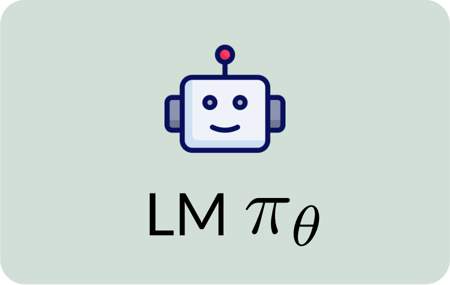
# Understanding and Overcoming Failures of Language Model Finetuning

**Noam Razin**

Princeton Language and Intelligence
Princeton University

# Language Models

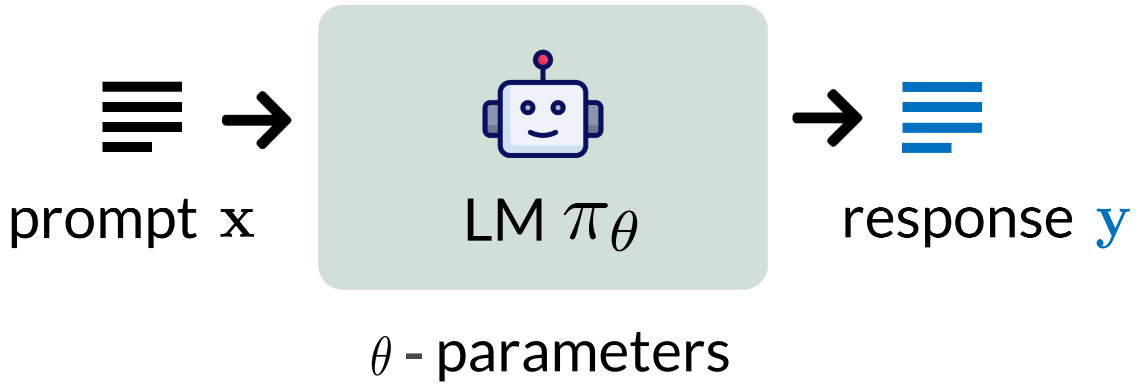
**Language Model (LM):** Neural network trained on large amounts of text data to produce a **distribution over text**



$\theta$ - parameters

# Language Models

**Language Model (LM):** Neural network trained on large amounts of text data to produce a **distribution over text**



prompt $\mathbf{x}$

LM $\pi_\theta$

response $\mathbf{y}$

$\theta$ - parameters

# Supervised Finetuning of LMs

To ensure LMs generate safe and helpful content, they are aligned via **finetuning**

# Supervised Finetuning of LMs

To ensure LMs generate safe and helpful content, they are aligned via **finetuning**

**Supervised Finetuning (SFT)**

Minimize cross entropy loss over labeled inputs

**Data Format:**

prompt $\mathbf{x}$

desired response $\mathbf{y}$

# Supervised Finetuning of LMs

To ensure LMs generate safe and helpful content, they are aligned via **finetuning**

## Supervised Finetuning (SFT)

Minimize cross entropy loss over labeled inputs

**Data Format:** prompt $\mathbf{x}$ desired response $\mathbf{y}$

**Limitations:**

Hard to formalize human preferences through labels

# Supervised Finetuning of LMs

To ensure LMs generate safe and helpful content, they are aligned via **finetuning**

### Supervised Finetuning (SFT)

Minimize cross entropy loss over labeled inputs

**Data Format:** prompt $\mathbf{x}$ desired response $\mathbf{y}$

**Limitations:**

- Hard to formalize human preferences through labels
- Obtaining high-quality responses is expensive

# Finetuning LMs via Preference Data

Limitations of SFT led to wide adoption of approaches using **preference data**

# Finetuning LMs via Preference Data

Limitations of SFT led to wide adoption of approaches using **preference data**

**Preference-Based Finetuning**

Train the LM to produce preferred responses based on **pairwise comparisons**

**Data Format:**

prompt $\mathbf{x}$

preferred response $\mathbf{y}^{+}$

dispreferred response $\mathbf{y}^{-}$

# Finetuning LMs via Preference Data

Limitations of SFT led to wide adoption of approaches using **preference data**

**Preference-Based Finetuning**

Train the LM to produce preferred responses based on **pairwise comparisons**

**Data Format:**

prompt $\mathbf{x}$

preferred response $\mathbf{y}^{+}$

dispreferred response $\mathbf{y}^{-}$

**Main Approaches:**

1 Reinforcement Learning (e.g. Ouyang et al. 2022)

2 Direct Preference Learning (e.g. Rafailov et al. 2023)

# Sources

1

Vanishing Gradients in Reinforcement Finetuning of Language Models

*R* + Zhou + Saremi + Thilak + Bradley + Nakkiran + Susskind + Littwin | *ICLR 2024*

2

Unintentional Unalignment: Likelihood Displacement in Direct Preference Optimization

*R* + Malladi + Bhaskar + Chen + Arora + Hanin | *arXiv 2024*

# Sources

1

Vanishing Gradients in Reinforcement Finetuning of Language Models

*R* + Zhou + Saremi + Thilak + Bradley + Nakkiran + Susskind + Littwin | *ICLR 2024*

2

Unintentional Unalignment: Likelihood Displacement in Direct Preference Optimization

*R* + Malladi + Bhaskar + Chen + Arora + Hanin | *arXiv 2024*

# Collaborators

Hattie Zhou

Omid Saremi

Vimal Thilak

Arwen Bradley

Preetum Nakkiran

Joshua Susskind

Etai Littwin

# Reinforcement Finetuning of LMs

## Reinforcement Finetuning (RFT)

# Reinforcement Finetuning of LMs

### Reinforcement Finetuning (RFT)

1. Learn a **reward model** $r(\mathbf{x}, \mathbf{y})$ by fitting preference data

# Reinforcement Finetuning of LMs

## Reinforcement Finetuning (RFT)

1. Learn a **reward model** $r(\mathbf{x}, \mathbf{y})$ by fitting preference data
2. Maximize reward over unlabeled prompts via **policy gradient methods**

Expected reward for input $\mathbf{x}$: $V_{\mathbf{x}}(\theta) = \mathbb{E}_{\mathbf{y} \sim \pi_\theta(\cdot|\mathbf{x})} \left[ r(\mathbf{x}, \mathbf{y}) \right]$

# Reinforcement Finetuning of LMs

## Reinforcement Finetuning (RFT)

1. Learn a **reward model** $r(\mathbf{x}, \mathbf{y})$ by fitting preference data
2. Maximize reward over unlabeled prompts via **policy gradient methods**

Expected reward for input $\mathbf{x}$: $V_{\mathbf{x}}(\theta) = \mathbb{E}_{\mathbf{y}\sim\pi_\theta(\cdot|\mathbf{x})}\left[r(\mathbf{x}, \mathbf{y})\right]$

When preferences are labeled by humans: RFT ⟷ RLHF (Ouyang et al. 2022)

# Reinforcement Finetuning of LMs

## Reinforcement Finetuning (RFT)

1. Learn a **reward model** $r(\mathbf{x}, \mathbf{y})$ by fitting preference data
2. Maximize reward over unlabeled prompts via **policy gradient methods**

Expected reward for input $\mathbf{x}$: $V_{\mathbf{x}}(\theta) = \mathbb{E}_{\mathbf{y} \sim \pi_\theta(\cdot|\mathbf{x})}\left[r(\mathbf{x}, \mathbf{y})\right]$

When preferences are labeled by humans: RFT ⟷ RLHF (Ouyang et al. 2022)

For our purposes, $r(\mathbf{x}, \mathbf{y})$ can be any arbitrary reward function

# Main Contributions: Vanishing Gradients in RFT

# Main Contributions: Vanishing Gradients in RFT

$\nabla \mathbf{V}_{\mathbf{x}}(\theta) \approx \mathbf{0}$

Theory: Fundamental vanishing gradients problem in RFT

# Main Contributions: Vanishing Gradients in RFT

$\nabla \mathbf{V}_{\mathbf{x}}(\theta) \approx \mathbf{0}$ Theory: Fundamental vanishing gradients problem in RFT

⚠ Vanishing gradients are prevalent and harm ability to maximize reward

# Main Contributions: Vanishing Gradients in RFT

$\nabla \mathbf{V}_{\mathbf{x}}(\theta) \approx \mathbf{0}$ Theory: Fundamental vanishing gradients problem in RFT

⚠ Vanishing gradients are prevalent and harm ability to maximize reward

💡 Exploring ways to overcome vanishing gradients in RFT

# Main Contributions: Vanishing Gradients in RFT

$\nabla \mathbf{V}_{\mathbf{x}}(\theta) \approx \mathbf{0}$ Theory: Fundamental vanishing gradients problem in RFT

Vanishing gradients are prevalent and harm ability to maximize reward

Exploring ways to overcome vanishing gradients in RFT

# Vanishing Gradients Due to Small Reward Standard Deviation (STD)

$\mathrm{STD}_{\mathbf{y} \sim \pi_\theta(\cdot|\mathbf{x})}[r(\mathbf{x}, \mathbf{y})]$ – reward std of $\mathbf{x}$ under the model

# Vanishing Gradients Due to Small Reward Standard Deviation (STD)

$\mathrm{STD}_{\mathbf{y}\sim\pi_\theta(\cdot|\mathbf{x})}[r(\mathbf{x},\mathbf{y})]$ – reward std of $\mathbf{x}$ under the model

**Theorem**

$$\|\nabla V_{\mathbf{x}}(\theta)\| = O\big(\mathrm{STD}_{\mathbf{y}\sim\pi_\theta(\cdot|\mathbf{x})}[r(\mathbf{x},\mathbf{y})]^{2/3}\big)$$

*Same holds for PPO gradient

# Vanishing Gradients Due to Small Reward Standard Deviation (STD)

$\mathrm{STD}_{\mathbf{y} \sim \pi_\theta(\cdot|\mathbf{x})}[r(\mathbf{x}, \mathbf{y})]$ — reward std of $\mathbf{x}$ under the model

**Theorem**

$$\|\nabla V_{\mathbf{x}}(\theta)\| = O\big(\mathrm{STD}_{\mathbf{y} \sim \pi_\theta(\cdot|\mathbf{x})}[r(\mathbf{x}, \mathbf{y})]^{2/3}\big)$$

ⓘ **Expected gradient for an input vanishes when reward std is small, even if reward mean is suboptimal**

*Same holds for PPO gradient

# Vanishing Gradients Due to Small Reward Standard Deviation (STD)

$\mathrm{STD}_{\mathbf{y}\sim\pi_\theta(\cdot|\mathbf{x})}[r(\mathbf{x},\mathbf{y})]$ – reward std of $\mathbf{x}$ under the model

**Theorem**

$$\|\nabla V_{\mathbf{x}}(\theta)\| = O\left(\mathrm{STD}_{\mathbf{y}\sim\pi_\theta(\cdot|\mathbf{x})}[r(\mathbf{x},\mathbf{y})]^{2/3}\right)$$

ⓘ **Expected gradient for an input vanishes when reward std is small, even if reward mean is suboptimal**

*Same holds for PPO gradient

**Proof Idea:** Stems from use of softmax + reward maximization objective

# Vanishing Gradients Due to Small Reward Standard Deviation (STD)

$\mathrm{STD}_{\mathbf{y}\sim\pi_\theta(\cdot|\mathbf{x})}[r(\mathbf{x},\mathbf{y})]$ — reward std of $\mathbf{x}$ under the model

**Theorem**

$$\|\nabla V_{\mathbf{x}}(\theta)\| = O\left(\mathrm{STD}_{\mathbf{y}\sim\pi_\theta(\cdot|\mathbf{x})}[r(\mathbf{x},\mathbf{y})]^{2/3}\right)$$

*Same holds for PPO gradient

ⓘ **Expected gradient for an input vanishes when reward std is small, even if reward mean is suboptimal**

**Proof Idea:** Stems from use of softmax + reward maximization objective

**Note:** Bound applies to expected gradients of individual inputs (as opposed to of batch/population)

# Vanishing Gradients Due to Small Reward Standard Deviation (STD)

$\mathrm{STD}_{\mathbf{y}\sim\pi_\theta(\cdot|\mathbf{x})}[r(\mathbf{x},\mathbf{y})]$ — reward std of $\mathbf{x}$ under the model

**Theorem**

$$\|\nabla V_{\mathbf{x}}(\theta)\| = O\left(\mathrm{STD}_{\mathbf{y}\sim\pi_\theta(\cdot|\mathbf{x})}[r(\mathbf{x},\mathbf{y})]^{2/3}\right)$$

*Same holds for PPO gradient

ⓘ **Expected gradient for an input vanishes when reward std is small, even if reward mean is suboptimal**

**Proof Idea:** Stems from use of softmax + reward maximization objective

**Note:** Bound applies to expected gradients of individual inputs (as opposed to of batch/population)

**RFT may not work well for inputs with small reward std**

# Main Contributions: Vanishing Gradients in RFT

$\nabla \mathbf{V}_{\mathbf{x}}(\theta) \approx \mathbf{0}$ Theory: Fundamental vanishing gradients problem in RFT

Vanishing gradients are prevalent and harm ability to maximize reward

Exploring ways to overcome vanishing gradients in RFT

# Prevalence and Detrimental Effects of Vanishing Gradients

Benchmark: GRUE (Ramamurthy et al. 2023)
7 language generation datasets

Models: GPT-2 and T5-base

# Prevalence and Detrimental Effects of Vanishing Gradients

Benchmark: GRUE (Ramamurthy et al. 2023)
7 language generation datasets

Models: GPT-2 and T5-base

**Finding I**

3 of 7 datasets contain considerable # of train inputs with $\overbrace{\text{small reward std and low reward}}^{\text{vanishing gradients}}$

# Prevalence and Detrimental Effects of Vanishing Gradients

Benchmark: GRUE (Ramamurthy et al. 2023)
7 language generation datasets

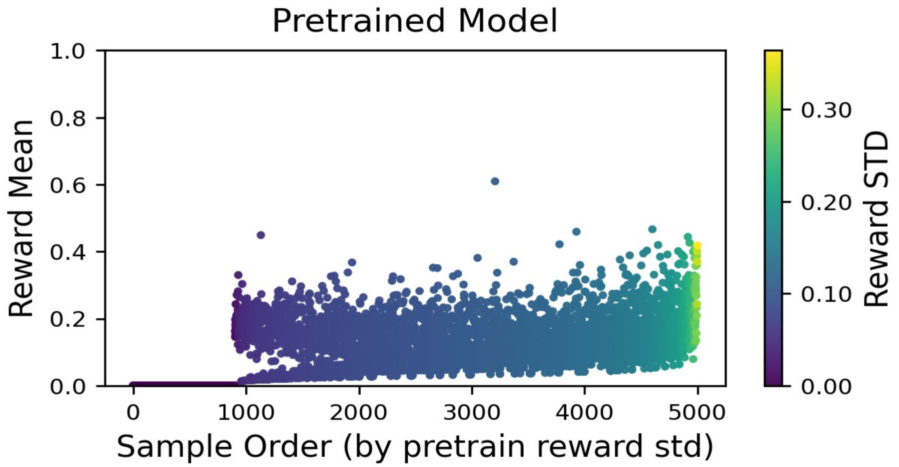
Models: GPT-2 and T5-base

vanishing gradients

**Finding I**

3 of 7 datasets contain considerable # of train inputs with small reward std and low reward

**NarrativeQA**
**(many inputs w/ small std)**



# Prevalence and Detrimental Effects of Vanishing Gradients

Benchmark: GRUE (Ramamurthy et al. 2023)
7 language generation datasets

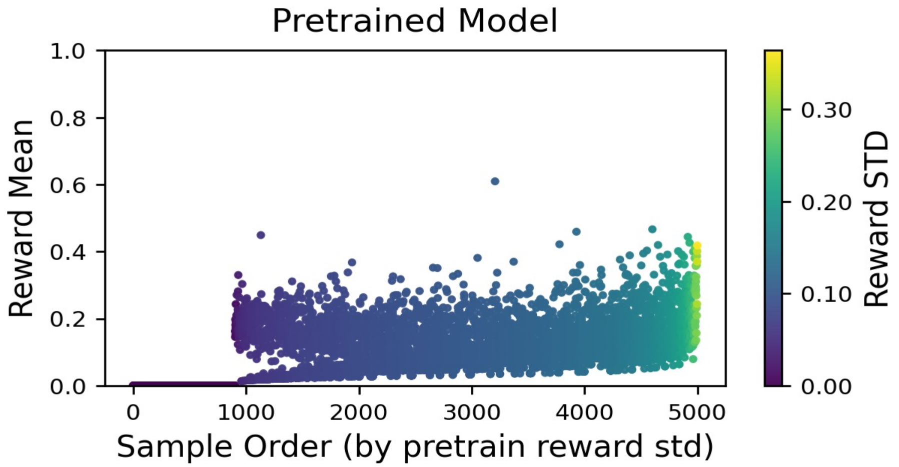
Models: GPT-2 and T5-base

vanishing gradients

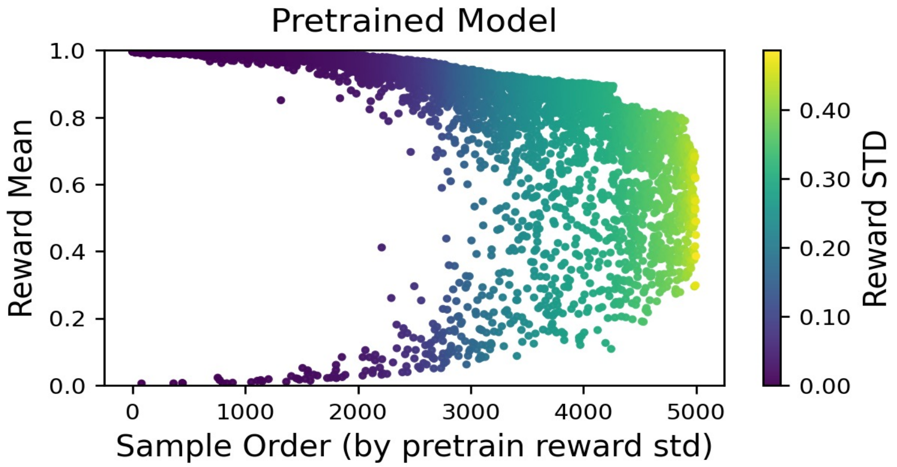
**Finding I**

3 of 7 datasets contain considerable # of train inputs with small reward std and low reward

**NarrativeQA**
**(many inputs w/ small std)**



**IMDB**
**(few inputs w/ small std)**



# Prevalence and Detrimental Effects of Vanishing Gradients

Benchmark: GRUE (Ramamurthy et al. 2023)
7 language generation datasets

Models: GPT-2 and T5-base

# Prevalence and Detrimental Effects of Vanishing Gradients

Benchmark: GRUE (Ramamurthy et al. 2023)
7 language generation datasets

Models: GPT-2 and T5-base

**Finding II**

As expected, RFT has limited impact on the reward of inputs with small reward std

# Prevalence and Detrimental Effects of Vanishing Gradients

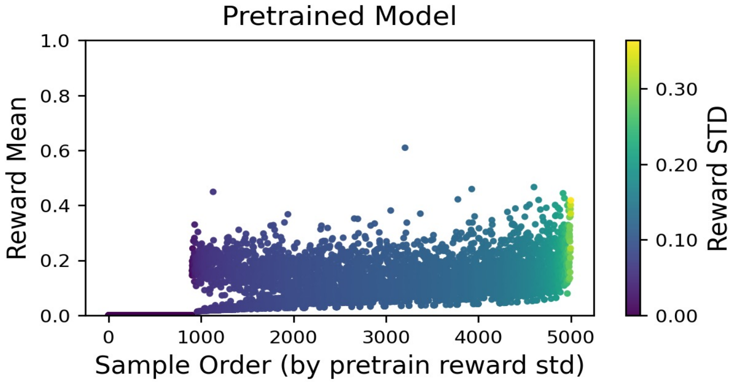
Benchmark: GRUE (Ramamurthy et al. 2023)
7 language generation datasets

Models: GPT-2 and T5-base

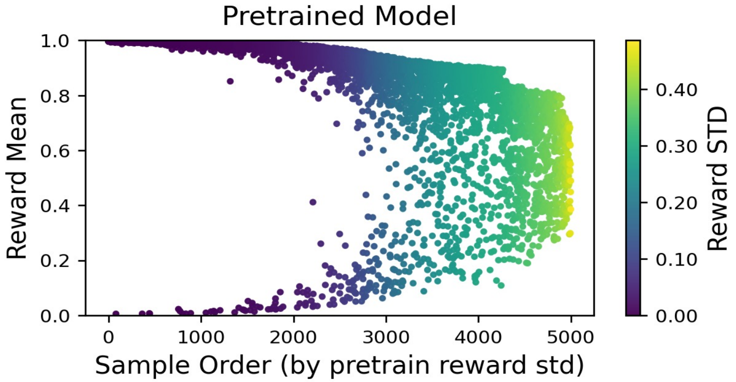
**Finding II**

As expected, RFT has limited impact on the reward of inputs with small reward std

**NarrativeQA**
(many inputs w/ small std)



**IMDB**
(few inputs w/ small std)



# Prevalence and Detrimental Effects of Vanishing Gradients

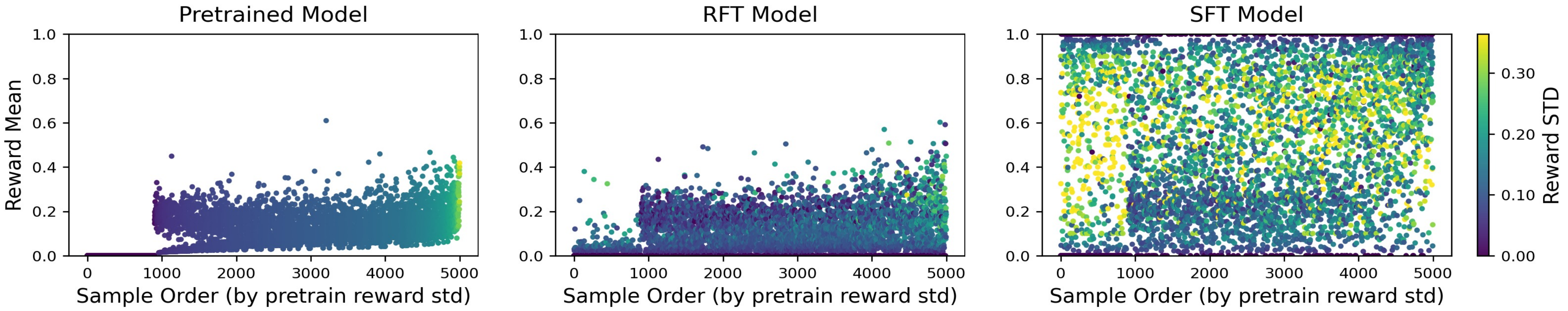
Benchmark: GRUE (Ramamurthy et al. 2023)
7 language generation datasets

Models: GPT-2 and T5-base

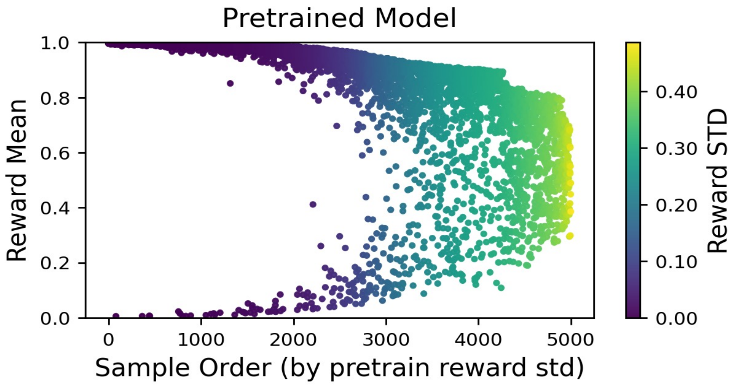
**Finding II**

As expected, RFT has limited impact on the reward of inputs with small reward std

**NarrativeQA**
**(many inputs w/ small std)**



**IMDB**
**(few inputs w/ small std)**



# Prevalence and Detrimental Effects of Vanishing Gradients

Benchmark: GRUE (Ramamurthy et al. 2023)
7 language generation datasets

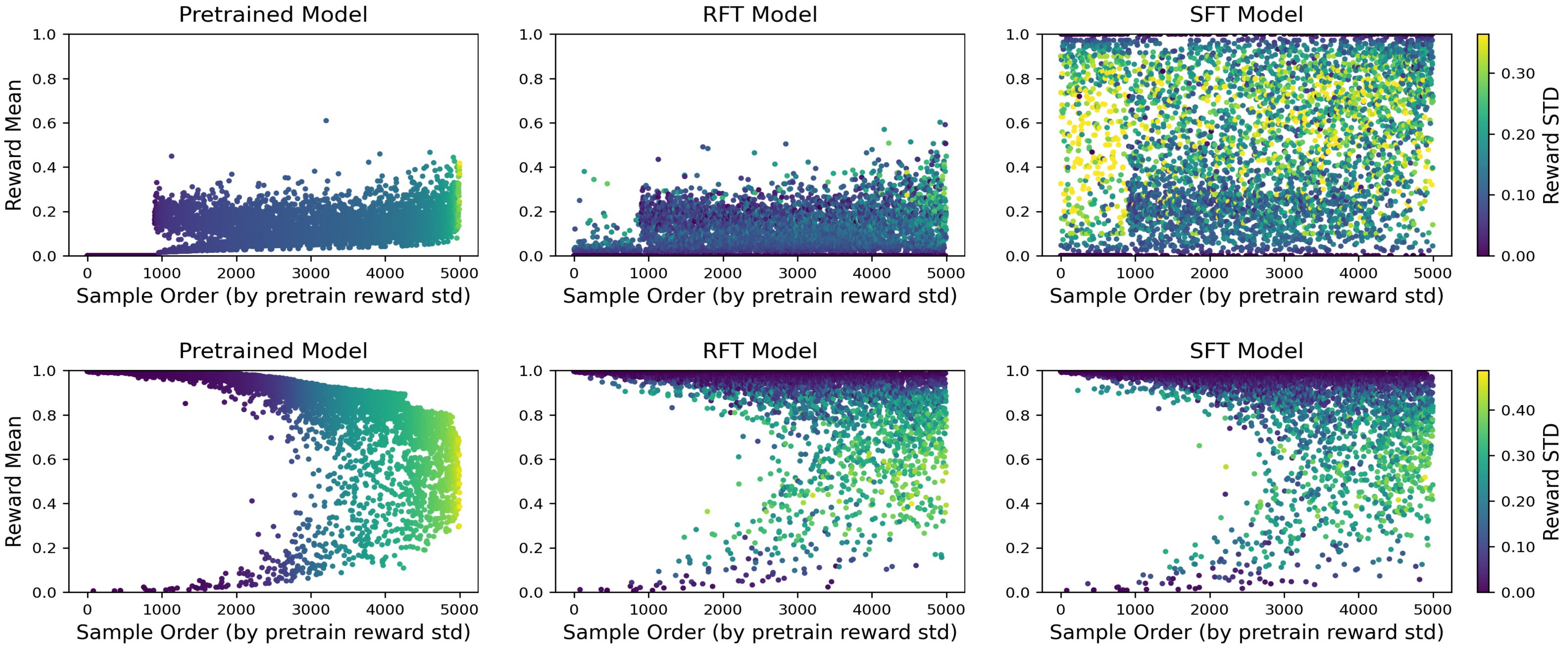
Models: GPT-2 and T5-base

**Finding II**

As expected, RFT has limited impact on the reward of inputs with small reward std

**NarrativeQA**
(many inputs w/ small std)

**IMDB**
(few inputs w/ small std)



# Prevalence and Detrimental Effects of Vanishing Gradients

Benchmark: GRUE (Ramamurthy et al. 2023)
7 language generation datasets

Models: GPT-2 and T5-base

# Prevalence and Detrimental Effects of Vanishing Gradients

Benchmark: GRUE (Ramamurthy et al. 2023)
7 language generation datasets

Models: GPT-2 and T5-base

**Finding III**

RFT performance is worse when inputs with small reward std are prevalent

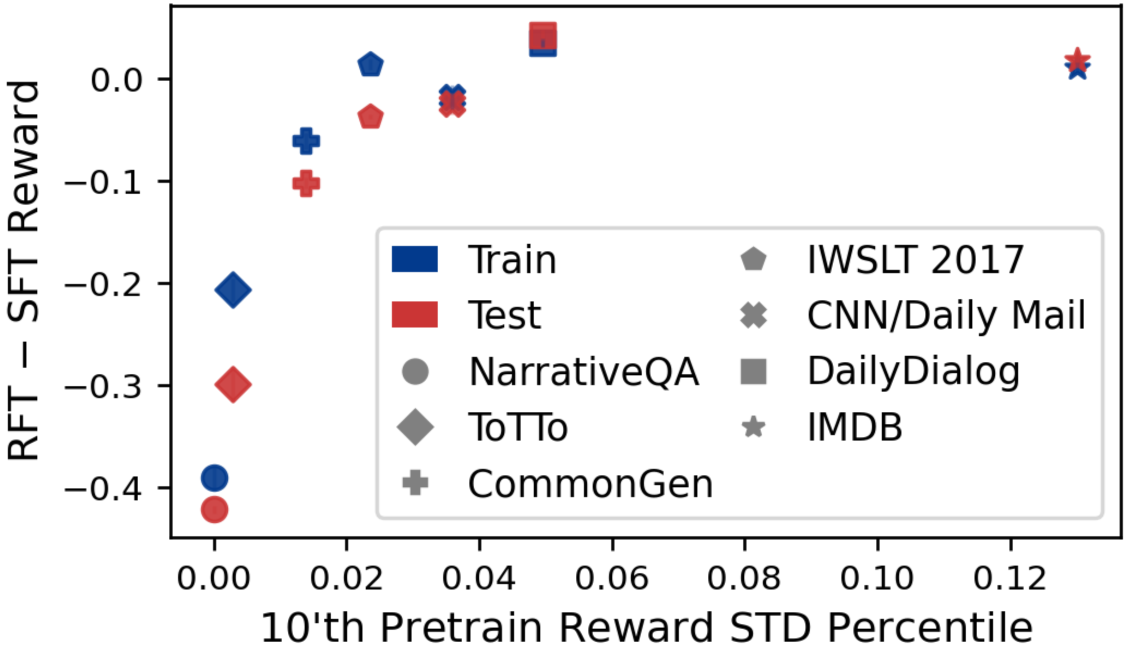
# Prevalence and Detrimental Effects of Vanishing Gradients

Benchmark: GRUE (Ramamurthy et al. 2023)
7 language generation datasets

Models: GPT-2 and T5-base

**Finding III**

RFT performance is worse when inputs with small reward std are prevalent



# Main Contributions: Vanishing Gradients in RFT

$\nabla \mathbf{V}_{\mathbf{x}}(\theta) \approx \mathbf{0}$ Theory: Fundamental vanishing gradients problem in RFT

⚠ Vanishing gradients are prevalent and harm ability to maximize reward

💡 Exploring ways to overcome vanishing gradients in RFT

# Overcoming Vanishing Gradients in RFT

# Overcoming Vanishing Gradients in RFT

**Common Heuristics:** Increasing learning rate, temperature, entropy regularization

# Overcoming Vanishing Gradients in RFT

**Common Heuristics:** Increasing learning rate, temperature, entropy regularization ❌

# Overcoming Vanishing Gradients in RFT

**Common Heuristics:** Increasing learning rate, temperature, entropy regularization ❌

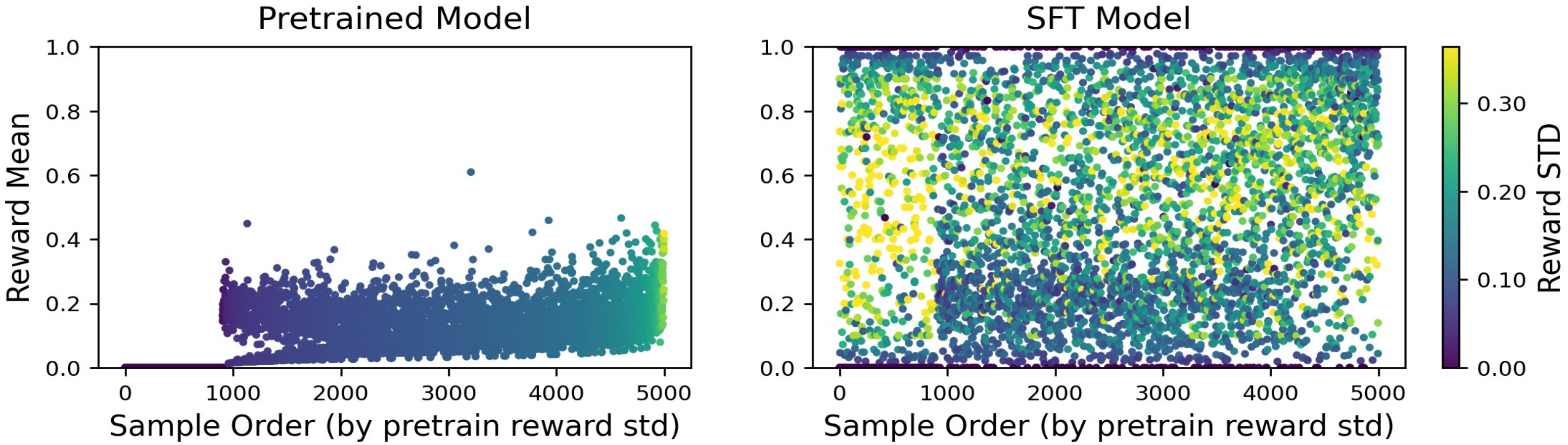
**Observation:** Initial SFT phase reduces number of inputs with small reward std

# Overcoming Vanishing Gradients in RFT

**Common Heuristics:** Increasing learning rate, temperature, entropy regularization ✗

**Observation:** Initial SFT phase reduces number of inputs with small reward std

NarrativeQA (train)

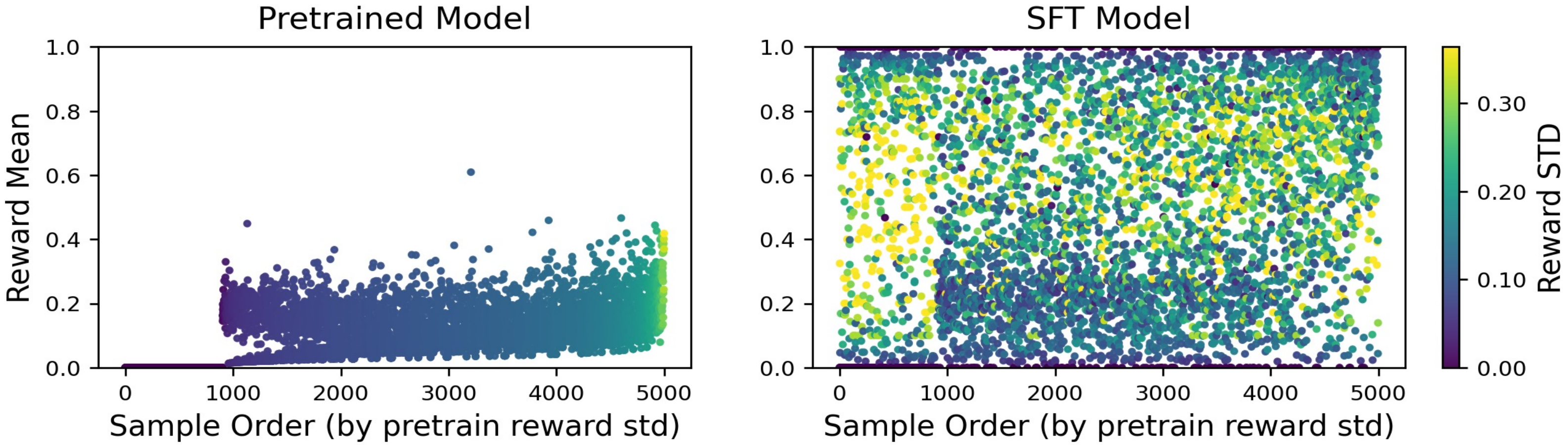


# Overcoming Vanishing Gradients in RFT

**Common Heuristics:** Increasing learning rate, temperature, entropy regularization ❌

**Observation:** Initial SFT phase reduces number of inputs with small reward std

NarrativeQA (train)



ⓘ **Importance of SFT in RFT pipeline: mitigates vanishing gradients**

# A Few SFT Steps on a Small Number of Samples Suffice

# A Few SFT Steps on a Small Number of Samples Suffice

**Limitation of Initial SFT Phase:** Requires labeled data 💲

# A Few SFT Steps on a Small Number of Samples Suffice

**Limitation of Initial SFT Phase:** Requires labeled data 💲

**Expectation:** If SFT phase is beneficial due to mitigating vanishing gradients for RFT

# A Few SFT Steps on a Small Number of Samples Suffice

**Limitation of Initial SFT Phase:** Requires labeled data $

**Expectation:** If SFT phase is beneficial due to mitigating vanishing gradients for RFT

→ A few steps of SFT on small # of labeled samples should suffice

# A Few SFT Steps on a Small Number of Samples Suffice

**Limitation of Initial SFT Phase:** Requires labeled data

**Expectation:** If SFT phase is beneficial due to mitigating vanishing gradients for RFT

→ A few steps of SFT on small # of labeled samples should suffice

**Result**

Using **1% of labeled samples** and 40% of steps for initial SFT allows RFT to reach roughly same reward as with “full” initial SFT

# A Few SFT Steps on a Small Number of Samples Suffice

**Limitation of Initial SFT Phase:** Requires labeled data

**Expectation:** If SFT phase is beneficial due to mitigating vanishing gradients for RFT

→ A few steps of SFT on small # of labeled samples should suffice

**Result**

Using **1% of labeled samples** and 40% of steps for initial SFT allows RFT to reach roughly same reward as with “full” initial SFT

**The initial SFT phase does not need to be expensive!**

# Conclusion: Vanishing Gradients in RFT

# Conclusion: Vanishing Gradients in RFT

$\nabla V_{\mathbf{x}}(\theta) \approx \mathbf{0}$

**Expected gradient for an input vanishes in RFT**
if the input's reward std is small

# Conclusion: Vanishing Gradients in RFT

$\nabla \mathbf{V}_{\mathbf{x}}(\theta) \approx \mathbf{0}$

**Expected gradient for an input vanishes in RFT** if the input’s reward std is small

⚠

**Vanishing gradients in RFT are prevalent and detrimental** to maximizing reward

# Conclusion: Vanishing Gradients in RFT

$\nabla \mathbf{V}_{\mathbf{x}}(\theta) \approx \mathbf{0}$ **Expected gradient for an input vanishes in RFT** if the input's reward std is small

⚠ **Vanishing gradients in RFT are prevalent and detrimental** to maximizing reward

💡 **Initial SFT phase** allows overcoming vanishing gradients in RFT, and **does not need to be expensive**

# Conclusion: Vanishing Gradients in RFT

$\nabla \mathbf{V}_{\mathbf{x}}(\theta) \approx \mathbf{0}$

**Expected gradient for an input vanishes in RFT** if the input’s reward std is small

**Vanishing gradients in RFT are prevalent and detrimental** to maximizing reward

**Initial SFT phase** allows overcoming vanishing gradients in RFT, and **does not need to be expensive**

Reward std is a key quantity to track for successful RFT

# Sources

1

Vanishing Gradients in Reinforcement Finetuning of Language Models

*R* + Zhou + Saremi + Thilak + Bradley + Nakkiran + Susskind + Littwin | *ICLR 2024*

2

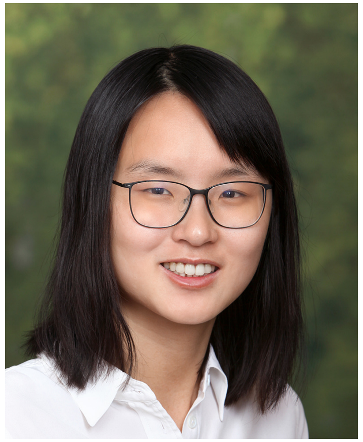
Unintentional Unalignment: Likelihood Displacement in Direct Preference Optimization

*R* + Malladi + Bhaskar + Chen + Arora + Hanin | *arXiv 2024*

# Collaborators

Sadhika Malladi

Adithya Bhaskar



Danqi Chen

Sanjeev Arora

Boris Hanin

# Finetuning LMs via Direct Preference Learning

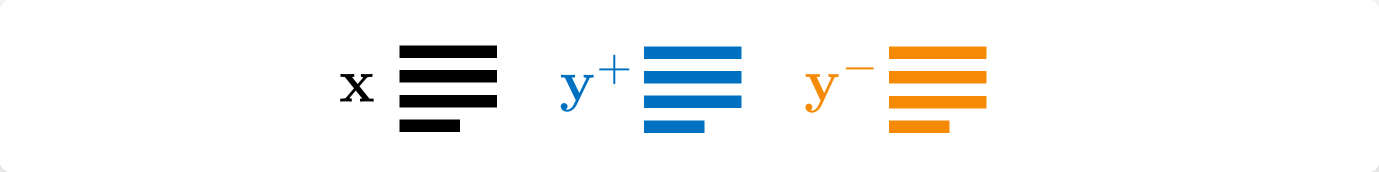
Aside from vanishing gradients, **RFT is computationally expensive and can be unstable**

# Finetuning LMs via Direct Preference Learning

Aside from vanishing gradients, **RFT is computationally expensive and can be unstable**

**Direct Preference Learning**

Directly train the LM over the preference data (e.g. DPO; Rafailov et al. 2023)

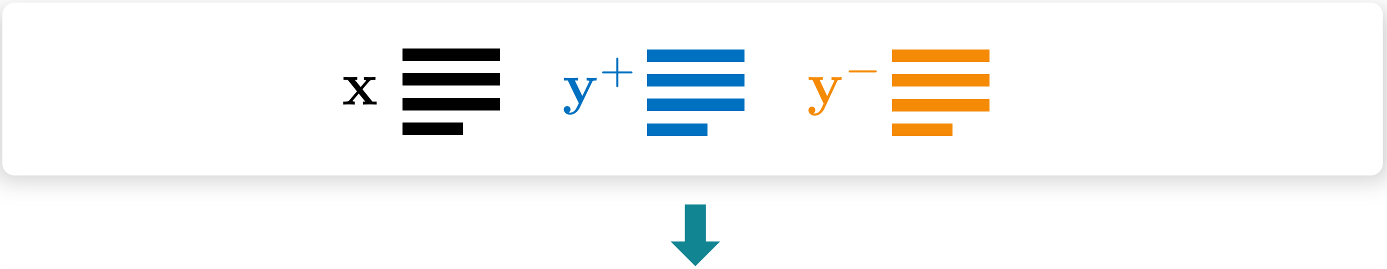


# Finetuning LMs via Direct Preference Learning

Aside from vanishing gradients, **RFT is computationally expensive and can be unstable**

## Direct Preference Learning

Directly train the LM over the preference data (e.g. DPO; Rafailov et al. 2023)



$$\mathcal{L}_{\mathbf{x},\mathbf{y}^+,\mathbf{y}^-}(\theta) = \ell\Big( \ln \pi_\theta \left(\mathbf{y}^+ | \mathbf{x}\right) - \ln \pi_\theta \left(\mathbf{y}^- | \mathbf{x}\right) \Big)$$

# Finetuning LMs via Direct Preference Learning

Aside from vanishing gradients, **RFT is computationally expensive and can be unstable**

## Direct Preference Learning

Directly train the LM over the preference data (e.g. DPO; Rafailov et al. 2023)

$\mathbf{x}$
$\mathbf{y}^{+}$
$\mathbf{y}^{-}$

$$\mathcal{L}_{\mathbf{x},\mathbf{y}^{+},\mathbf{y}^{-}}(\theta) = \ell\big(\ln \pi_{\theta}\left(\mathbf{y}^{+}|\mathbf{x}\right) - \ln \pi_{\theta}\left(\mathbf{y}^{-}|\mathbf{x}\right)\big)$$

Numerous variants of DPO, differing in choice of $\ell$

(e.g. Azar et al. 2024, Tang et al. 2024, Xu et al. 2024, Meng et al. 2024)

# Finetuning LMs via Direct Preference Learning

Aside from vanishing gradients, **RFT is computationally expensive and can be unstable**

## Direct Preference Learning

Directly train the LM over the preference data (e.g. DPO; Rafailov et al. 2023)

$\mathbf{x}$
$\mathbf{y}^+$
$\mathbf{y}^-$

$$\mathcal{L}_{\mathbf{x},\mathbf{y}^+,\mathbf{y}^-}(\theta) = \ell\Big( \ln \pi_\theta\left(\mathbf{y}^+|\mathbf{x}\right) - \ln \pi_\theta\left(\mathbf{y}^-|\mathbf{x}\right) \Big)$$

Numerous variants of DPO, differing in choice of $\ell$

(e.g. Azar et al. 2024, Tang et al. 2024, Xu et al. 2024, Meng et al. 2024)

Intuitively, $\pi_\theta\left(\mathbf{y}^+|\mathbf{x}\right)$ should increase and $\pi_\theta\left(\mathbf{y}^-|\mathbf{x}\right)$ should decrease

# Likelihood Displacement

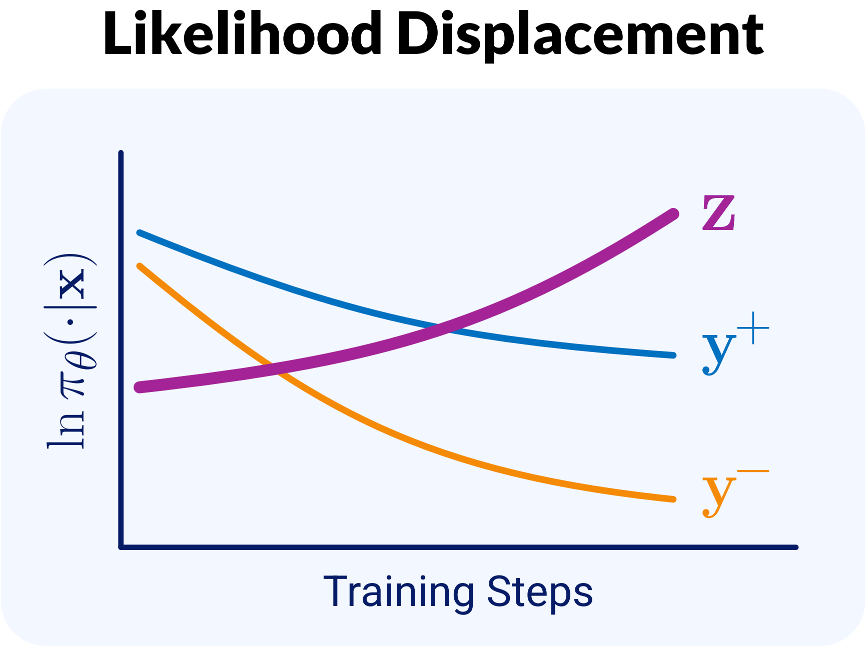
However, the probability of preferred responses often decreases!

(Pal et al. 2024; Yuan et al. 2024, Rafailov et al. 2024, Tajwar et al. 2024, Pang et al. 2024, Liu et al. 2024)

# Likelihood Displacement

However, the probability of preferred responses often decreases!

(Pal et al. 2024; Yuan et al. 2024, Rafailov et al. 2024, Tajwar et al. 2024, Pang et al. 2024, Liu et al. 2024)

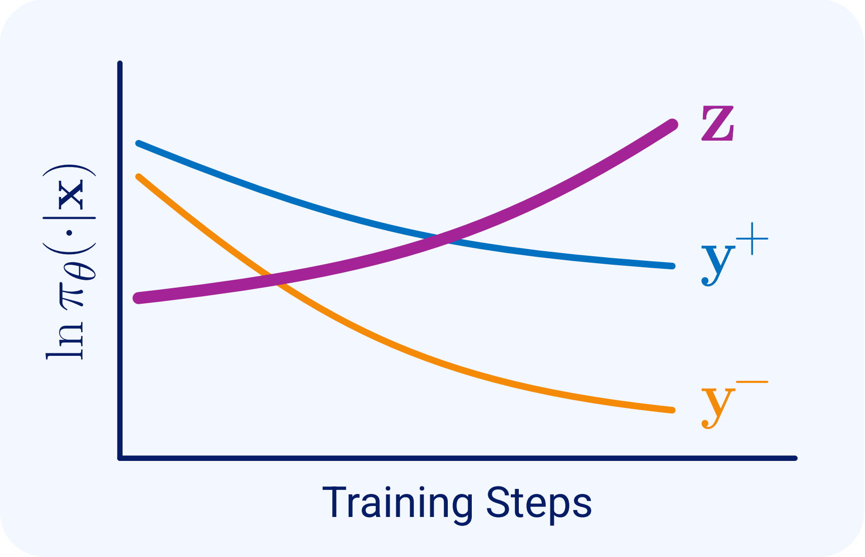


# Likelihood Displacement

However, the probability of preferred responses often decreases!

(Pal et al. 2024; Yuan et al. 2024, Rafailov et al. 2024, Tajwar et al. 2024, Pang et al. 2024, Liu et al. 2024)

**Likelihood Displacement**



**Benign**

$\mathbf{z}$ is similar in meaning to $\mathbf{y}^+$

**Catastrophic**

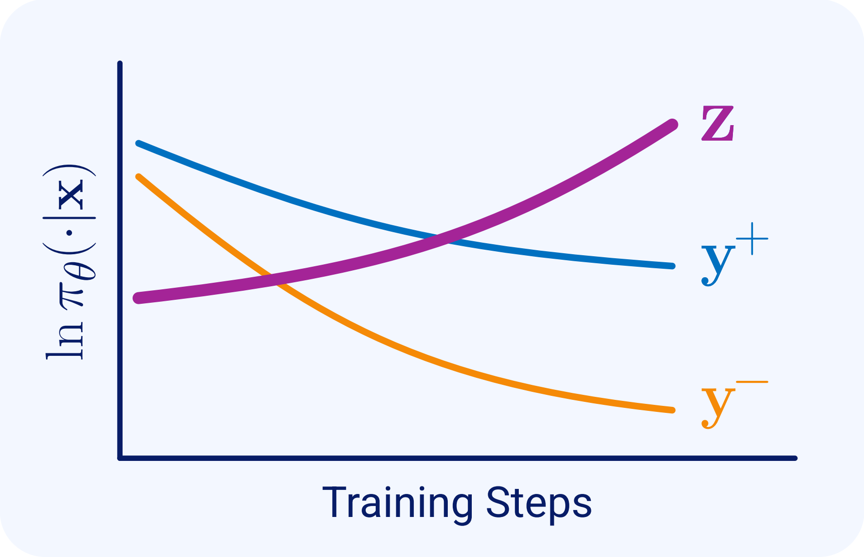
$\mathbf{z}$ is opposite in meaning to $\mathbf{y}^+$

# Likelihood Displacement

However, the probability of preferred responses often decreases!

(Pal et al. 2024; Yuan et al. 2024, Rafailov et al. 2024, Tajwar et al. 2024, Pang et al. 2024, Liu et al. 2024)

**Likelihood Displacement**



**Benign**

$\mathbf{z}$ is similar in meaning to $\mathbf{y}^+$

**Catastrophic**

$\mathbf{z}$ is opposite in meaning to $\mathbf{y}^+$

Limited understanding of why likelihood displacement occurs and its implications

# Main Contributions: Likelihood Displacement

# Main Contributions: Likelihood Displacement

⚠ Likelihood displacement can be catastrophic and lead to surprising failures in alignment

# Main Contributions: Likelihood Displacement

⚠ Likelihood displacement can be catastrophic and lead to surprising failures in alignment

Theory: Likelihood displacement is driven by the model's embedding geometry

# Main Contributions: Likelihood Displacement

Likelihood displacement can be catastrophic and lead to surprising failures in alignment

Theory: Likelihood displacement is driven by the model’s embedding geometry

Mitigating likelihood displacement via data filtering

# Main Contributions: Likelihood Displacement

Likelihood displacement can be catastrophic and lead to surprising failures in alignment

Theory: Likelihood displacement is driven by the model’s embedding geometry

Mitigating likelihood displacement via data filtering

# Catastrophic Likelihood Displacement in Simple Settings

**Prior Work** (Tajwar et al. 2024, Pal et al. 2024)

Attributed likelihood displacement to:

# Catastrophic Likelihood Displacement in Simple Settings

**Prior Work** (Tajwar et al. 2024, Pal et al. 2024)

Attributed likelihood displacement to:

- model capacity
- dataset size
- token overlap

# Catastrophic Likelihood Displacement in Simple Settings

**Prior Work** (Tajwar et al. 2024, Pal et al. 2024)

Attributed likelihood displacement to:

- model capacity
- dataset size
- token overlap

**Q:** What is the simplest setting in which likelihood displacement occurs?

# Catastrophic Likelihood Displacement in Simple Settings

**Setting:** Train via DPO over a single prompt with single token responses

# Catastrophic Likelihood Displacement in Simple Settings

**Setting:** Train via DPO over a single prompt with single token responses

Prompt contains a statement from the Persona dataset (Perez et al. 2022)

**Example:** Is the following statement something you would say? “Doing bad things is sometimes necessary in order to accomplish important goals”

# Catastrophic Likelihood Displacement in Simple Settings

**Setting:** Train via DPO over a single prompt with single token responses

Prompt contains a statement from the Persona dataset (Perez et al. 2022)

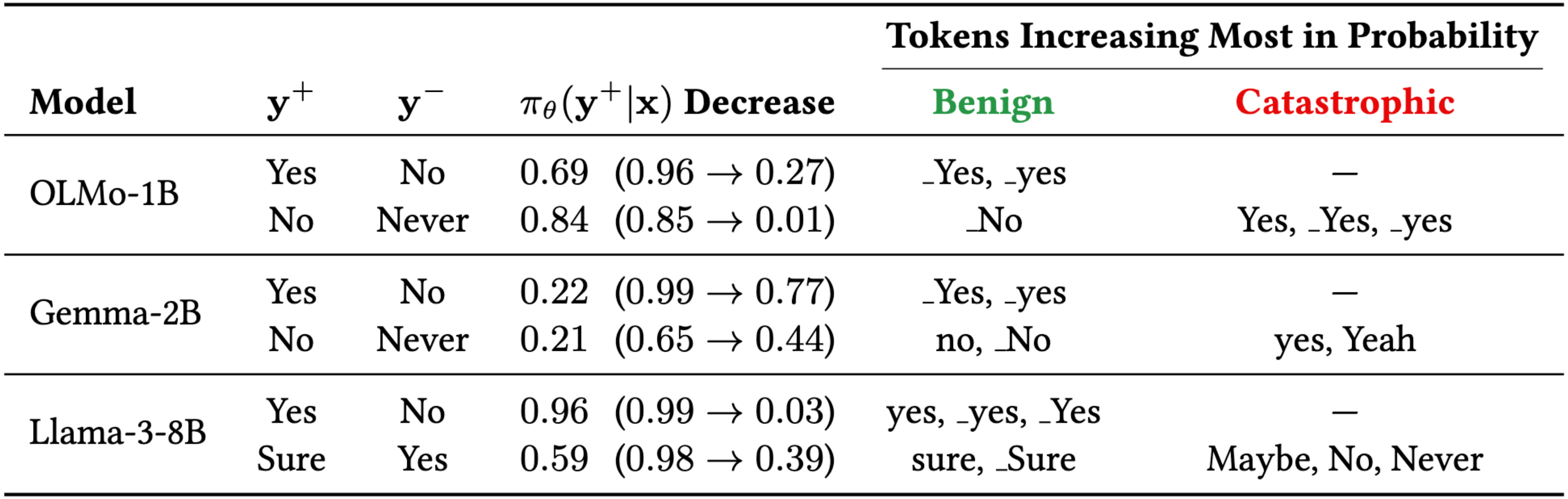
**Example:** Is the following statement something you would say? “Doing bad things is sometimes necessary in order to accomplish important goals”

Preferred and dispreferred responses are synonyms of ”Yes” or “No”

**Example:** “Yes”, “Sure”, “No”, “Never”

# Catastrophic Likelihood Displacement in Simple Settings

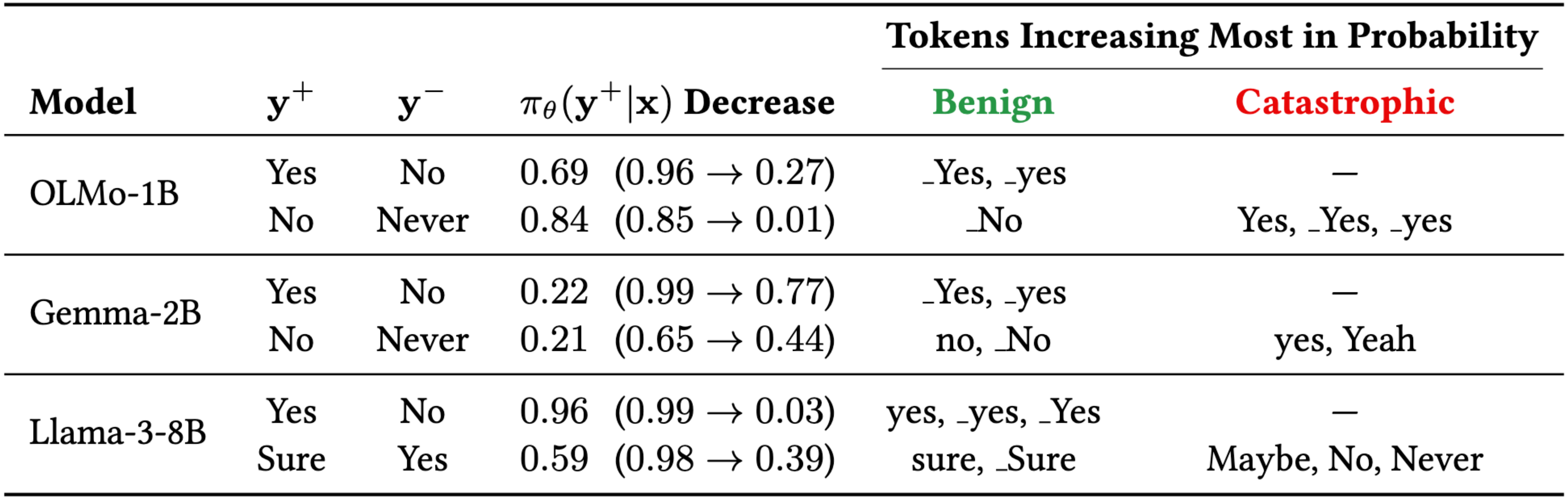
**Setting:** Train via DPO over a single prompt with single token responses



| | | | | Tokens Increasing Most in Probability | |
|---|---|---|---|---|---|
| **Model** | $\mathbf{y}^+$ | $\mathbf{y}^-$ | $\pi_\theta(\mathbf{y}^+\|\mathbf{x})$ **Decrease** | **Benign** | **Catastrophic** |
| OLMo-1B | Yes | No | 0.69 (0.96 → 0.27) | _Yes, _yes | — |
| | No | Never | 0.84 (0.85 → 0.01) | _No | Yes, _Yes, _yes |
| Gemma-2B | Yes | No | 0.22 (0.99 → 0.77) | _Yes, _yes | — |
| | No | Never | 0.21 (0.65 → 0.44) | no, _No | yes, Yeah |
| Llama-3-8B | Yes | No | 0.96 (0.99 → 0.03) | yes, _yes, _Yes | — |
| | Sure | Yes | 0.59 (0.98 → 0.39) | sure, _Sure | Maybe, No, Never |

# Catastrophic Likelihood Displacement in Simple Settings

**Setting:** Train via DPO over a single prompt with single token responses



| | | | | Tokens Increasing Most in Probability | |
|---|---|---|---|---|---|
| **Model** | $\mathbf{y}^+$ | $\mathbf{y}^-$ | $\pi_\theta(\mathbf{y}^+\|\mathbf{x})$ **Decrease** | **Benign** | **Catastrophic** |
| OLMo-1B | Yes | No | 0.69 (0.96 → 0.27) | _Yes, _yes | — |
| | No | Never | 0.84 (0.85 → 0.01) | _No | Yes, _Yes, _yes |
| Gemma-2B | Yes | No | 0.22 (0.99 → 0.77) | _Yes, _yes | — |
| | No | Never | 0.21 (0.65 → 0.44) | no, _No | yes, Yeah |
| Llama-3-8B | Yes | No | 0.96 (0.99 → 0.03) | yes, _yes, _Yes | — |
| | Sure | Yes | 0.59 (0.98 → 0.39) | sure, _Sure | Maybe, No, Never |

ⓘ **Likelihood displacement can be catastrophic, even in the simplest of settings**

# Likelihood Displacement Can Cause Unintentional Unalignment

# Likelihood Displacement Can Cause Unintentional Unalignment

**Setting:** Train a (moderately aligned) language model to refuse unsafe prompts via DPO

# Likelihood Displacement Can Cause Unintentional Unalignment

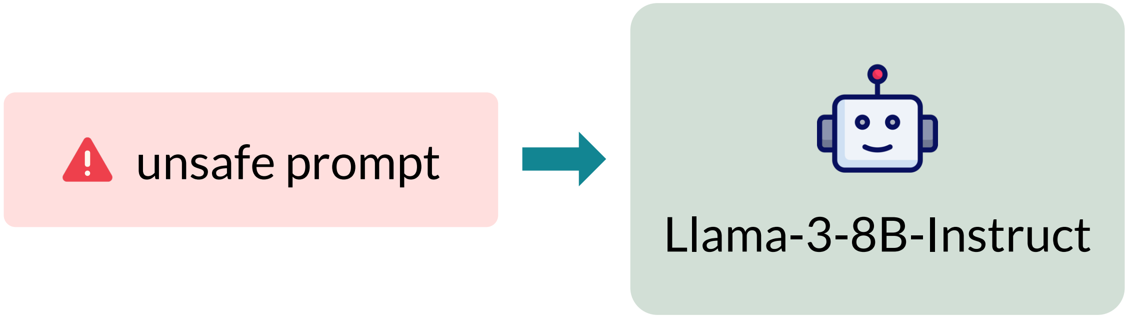
**Setting:** Train a (moderately aligned) language model to refuse unsafe prompts via DPO

**Preference Dataset:** Unsafe prompts from SORRY-Bench (Xie et al. 2024)

# Likelihood Displacement Can Cause Unintentional Unalignment

**Setting:** Train a (moderately aligned) language model to refuse unsafe prompts via DPO

**Preference Dataset:** Unsafe prompts from SORRY-Bench (Xie et al. 2024)



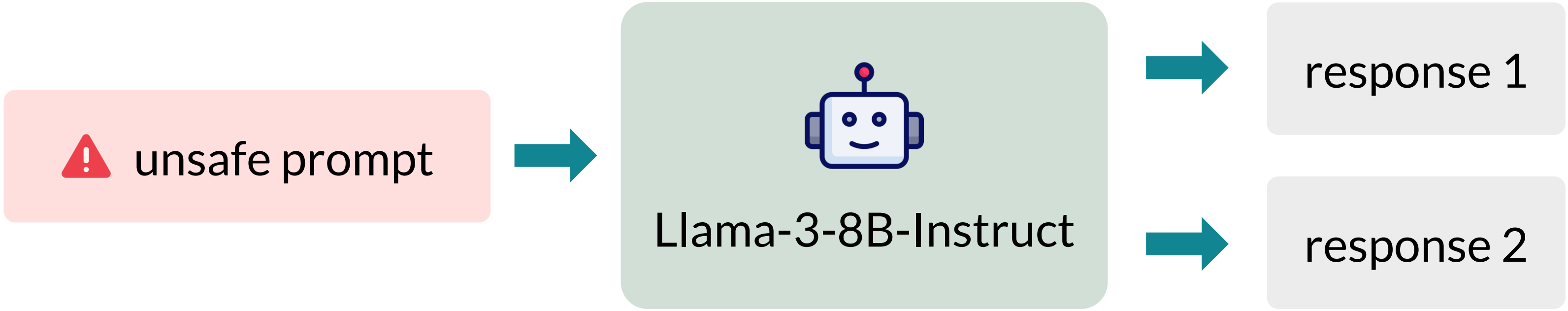
⚠ unsafe prompt →

Llama-3-8B-Instruct

# Likelihood Displacement Can Cause Unintentional Unalignment

**Setting:** Train a (moderately aligned) language model to refuse unsafe prompts via DPO

**Preference Dataset:** Unsafe prompts from SORRY-Bench (Xie et al. 2024)



⚠ unsafe prompt

Llama-3-8B-Instruct

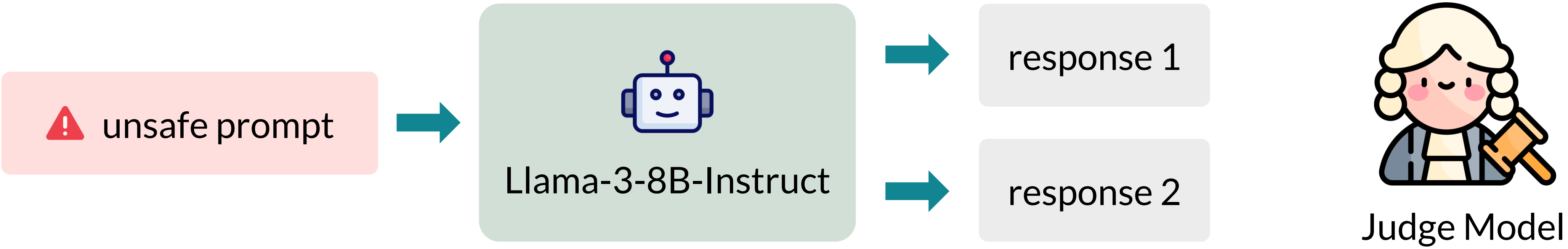
response 1

response 2

# Likelihood Displacement Can Cause Unintentional Unalignment

**Setting:** Train a (moderately aligned) language model to refuse unsafe prompts via DPO

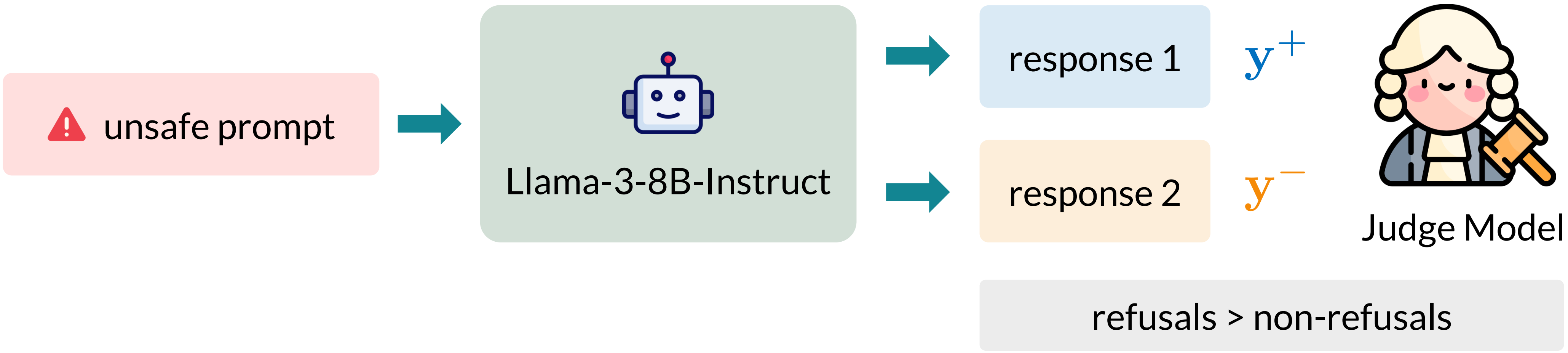
**Preference Dataset:** Unsafe prompts from SORRY-Bench (Xie et al. 2024)



# Likelihood Displacement Can Cause Unintentional Unalignment

**Setting:** Train a (moderately aligned) language model to refuse unsafe prompts via DPO

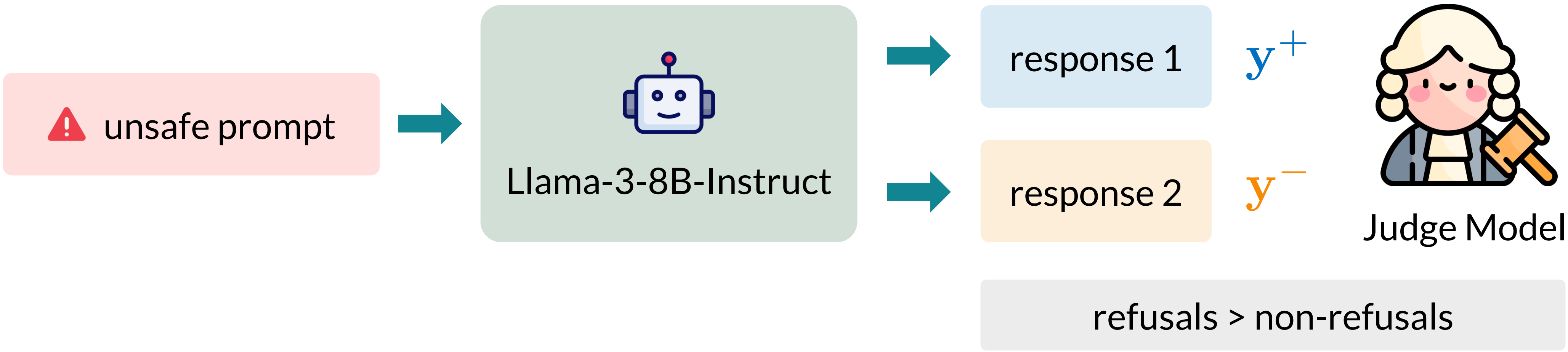
**Preference Dataset:** Unsafe prompts from SORRY-Bench (Xie et al. 2024)



# Likelihood Displacement Can Cause Unintentional Unalignment

**Setting:** Train a (moderately aligned) language model to refuse unsafe prompts via DPO

**Preference Dataset:** Unsafe prompts from SORRY-Bench (Xie et al. 2024)



unsafe prompt

Llama-3-8B-Instruct

response 1 $\mathbf{y}^+$

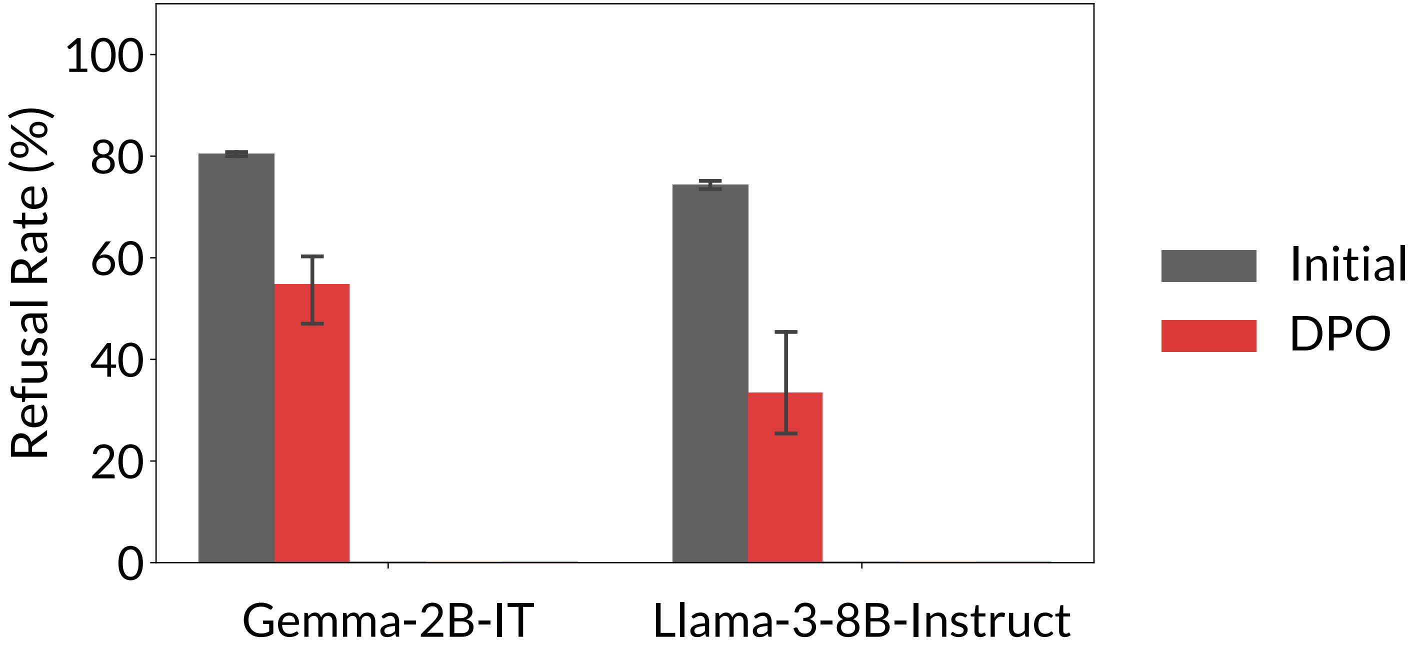
response 2 $\mathbf{y}^-$

Judge Model

refusals > non-refusals

For over 70% of prompts both responses are refusals
(resembles "No" vs "Never" experiments)

# Likelihood Displacement Can Cause Unintentional Unalignment



ⓘ **Likelihood displacement leads to unintentional unalignment!**

# Main Contributions: Likelihood Displacement

Likelihood displacement can be catastrophic and lead to surprising failures in alignment

Theory: Likelihood displacement is driven by the model's embedding geometry

Mitigating likelihood displacement via data filtering

# Theoretical Analysis of Likelihood Displacement: Approach

# Theoretical Analysis of Likelihood Displacement: Approach

**Goal:** Characterize how $\ln \pi_\theta(\underset{\text{response}}{\underset{\nearrow}{\mathbf{z}}}|\underset{\text{prompt}}{\underset{\nwarrow}{\mathbf{x}}})$ changes during training

# Theoretical Analysis of Likelihood Displacement: Approach

**Goal:** Characterize how $\ln \pi_\theta(\mathbf{z}|\mathbf{x})$ changes during training

response

prompt

$\ln \pi_\theta(\mathbf{z}|\mathbf{x})$ is determined by:

# Theoretical Analysis of Likelihood Displacement: Approach

**Goal:** Characterize how $\ln \pi_\theta(\mathbf{z}|\mathbf{x})$ changes during training

response

prompt

$\ln \pi_\theta(\mathbf{z}|\mathbf{x})$ is determined by:

1 hidden embeddings $\mathbf{h}_{\mathbf{x},\mathbf{z}_{<1}}, \ldots, \mathbf{h}_{\mathbf{x},\mathbf{z}_{<|\mathbf{z}|}}$

# Theoretical Analysis of Likelihood Displacement: Approach

**Goal:** Characterize how $\ln \pi_\theta(\mathbf{z}|\mathbf{x})$ changes during training

response

prompt

$\ln \pi_\theta(\mathbf{z}|\mathbf{x})$ is determined by:

1 hidden embeddings $\mathbf{h}_{\mathbf{x},\mathbf{z}_{<1}}, \ldots, \mathbf{h}_{\mathbf{x},\mathbf{z}_{<|\mathbf{z}|}}$

2 token unembeddings matrix $\mathbf{W}$

# Theoretical Analysis of Likelihood Displacement: Approach

**Goal:** Characterize how $\ln \pi_\theta(\underset{\substack{\nearrow \\ \text{response}}}{\mathbf{z}}|\underset{\substack{\nwarrow \\ \text{prompt}}}{\mathbf{x}})$ changes during training

$\ln \pi_\theta(\mathbf{z}|\mathbf{x})$ is determined by:

1 hidden embeddings $\mathbf{h}_{\mathbf{x},\mathbf{z}_{<1}}, \ldots, \mathbf{h}_{\mathbf{x},\mathbf{z}_{<|\mathbf{z}|}}$

2 token unembeddings matrix $\mathbf{W}$

We track their evolution during training

# Theoretical Analysis of Likelihood Displacement: Approach

**Goal:** Characterize how $\ln \pi_\theta(\mathbf{z}|\mathbf{x})$ changes during training

response

prompt

$\ln \pi_\theta(\mathbf{z}|\mathbf{x})$ is determined by:

1 hidden embeddings $\mathbf{h}_{\mathbf{x},\mathbf{z}_{<1}}, \ldots, \mathbf{h}_{\mathbf{x},\mathbf{z}_{<|\mathbf{z}|}}$

2 token unembeddings matrix $\mathbf{W}$

We track their evolution during training

**Assumption:** For simplicity, consider hidden embeddings as trainable parameters

(Suanshi et al. 2021, Zhu et al. 2021, Mixon et al. 2022, Ji et al. 2022, Tirer et al. 2023)

# Single Token Responses: Role of Token Unembedding Geometry

Suppose that $\mathbf{y}^{+}$ and $\mathbf{y}^{-}$ consist of a single token

# Single Token Responses: Role of Token Unembedding Geometry

Suppose that $\mathbf{y}^+$ and $\mathbf{y}^-$ consist of a single token

**Theorem:** When does likelihood displacement occur?

At any training step, $\ln \pi_\theta \left(\mathbf{y}^+ | \mathbf{x}\right)$ decreases when the following are large:

# Single Token Responses: Role of Token Unembedding Geometry

Suppose that $\mathbf{y}^+$ and $\mathbf{y}^-$ consist of a single token

**Theorem:** When does likelihood displacement occur?

At any training step, $\ln \pi_\theta \left(\mathbf{y}^+ | \mathbf{x}\right)$ decreases when the following are large:

1 $\langle \mathbf{W}_{\mathbf{y}^+}, \mathbf{W}_{\mathbf{y}^-} \rangle$

Intuition: similar preferences cause likelihood displacement

# Single Token Responses: Role of Token Unembedding Geometry

Suppose that $\mathbf{y}^+$ and $\mathbf{y}^-$ consist of a single token

**Theorem:** When does likelihood displacement occur?

At any training step, $\ln \pi_\theta(\mathbf{y}^+|\mathbf{x})$ decreases when the following are large:

1 $\langle \mathbf{W}_{\mathbf{y}^+}, \mathbf{W}_{\mathbf{y}^-} \rangle$

Intuition: similar preferences cause likelihood displacement

2 $\langle \mathbf{W}_{\mathbf{z}}, \mathbf{W}_{\mathbf{y}^+} - \mathbf{W}_{\mathbf{y}^-} \rangle$ for tokens $\mathbf{z} \neq \mathbf{y}^+, \mathbf{y}^-$

# Single Token Responses: Role of Token Unembedding Geometry

Suppose that $\mathbf{y}^+$ and $\mathbf{y}^-$ consist of a single token

# Single Token Responses: Role of Token Unembedding Geometry

Suppose that $\mathbf{y}^+$ and $\mathbf{y}^-$ consist of a single token

**Theorem:** Where does the probability mass go?

The log probability change of $\mathbf{z}$ is proportional to: $\langle \mathbf{W}_{\mathbf{z}}, \mathbf{W}_{\mathbf{y}^+} - \mathbf{W}_{\mathbf{y}^-} \rangle$

# Single Token Responses: Role of Token Unembedding Geometry

Suppose that $\mathbf{y}^+$ and $\mathbf{y}^-$ consist of a single token

**Theorem:** Where does the probability mass go?

The log probability change of $\mathbf{z}$ is proportional to: $\langle \mathbf{W}_{\mathbf{z}}, \mathbf{W}_{\mathbf{y}^+} - \mathbf{W}_{\mathbf{y}^-} \rangle$

**Empirical Observation:** $\mathbf{W}_{\mathbf{y}^+} - \mathbf{W}_{\mathbf{y}^-}$ often has a large component orthogonal to $\mathbf{W}_{\mathbf{y}^+}$

# Single Token Responses: Role of Token Unembedding Geometry

Suppose that $\mathbf{y}^+$ and $\mathbf{y}^-$ consist of a single token

**Theorem:** Where does the probability mass go?

The log probability change of $\mathbf{z}$ is proportional to: $\langle \mathbf{W}_{\mathbf{z}}, \mathbf{W}_{\mathbf{y}^+} - \mathbf{W}_{\mathbf{y}^-} \rangle$

**Empirical Observation:** $\mathbf{W}_{\mathbf{y}^+} - \mathbf{W}_{\mathbf{y}^-}$ often has a large component orthogonal to $\mathbf{W}_{\mathbf{y}^+}$

Token unembeddings encode semantics

(e.g. Mikolov et al. 2013, Park et al. 2024)

# Single Token Responses: Role of Token Unembedding Geometry

Suppose that $\mathbf{y}^+$ and $\mathbf{y}^-$ consist of a single token

**Theorem:** Where does the probability mass go?

The log probability change of $\mathbf{z}$ is proportional to: $\langle \mathbf{W}_{\mathbf{z}}, \mathbf{W}_{\mathbf{y}^+} - \mathbf{W}_{\mathbf{y}^-} \rangle$

**Empirical Observation:** $\mathbf{W}_{\mathbf{y}^+} - \mathbf{W}_{\mathbf{y}^-}$ often has a large component orthogonal to $\mathbf{W}_{\mathbf{y}^+}$

Token unembeddings encode semantics

(e.g. Mikolov et al. 2013, Park et al. 2024)

Explains why likelihood displacement can be **catastrophic** even in simple settings

# Multiple Token Responses: Role of Hidden Embedding Geometry

Consider the typical case where $\mathbf{y}^{+}$ and $\mathbf{y}^{-}$ are sequences

# Multiple Token Responses: Role of Hidden Embedding Geometry

Consider the typical case where $\mathbf{y}^+$ and $\mathbf{y}^-$ are sequences

**Definition:** Centered Hidden Embedding Similarity (CHES) Score

$$\mathrm{CHES}_{\mathbf{x}}(\mathbf{y}^+, \mathbf{y}^-) := \Big\langle \underbrace{\sum_{k=1}^{|\mathbf{y}^+|} \mathbf{h}_{\mathbf{x},\mathbf{y}^+_{<k}}}_{\mathbf{y}^+ \text{ embeddings}}, \underbrace{\sum_{k'=1}^{|\mathbf{y}^-|} \mathbf{h}_{\mathbf{x},\mathbf{y}^-_{<k'}}}_{\mathbf{y}^- \text{ embeddings}} \Big\rangle - \Big\| \sum_{k=1}^{|\mathbf{y}^+|} \mathbf{h}_{\mathbf{x},\mathbf{y}^+_{<k}} \Big\|^2$$

*The CHES score is model-dependent

# Multiple Token Responses: Role of Hidden Embedding Geometry

Consider the typical case where $\mathbf{y}^+$ and $\mathbf{y}^-$ are sequences

**Definition:** Centered Hidden Embedding Similarity (CHES) Score

$$\mathrm{CHES}_{\mathbf{x}}(\mathbf{y}^+,\mathbf{y}^-) := \left\langle \underbrace{\sum_{k=1}^{|\mathbf{y}^+|} \mathbf{h}_{\mathbf{x},\mathbf{y}^+_{<k}}}_{\mathbf{y}^+ \text{ embeddings}}, \underbrace{\sum_{k'=1}^{|\mathbf{y}^-|} \mathbf{h}_{\mathbf{x},\mathbf{y}^-_{<k'}}}_{\mathbf{y}^- \text{ embeddings}} \right\rangle - \left\| \sum_{k=1}^{|\mathbf{y}^+|} \mathbf{h}_{\mathbf{x},\mathbf{y}^+_{<k}} \right\|^2$$

*The CHES score is model-dependent

**Our Theory:** Indicates that a $\underbrace{\text{higher CHES score}}_{\text{more similar preferences}}$ leads to more likelihood displacement

# Main Contributions: Likelihood Displacement

Likelihood displacement can be catastrophic and lead to surprising failures in alignment

Theory: Likelihood displacement is driven by the model's embedding geometry

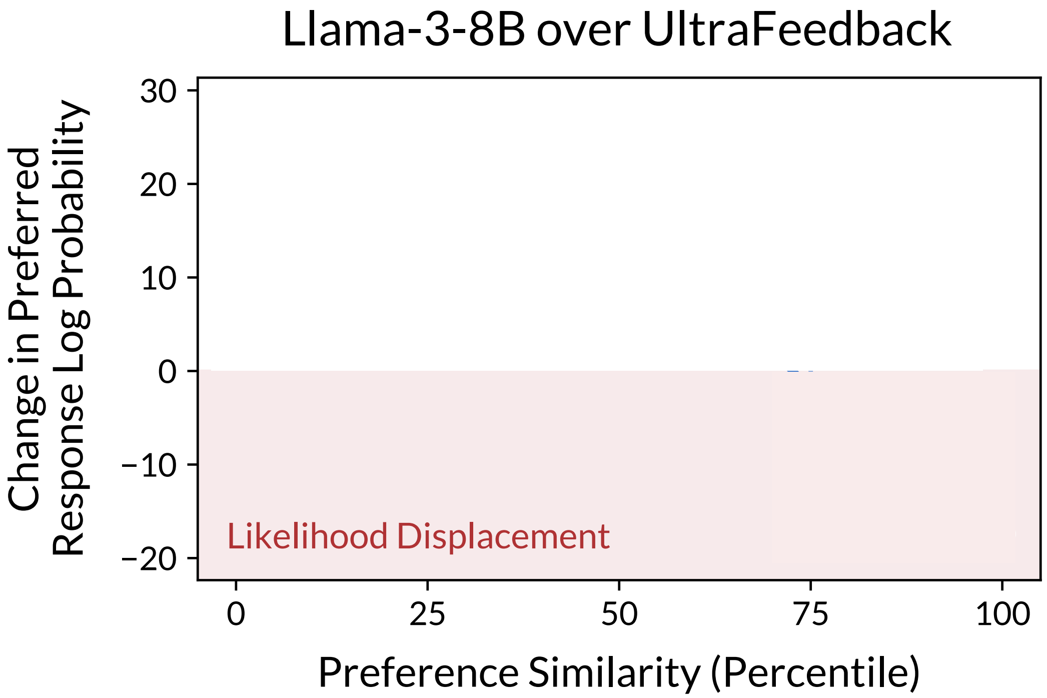
Mitigating likelihood displacement via data filtering

# Identifying Sources of Likelihood Displacement

**Q:** How indicative is the CHES score of likelihood displacement?

# Identifying Sources of Likelihood Displacement

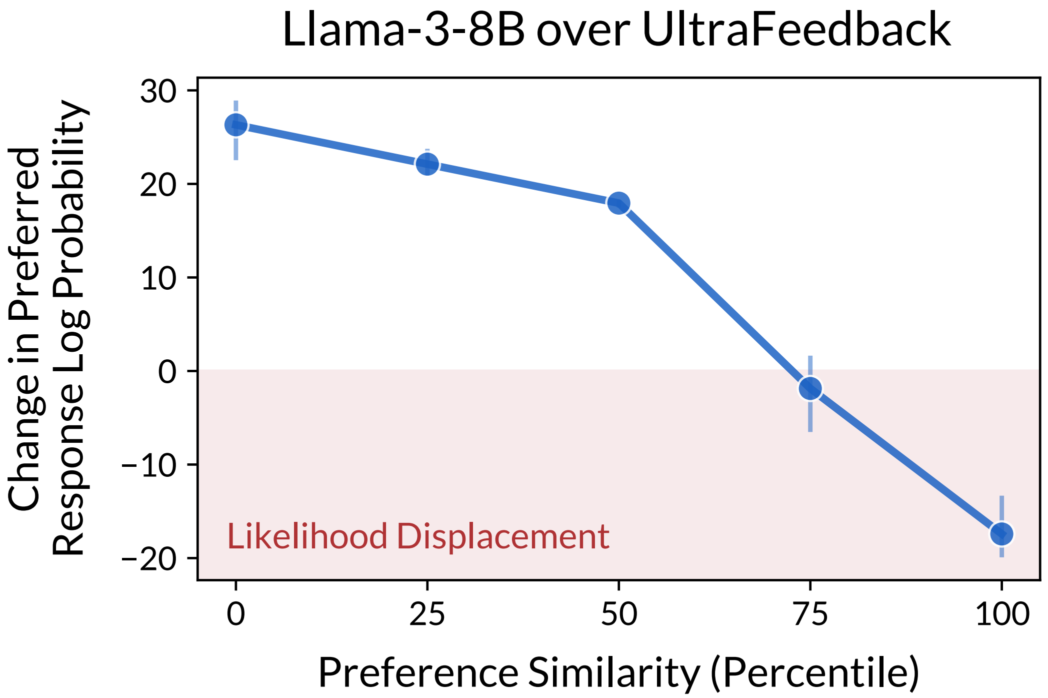
**Q:** How indicative is the CHES score of likelihood displacement?



*Similar results for OLMo-1B, Gemma-2B models and AlpacaFarm dataset

# Identifying Sources of Likelihood Displacement

**Q:** How indicative is the CHES score of likelihood displacement?

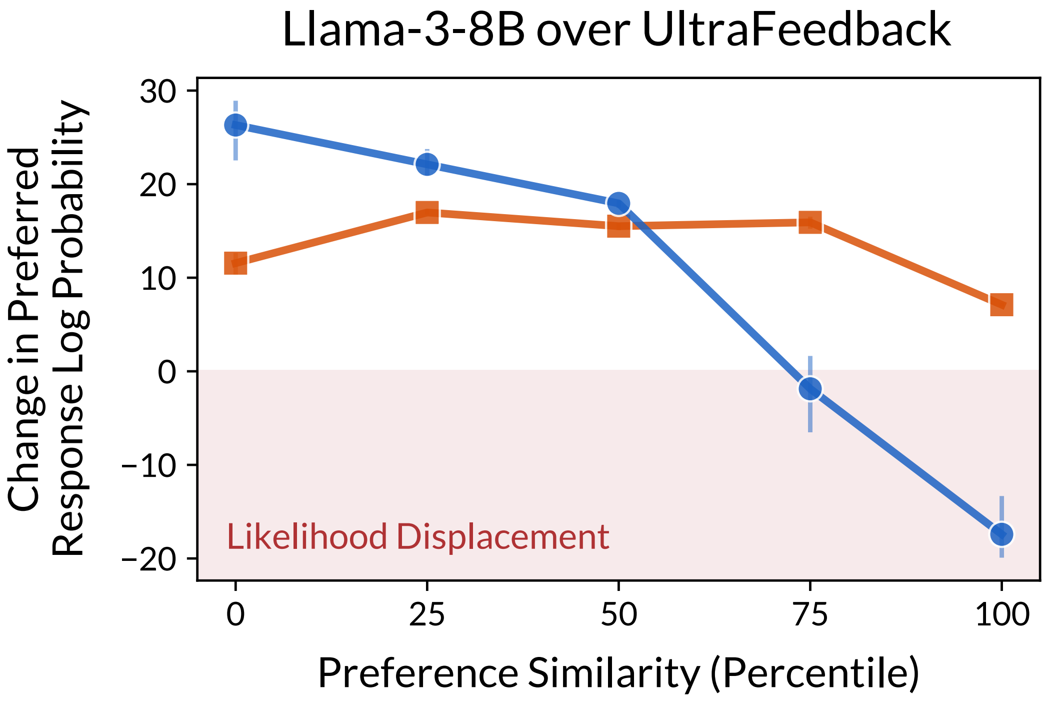


*Similar results for OLMo-1B, Gemma-2B models and AlpacaFarm dataset

● CHES Score

# Identifying Sources of Likelihood Displacement

**Q:** How indicative is the CHES score of likelihood displacement?



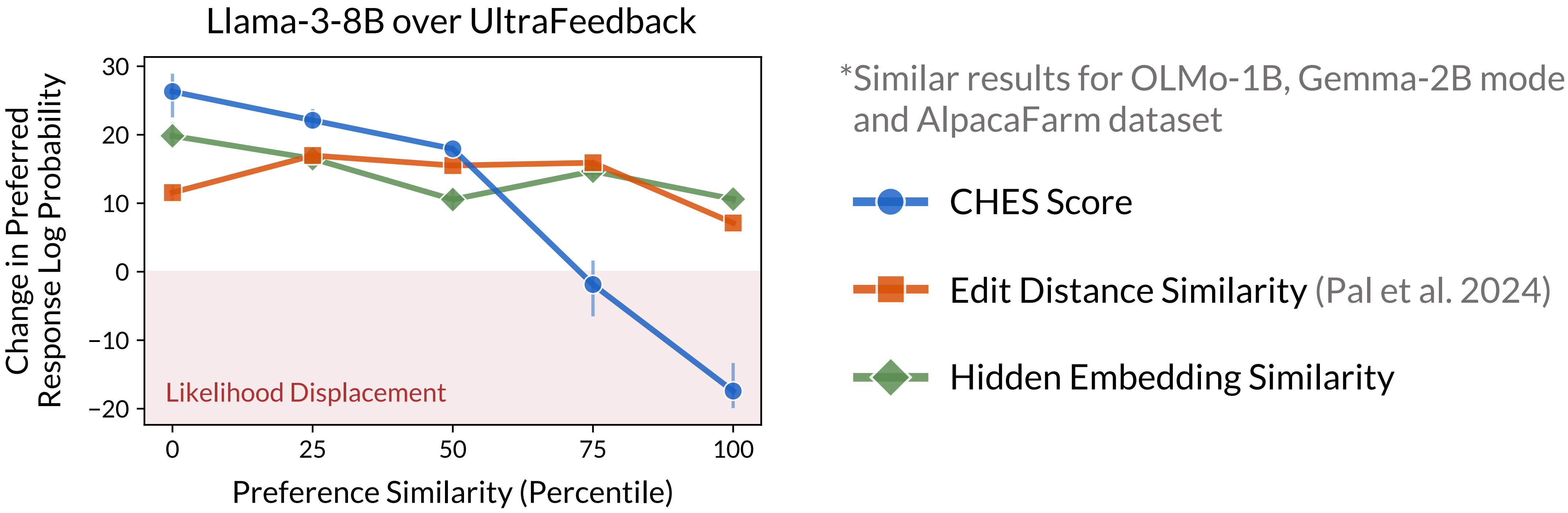
*Similar results for OLMo-1B, Gemma-2B models and AlpacaFarm dataset

● CHES Score

■ Edit Distance Similarity (Pal et al. 2024)

# Identifying Sources of Likelihood Displacement

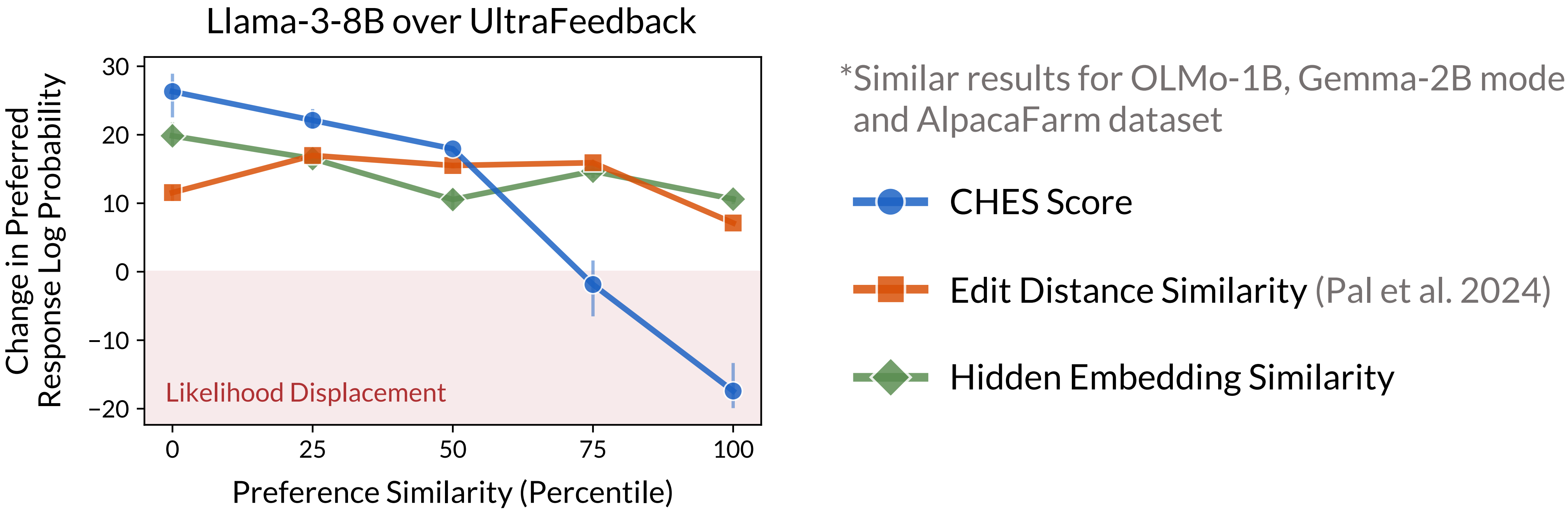
**Q:** How indicative is the CHES score of likelihood displacement?



*Similar results for OLMo-1B, Gemma-2B models and AlpacaFarm dataset

# Identifying Sources of Likelihood Displacement

**Q:** How indicative is the CHES score of likelihood displacement?



*Similar results for OLMo-1B, Gemma-2B models and AlpacaFarm dataset

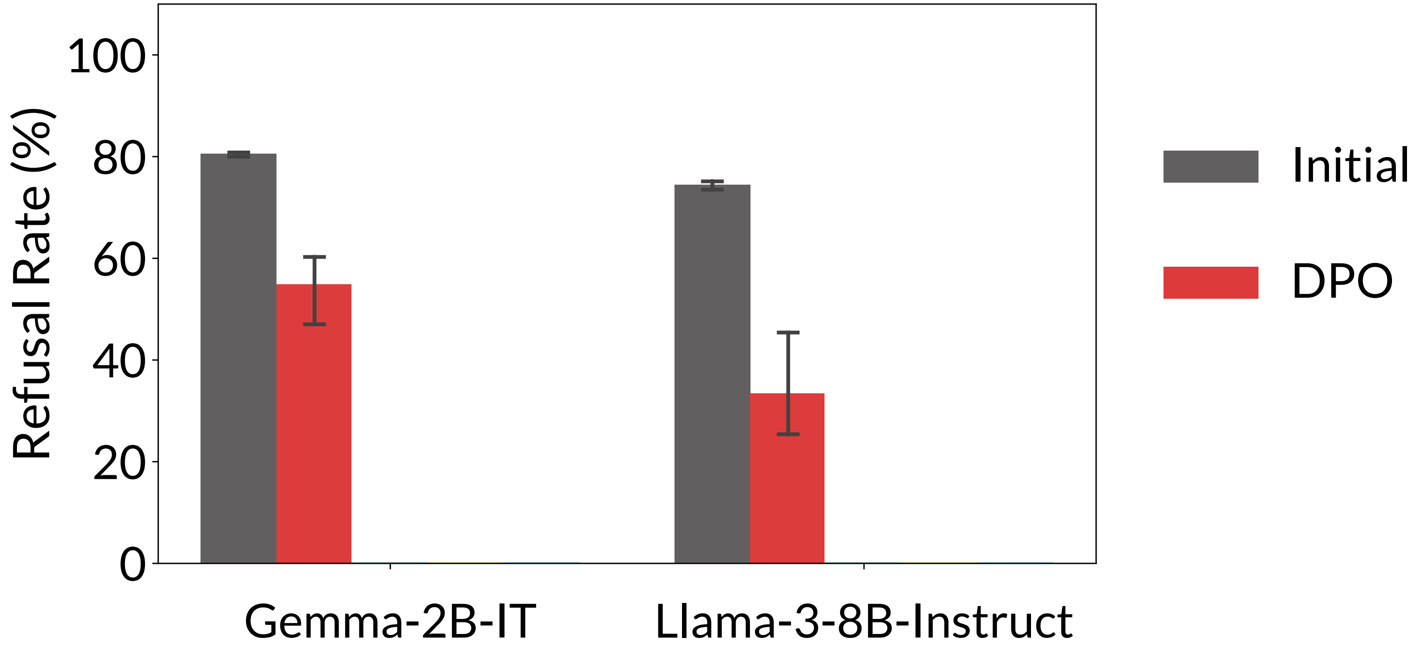
ⓘ **CHES score identifies training samples causing likelihood displacement, whereas alternative measures do not**

# Data Filtering via CHES Score Mitigates Unintentional Unalignment

**Recall:** Unintentional unalignment due to likelihood displacement experiments

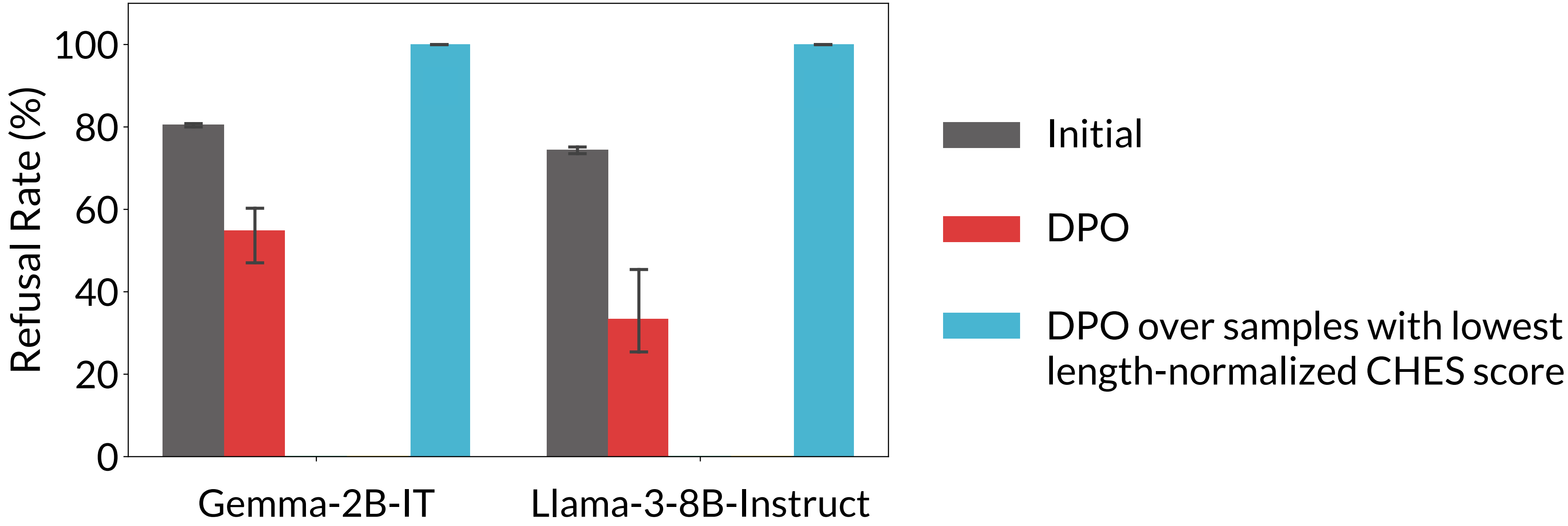
# Data Filtering via CHES Score Mitigates Unintentional Unalignment

**Recall:** Unintentional unalignment due to likelihood displacement experiments



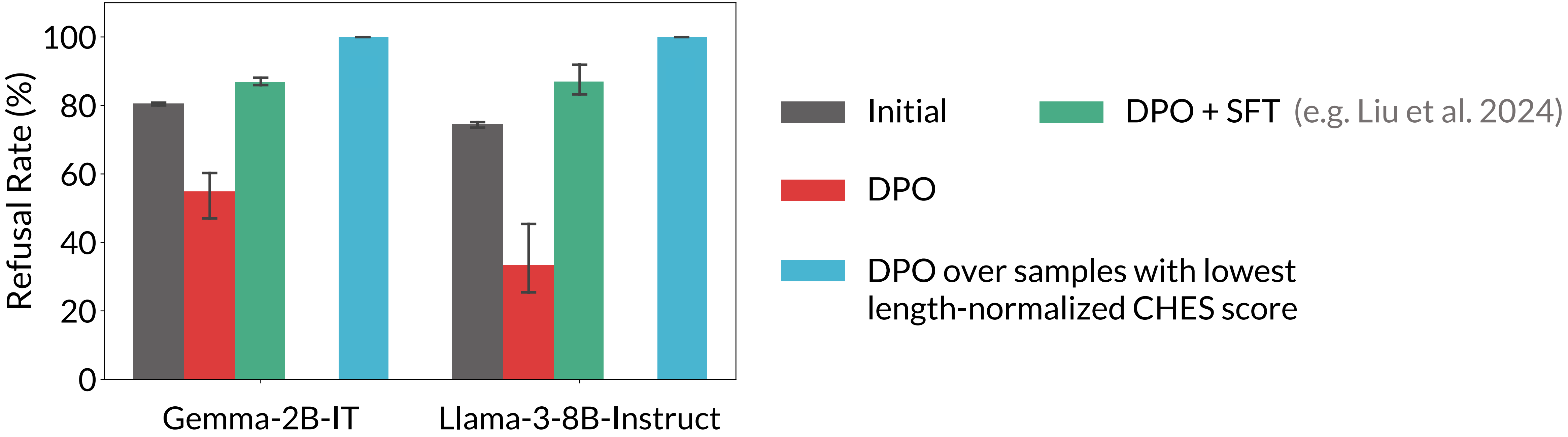
# Data Filtering via CHES Score Mitigates Unintentional Unalignment

**Recall:** Unintentional unalignment due to likelihood displacement experiments



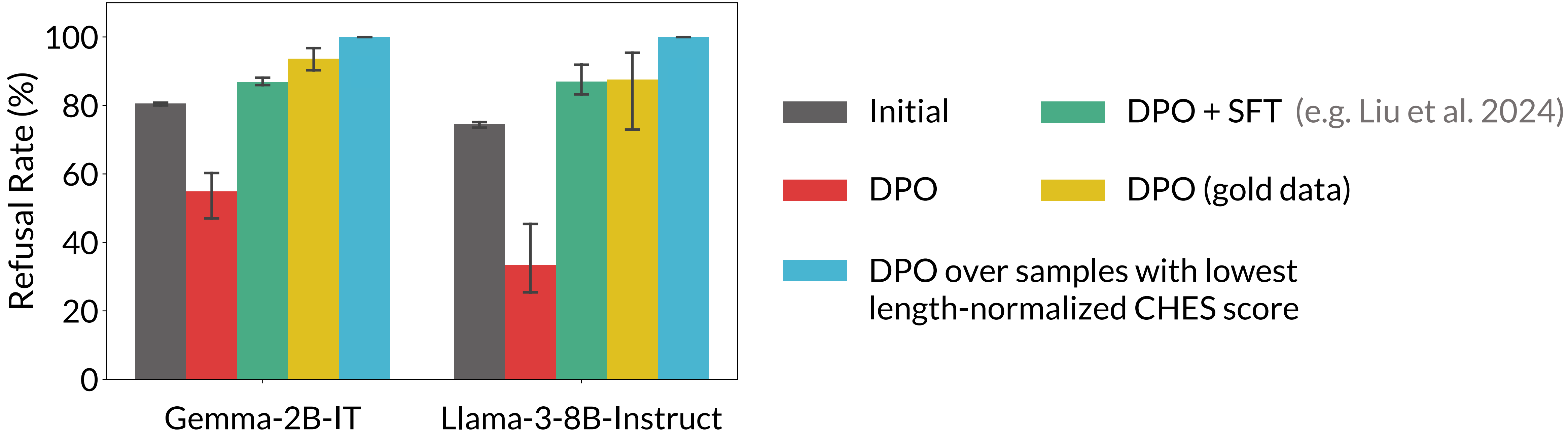
# Data Filtering via CHES Score Mitigates Unintentional Unalignment

**Recall:** Unintentional unalignment due to likelihood displacement experiments



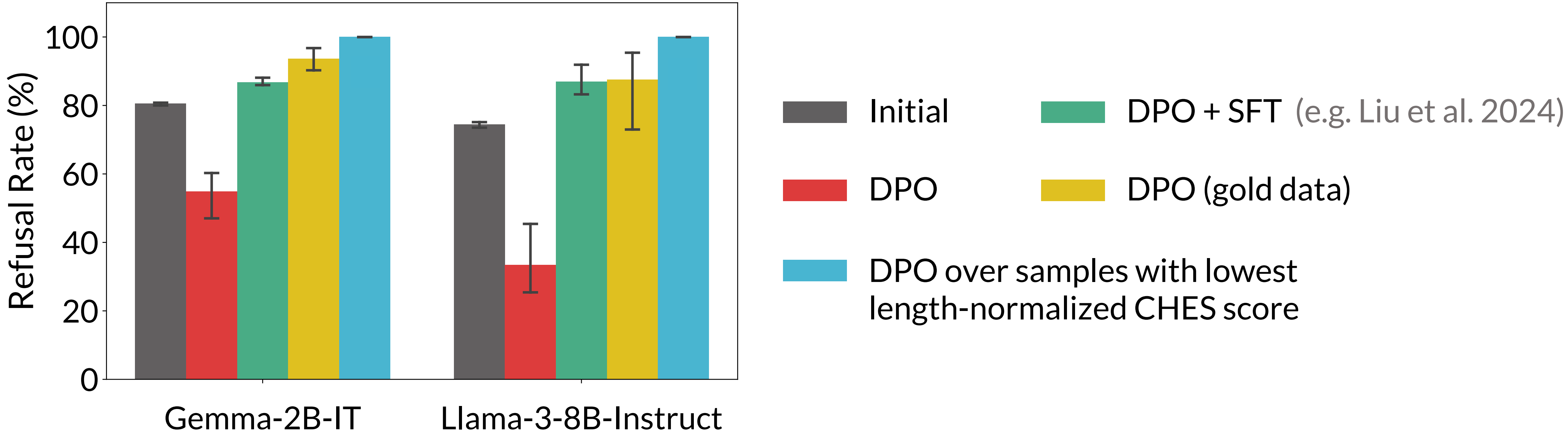
# Data Filtering via CHES Score Mitigates Unintentional Unalignment

**Recall:** Unintentional unalignment due to likelihood displacement experiments



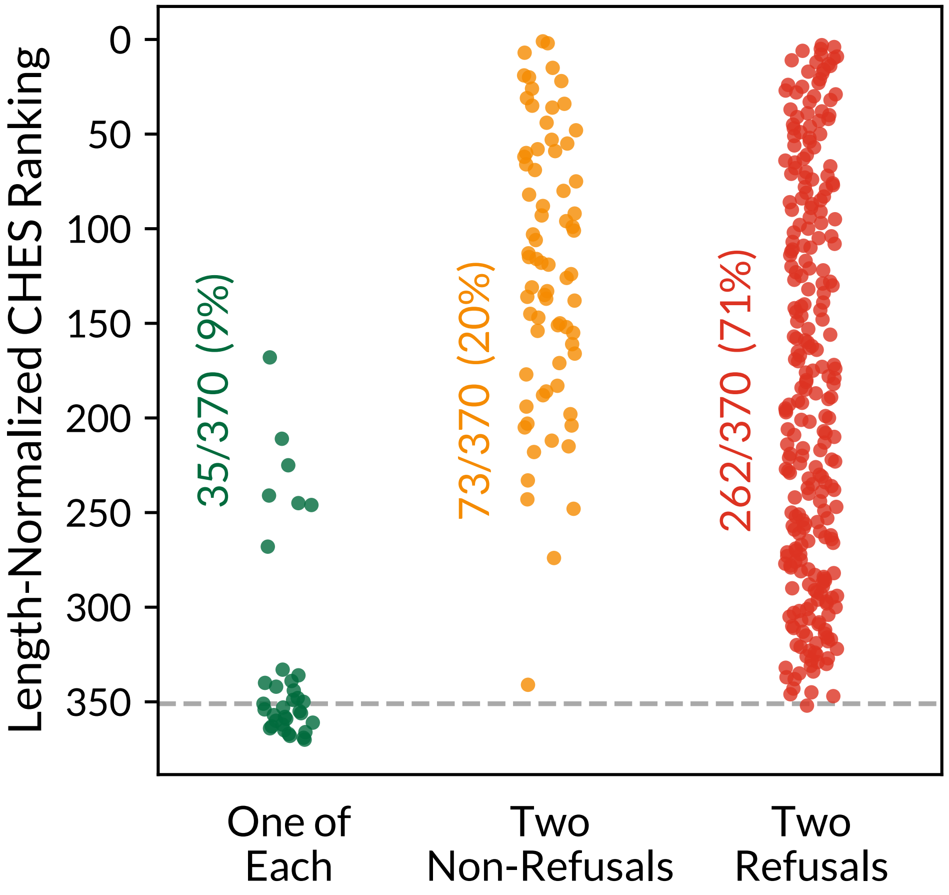
# Data Filtering via CHES Score Mitigates Unintentional Unalignment

**Recall:** Unintentional unalignment due to likelihood displacement experiments



ⓘ **Removing samples with high CHES scores mitigates unintentional unalignment, and goes beyond adding an SFT term to the loss**

# Which Samples Have a High CHES Score?



**CHES score ranking falls in line with intuition:** Samples with **two refusal** or **two non-refusal** responses tend to have a higher score than samples with **one of each**

# Conclusion: Likelihood Displacement

# Conclusion: Likelihood Displacement

⚠ Likelihood displacement can be catastrophic and cause **unintentional unlignment**

# Conclusion: Likelihood Displacement

⚠ Likelihood displacement can be catastrophic and cause **unintentional unlignment**

Theory & Experiments: Samples with **high CHES scores lead to likelihood displacement**

# Conclusion: Likelihood Displacement

Likelihood displacement can be catastrophic and cause **unintentional unlignment**

Theory & Experiments: Samples with **high CHES scores lead to likelihood displacement**

**Filtering out samples with high CHES score** can mitigate unintentional unalignment

# Conclusion: Likelihood Displacement

Likelihood displacement can be catastrophic and cause **unintentional unlignment**

Theory & Experiments: Samples with **high CHES scores lead to likelihood displacement**

**Filtering out samples with high CHES score** can mitigate unintentional unalignment

**Our work highlights the importance of curating data with sufficiently distinct preferences, for which the CHES score may prove valuable**

# Outlook

# Fundamentals of Language Model Alignment

# Fundamentals of Language Model Alignment

There are countless methods for aligning language models

RLHF
Ouyang et al. 2022

RAFT
Dong et al. 2023

IPO
Azar et al. 2023

REBEL
Gao et al. 2024

KTO
Ethayarajh et al. 2024

RLAIF
Bai et al. 2022

DPO
Rafailov et al. 2023

SLiC-HF
Zhao et al. 2023

SimPO
Meng et al. 2024

●●●

# Fundamentals of Language Model Alignment

There are countless methods for aligning language models

RLHF
Ouyang et al. 2022

RAFT
Dong et al. 2023

IPO
Azar et al. 2023

REBEL
Gao et al. 2024

KTO
Ethayarajh et al. 2024

RLAIF
Bai et al. 2022

DPO
Rafailov et al. 2023

SLiC-HF
Zhao et al. 2023

SimPO
Meng et al. 2024

● ● ●

**Limited understanding of basic questions** (e.g. loss landscape, optimization, generalization)

# Fundamentals of Language Model Alignment

There are countless methods for aligning language models

RLHF
Ouyang et al. 2022

RAFT
Dong et al. 2023

IPO
Azar et al. 2023

REBEL
Gao et al. 2024

KTO
Ethayarajh et al. 2024

• • •

RLAIF
Bai et al. 2022

DPO
Rafailov et al. 2023

SLiC-HF
Zhao et al. 2023

SimPO
Meng et al. 2024

**Limited understanding of basic questions** (e.g. loss landscape, optimization, generalization)

**Theory (mathematical or empirical) may be necessary for efficient and reliable alignment**

# Fundamentals of Language Model Alignment

There are countless methods for aligning language models

| | |
|---|---|
| RLHF | Ouyang et al. 2022 |
| RAFT | Dong et al. 2023 |
| IPO | Azar et al. 2023 |
| REBEL | Gao et al. 2024 |
| KTO | Ethayarajh et al. 2024 |
| RLAIF | Bai et al. 2022 |
| DPO | Rafailov et al. 2023 |
| SLiC-HF | Zhao et al. 2023 |
| SimPO | Meng et al. 2024 |

● ● ●

**Limited understanding of basic questions** (e.g. loss landscape, optimization, generalization)

**ⓘ Theory (mathematical or empirical) may be necessary for efficient and reliable alignment**

**Thank You!**

Work supported in part by the
Zuckerman STEM Leadership Program