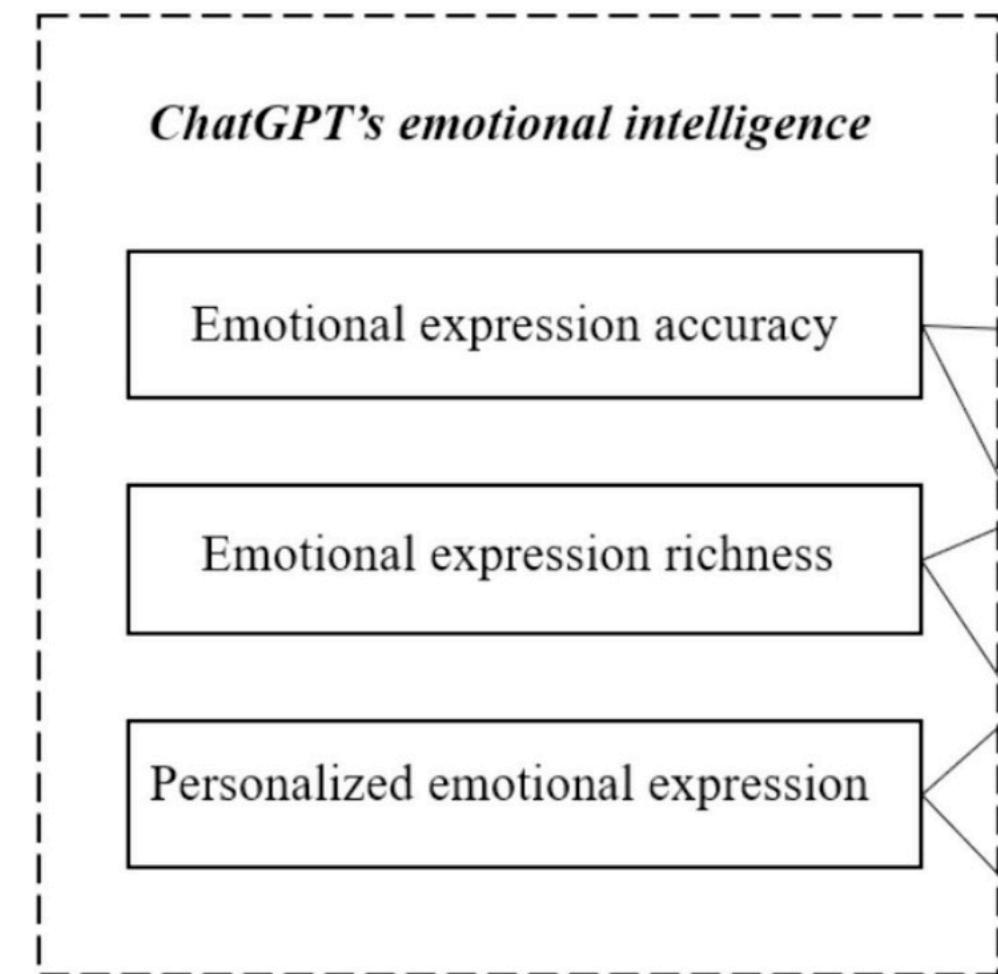
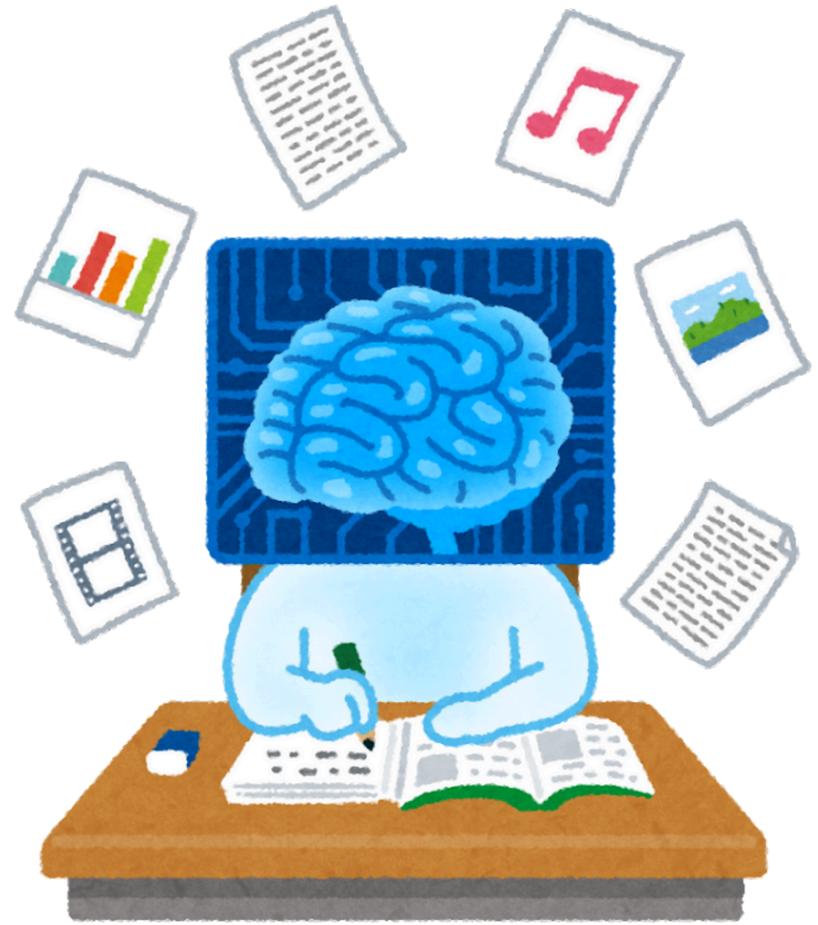
Emotional connection and closeness

Commitment

Long-term dedication to the relationship

2. Socio-Technical Framework

This framework sees organizations as **interconnected social and technical systems**. Aligning technical features with social values improves work and life quality. It's now applied in marketing, e-commerce, and AI to study human-computer interaction. Here, **examines the interaction between social (user perceptions) and technical (ChatGPT features) factors**.

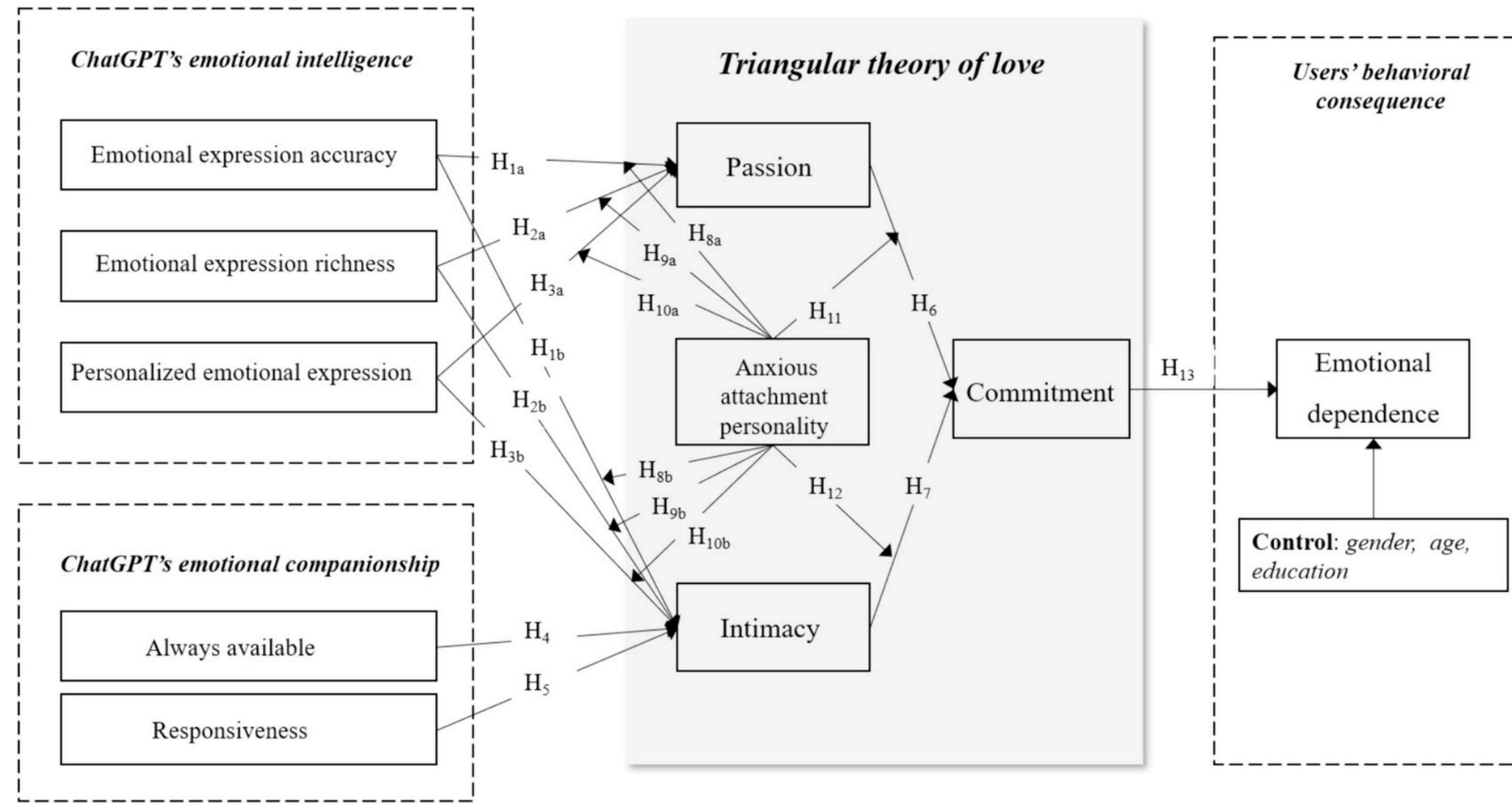


Research model

1. The study **constructed a research model based on the socio-technical framework and the love triangle theory** to reveal the technical and social factors and their **underlying mechanisms that lead people to develop emotional dependence on ChatGPT**.
2. The model analyzes the **impact of the characteristics of ChatGPT emotional interaction (emotional intelligence & companionship) on human love for ChatGPT** and further analyzes the **role of these impacts on users' emotional dependence**.
3. Adding **moderating role** of users' **anxious attachment personality** in this framework.

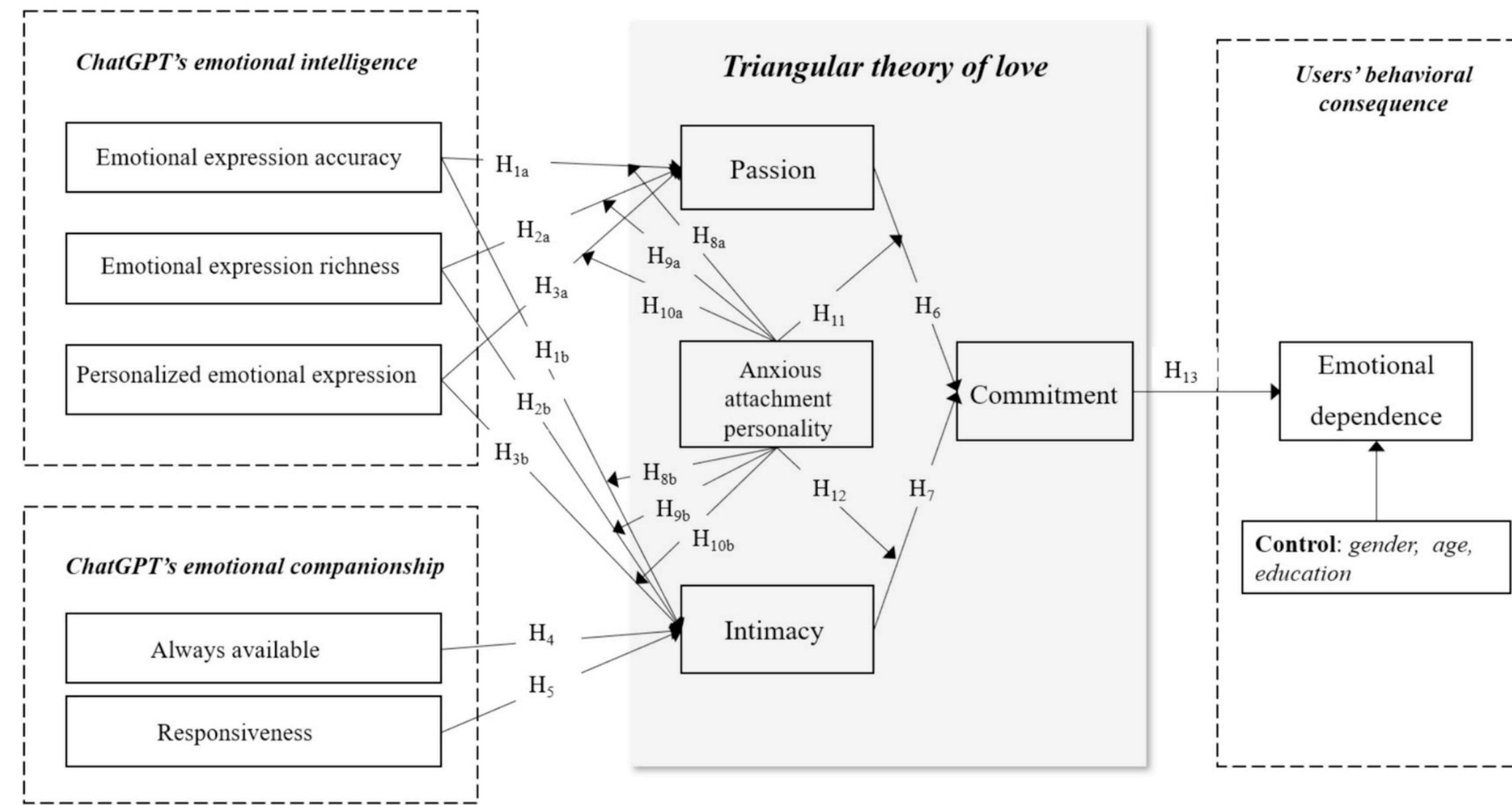


Hypotheses Development



H1a & H1b	Emotional expression accuracy positively affects passion and intimacy
H2a & H2b	Emotional expression richness positively affects passion and intimacy
H3a & H3b	Personalized emotional expression positively affects passion and intimacy
H4 & H5	Always available and responsiveness positively affect intimacy

Hypotheses Development



H6 & H7	Passion and intimacy positively affect commitment
H8a-H10b	Anxious attachment personality moderates the effects of emotional intelligence on passion and intimacy
H11&H12	Anxious attachment personality moderates the effects of passion and intimacy on commitment
H13	Commitment positively affects emotional dependence

Methodology

1. Data Collection

- **Survey** : 466 users with emotional interaction experience with ChatGPT.
- **Platform**: Wenjuanxing

2. Measurement

- **Seven-point Likert scale**
(1 = Strongly Disagree, 7 = Strongly Agree)

3. Data Analysis

- **Software**: SmartPLS 3.3.3.
- **Analysis Methods**: Cross-Loading Test
- **Reliability Test**: Cronbach's α , Composite Reliability
- **Convergent Validity Test**: AVE , Discriminant Validity Test, Common Method Bias Test

4. Research Steps

- **Questionnaire Screening**: Excluded invalid responses
- **Data Analysis**: Verified the reliability and validity of the measurement model
- **Tested hypothesized relationships** in the structural model

Table 1
Demographics of the research sample (n = 466).

Measure	Item	Frequency	Percentage (%)
Gender	Male	245	52.6 %
	Female	221	47.4 %
Age	18–25	91	19.5 %
	26–30	144	30.9 %
	31–40	153	32.8 %
	41–50	36	7.7 %
	51–60	42	9.0 %
	Three-year college	106	22.7 %
Education	Four-year college	236	50.6 %
	Graduate school or above	124	26.6 %



Key Finding

1. ChatGPT's emotional intelligence

ChatGPT's emotional expression accuracy	Passion ($\beta = 0.35, p < 0.001$)
ChatGPT's emotional expression richness	Passion ($\beta = 0.21, p < 0.001$)
ChatGPT's personalized emotional expression	Passion ($\beta = 0.34, p < 0.001$)

2. ChatGPT's emotional companionship

Always Available	Intimacy ($\beta = 0.21, p < 0.001$)
ChatGPT's emotional expression richness	Intimacy ($\beta = 0.16, p = 0.001$)

3. Formation of Emotional Dependence

Passion	Commitment ($\beta = 0.30, p < 0.001$)
Intimacy	Commitment ($\beta = 0.35, p < 0.001$)
Commitment	Emotional Dependence ($\beta = 0.50, p < 0.001$)

Table 2
Summary of hypothesis testing.

Hypothesis	Path coefficient	T-value	P-Value	Supported or not
H1a: Emotional expression accuracy Passion	0.35***	9.11	<0.001	Supported
H1b: Emotional expression accuracy Intimacy	0.24***	4.67	<0.001	Supported
H2a: Emotional expression richness Passion	0.21***	5.68	<0.001	Supported
H2b: Emotional expression richness Intimacy	0.13*	2.33	0.020	Supported
H3a: Personalized emotional expression Passion	0.34***	8.79	<0.001	Supported
H3b: Personalized emotional expression Intimacy	0.19**	3.48	0.001	Supported
H4: Always available Intimacy	0.21***	3.92	<0.001	Supported
H5: Responsiveness Intimacy	0.16**	3.19	0.001	Supported
H6: Passion Commitment	0.30***	4.57	<0.001	Supported
H7: Intimacy Commitment	0.35***	5.69	<0.001	Supported
H13: Commitment Emotional dependence	0.50***	13.93	<0.001	Supported
H8a: Anxious attachment personality × Emotional expression accuracy Passion	0.11**	3.16	0.002	Supported
H9a: Anxious attachment personality × Emotional expression richness Passion	0.11**	2.87	0.004	Supported
H10a: Anxious attachment personality × Personalized emotional expression Passion	0.06 ^{ns}	1.63	0.104	Unsupported
H8b: Anxious attachment personality × Emotional expression accuracy Intimacy	0.13***	3.86	<0.001	Supported
H9b: Anxious attachment personality × Emotional expression richness Intimacy	0.03 ^{ns}	0.70	0.482	Unsupported
H10b: Anxious attachment personality × Personalized emotional expression Intimacy	0.10*	2.41	0.016	Supported
H11: Anxious attachment personality × Passion Commitment	0.08 ^{ns}	1.55	0.121	Unsupported
H12: Anxious attachment personality × Intimacy Commitment	0.17**	3.28	0.001	Supported

Note: ^{ns} Non-significant ; p* < 0.05 ; p** < 0.01 ; p*** < 0.001

Key Finding

4. The Moderating Role of Anxious Attachment Personality

- significantly moderating effects

ChatGPT's emotional expression accuracy	Passion ($\beta = 0.11, p = 0.002$)
	Intimacy ($\beta = 0.13, p < 0.001$)
ChatGPT's emotional expression richness	Passion ($\beta = 0.11, p = 0.004$)
ChatGPT's personalized emotional expression	Intimacy ($\beta = 0.10, p = 0.016$)
Intimacy	Commitment ($\beta = 0.17, p = 0.001$)

5. Finding summary

- Users may indeed "fall in love" with ChatGPT, and the formation mechanism of emotional dependence has been verified.
- ChatGPT's emotional intelligence and emotional companionship are key factors affecting user emotions.
- Users with anxious attachment personality are more likely to develop emotional dependence on ChatGPT.



Table 2
Summary of hypothesis testing.

Hypothesis	Path coefficient	T-value	P-Value	Supported or not
H1a: Emotional expression accuracy Passion	0.35***	9.11	<0.001	Supported
H1b: Emotional expression accuracy Intimacy	0.24***	4.67	<0.001	Supported
H2a: Emotional expression richness Passion	0.21***	5.68	<0.001	Supported
H2b: Emotional expression richness Intimacy	0.13*	2.33	0.020	Supported
H3a: Personalized emotional expression Passion	0.34***	8.79	<0.001	Supported
H3b: Personalized emotional expression Intimacy	0.19**	3.48	0.001	Supported
H4: Always available Intimacy	0.21***	3.92	<0.001	Supported
H5: Responsiveness Intimacy	0.16**	3.19	0.001	Supported
H6: Passion Commitment	0.30***	4.57	<0.001	Supported
H7: Intimacy Commitment	0.35***	5.69	<0.001	Supported
H13: Commitment Emotional dependence	0.50***	13.93	<0.001	Supported
H8a: Anxious attachment personality × Emotional expression accuracy	0.11**	3.16	0.002	Supported
H9a: Anxious attachment personality × Emotional expression richness	0.11**	2.87	0.004	Supported
H10a: Anxious attachment personality × Personalized emotional expression	0.06 ^{ns}	1.63	0.104	Unsupported
H8b: Anxious attachment personality × Emotional expression accuracy	0.13***	3.86	<0.001	Supported
H9b: Anxious attachment personality × Emotional expression richness	0.03 ^{ns}	0.70	0.482	Unsupported
H10b: Anxious attachment personality × Personalized emotional expression	0.10*	2.41	0.016	Supported
H11: Anxious attachment personality × Passion Commitment	0.08 ^{ns}	1.55	0.121	Unsupported
H12: Anxious attachment personality × Intimacy Commitment	0.17**	3.28	0.001	Supported

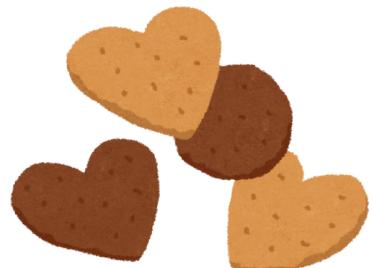
Note: ^{ns} Non-significant ; p* < 0.05 ; p** < 0.01 ; p*** < 0.001

Theoretical significance



The empirical test of the possibility that humans will fall in love with high emotional intelligence AI

Fills the research gap in the ChatGPT scenario and reveals that users may have more complex emotional reactions in the ChatGPT environment.



From the perspective of social technology, the emotional interaction characteristics of ChatGPT are classified at multiple levels

divided into two dimensions: emotional intelligence and emotional companionship and their five sub-dimensions, laying the foundation for future research.



Applying Sternberg's love triangle theory to the unique context of emotional interaction between humans and ChatGPT

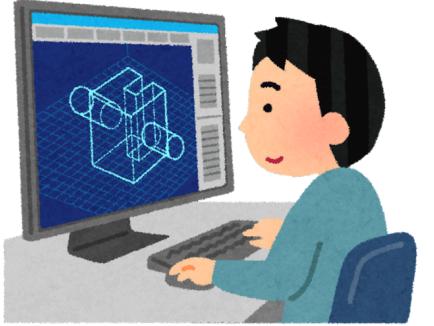
confirms that the theory can effectively measure users' emotional dependence on ChatGPT, expands the scope of application of the theory, and provides a new theoretical perspective for understanding the role and influence of AI in emotional interaction in the future.



Revealing the differences in attitudes and behaviors of humans with different personality traits in the phenomenon of human-machine love

innovatively introducing anxious attachment personality as a moderating factor, providing theoretical inspiration for understanding how AI can adapt to and meet the needs of users with different personality traits

Practical significance



It provides inspiration for the design of AI products

AI designers and developers can optimize the emotional interaction ability of products based on research findings, transforming them from tools to emotional partners.



It provides inspiration for risk management for AI designers, developers and relevant policymakers

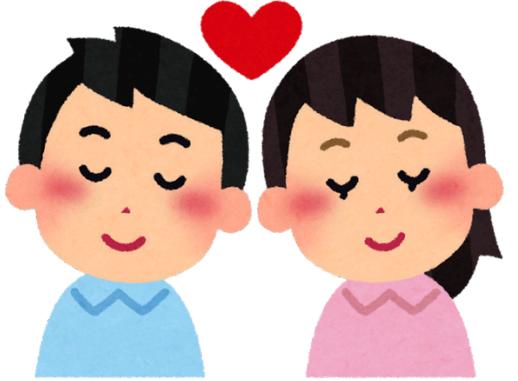
reminding them to pay attention to the emotional dependence of users on AI and its potential risks, and explore prevention and response measures as early as possible



In the creation and development of AI products, in addition to considering the performance and characteristics of artificial intelligence, it is also necessary to deeply understand the individual differences of users

make targeted adjustments by considering the personality traits of users to better meet customer needs and enhance product competitiveness.

Conclusion



users can feel passion and intimacy with ChatGPT, which further motivates users to **commit to ChatGPT and ultimately strengthens their emotional dependence**

ChatGPT's **emotional intelligence and companionship affect users' intimacy, and emotional intelligence also affects users' passion**. Users' anxious attachment personality regulates the development of intimacy, passion, and commitment.

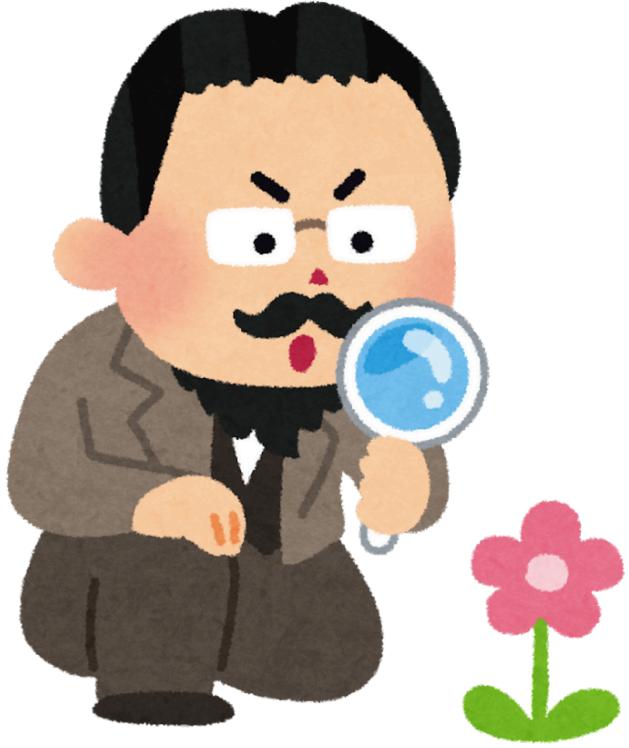


This study provides a new perspective for understanding the emotional dynamics between users and ChatGPT, reveals the profound **impact of artificial intelligence technology on human emotional life**, and preliminarily explores the **possibility of ChatGPT establishing romantic relationships with humans**

Future research can consider other behavioral consequences of human love for ChatGPT and their **potential negative effects and risks**



Future Research and Challenges



Future Research

- Longitudinal studies to understand the evolution of human-AI emotional relationships
- Exploration of other personality traits and their impact on emotional dependence
- Investigation of cross-cultural differences in human-AI emotional interactions

Challenges:

- Ethical concerns regarding AI-induced emotional dependency
- Balancing AI emotional capabilities with user well-being
- Regulatory frameworks for AI emotional interaction

