Learning dynamics of LLM finetuning

报告人：刁俊程

日　　期：2025年5月7日

中国科学院计算技术研究所
Institute of Computing Technology,Chinese Academy of Sciences

提纲

- 研究背景
- 问题定义
- 研究内容
- 实验
- 总结

中科院计算所

提纲

- 研究背景
- 问题定义
- 研究内容
- 实验
- 总结

中科院计算所

研究背景

- Learning Dynamics
  - 定义
    - Learning Dynamics，就是研究每次模型经过一个batch的数据训练后，参数发生了变化，这个变化如何影响模型的表现。具体来说就是先做一个观察数据集，这个数据集是静态的，观察模型在这个数据集上的表现。
  - 研究内容
    - 这篇论文从学习Learning Dynamics的角度开展研究，首先将模型预测的变化分解为三个起不同作用的项，来形式化LLM微调的Learning Dynamics。

提纲

- 研究背景
- 问题定义
- 研究内容
- 实验
- 总结

中科院计算所

问题定义

中科院计算所

结论

Proposition 1. *Let* $\pi = \mathrm{Softmax}(\mathbf{z})$ *and* $\mathbf{z} = h_\theta(\mathbf{x})$. *The one-step learning dynamics decompose as*

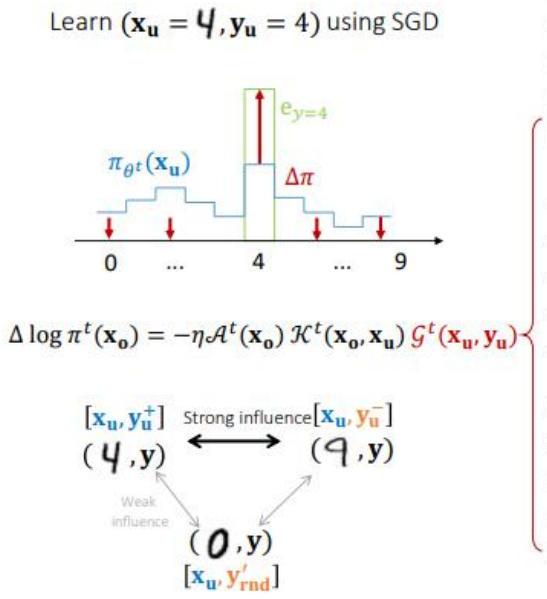
$$\underbrace{\Delta \log \pi^t(\mathbf{y} \mid \mathbf{x}_o)}_{V\times 1} = -\eta \underbrace{\mathcal{A}^t(\mathbf{x}_o)}_{V\times V} \underbrace{\mathcal{K}^t(\mathbf{x}_o, \mathbf{x}_u)}_{V\times V} \underbrace{\mathcal{G}^t(\mathbf{x}_u, \mathbf{y}_u)}_{V\times 1} + \mathcal{O}(\eta^2 \|\nabla_\theta \mathbf{z}(\mathbf{x}_u)\|_{\mathrm{op}}^2), \tag{3}$$

where $\mathcal{A}^t(\mathbf{x}_o) = \nabla_{\mathbf{z}} \log \pi_{\theta^t}(\mathbf{x}_o) = I - \mathbf{1}\pi_{\theta^t}^\top(\mathbf{x}_o)$, $\mathcal{K}^t(\mathbf{x}_o, \mathbf{x}_u) = (\nabla_\theta \mathbf{z}(\mathbf{x}_o)|_{\theta^t})(\nabla_\theta \mathbf{z}(\mathbf{x}_u)|_{\theta^t})^\top$ *is the empirical neural tangent kernel of the logit network* $\mathbf{z}$, *and* $\mathcal{G}^t(\mathbf{x}_u, \mathbf{y}_u) = \nabla_{\mathbf{z}} \mathcal{L}(\mathbf{x}_u, \mathbf{y}_u)|_{\mathbf{z}^t}$.

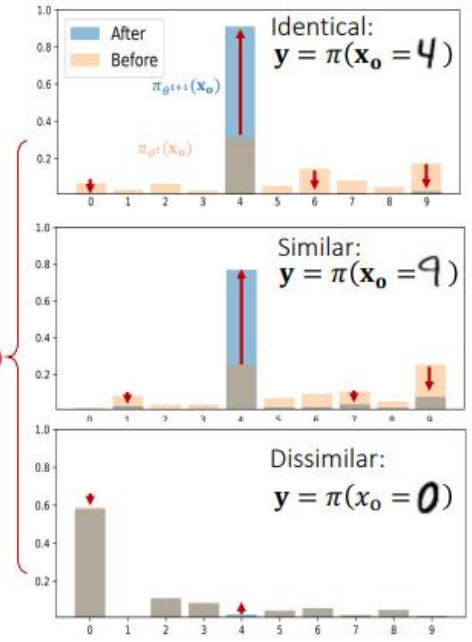
- x_u代表训练数据的样本，是用来做梯度更新的输入数据；x_o代表一个观察集合样本，是用来测试这一步训练前后模型输出差异。
- K的部分代表了在函数空间中的样本相关性（不是两个样本乍看之下像不像，而是通过梯度核（NTK）映射之后像不像）

问题定义

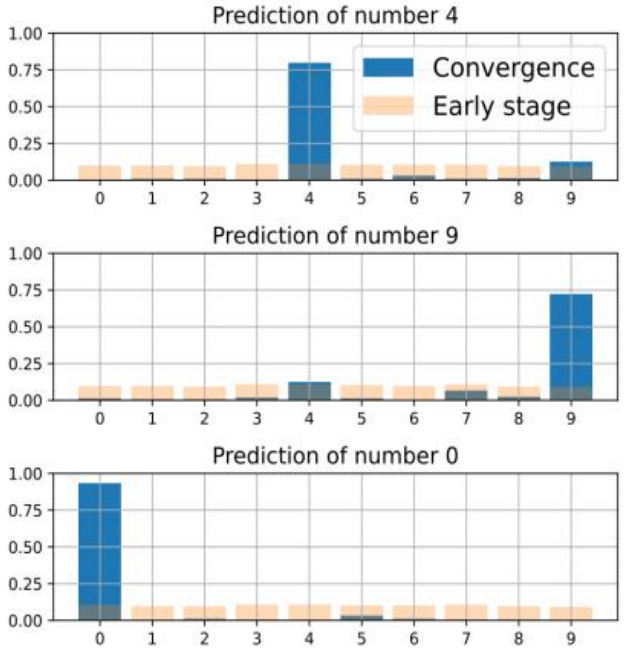
- 案例分析-MNIST案例



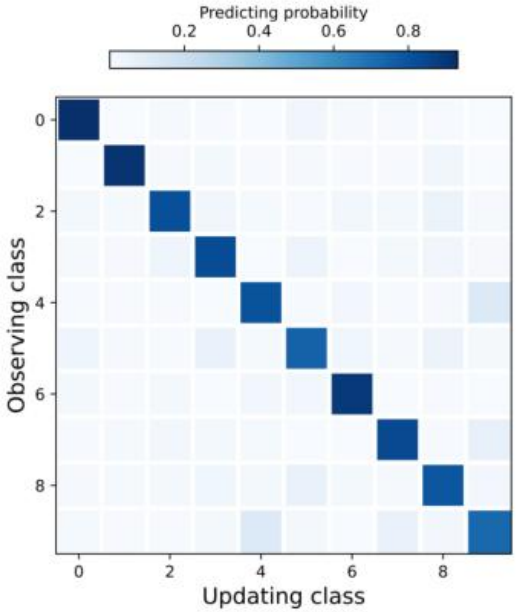
(a) Adaptation vector created by $(\mathbf{x_u}, \mathbf{y_u})$



(b) One-step change with the same $\mathcal{G}^t(\mathbf{x_u}, \mathbf{y_u})$ (large η)



(c) Accumulated change of epochs



(d) Correlation of the accumulated change

提纲

- 研究背景
- 问题定义
- 研究内容
- 实验
- 总结

研究内容

- SFT LOSS下的单步分解
  - 损失函数

$$\mathcal{L}_{\mathrm{SFT}}(\mathbf{x}_u, \mathbf{y}_u^+) \triangleq -\sum_{l=1}^{L} \log \pi(y = y_l^+ \mid \mathbf{y}_{<l}^+, \mathbf{x}_u) = -\sum_{l=1}^{L} \mathbf{e}_{y_l^+} \cdot \log \pi(\mathbf{y} \mid \mathbf{x}_u, \mathbf{y}_{<l}^+).$$

  - 分解

$$[\underbrace{\Delta \log \pi^t(\mathbf{y} \mid \boldsymbol{\chi}_o)}_{V\times M}]_m = -\sum_{l=1}^{L} \eta [\underbrace{\mathcal{A}^t(\boldsymbol{\chi}_o)}_{V\times V\times M}]_m [\underbrace{\mathcal{K}^t(\boldsymbol{\chi}_o, \boldsymbol{\chi}_u)}_{V\times V\times L}]_l [\underbrace{\mathcal{G}^t(\boldsymbol{\chi}_u)}_{V\times L}]_l + \mathcal{O}(\eta^2),$$

研究内容

- DPO LOSS下的单步分解
  - 损失函数

$$\mathcal{L}_{\mathrm{DPO}}(\theta) = -\mathbb{E}_{(\mathbf{x}_u, \mathbf{y}_u^+, \mathbf{y}_u^-)\sim\mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_{\theta^t}(\mathbf{y}_u^+ \mid \boldsymbol{\chi}_u^+)}{\pi_{\mathrm{ref}}(\mathbf{y}_u^+ \mid \boldsymbol{\chi}_u^+)} - \beta \log \frac{\pi_{\theta^t}(\mathbf{y}_u^- \mid \boldsymbol{\chi}_u^-)}{\pi_{\mathrm{ref}}(\mathbf{y}_u^- \mid \boldsymbol{\chi}_u^-)} \right) \right]$$

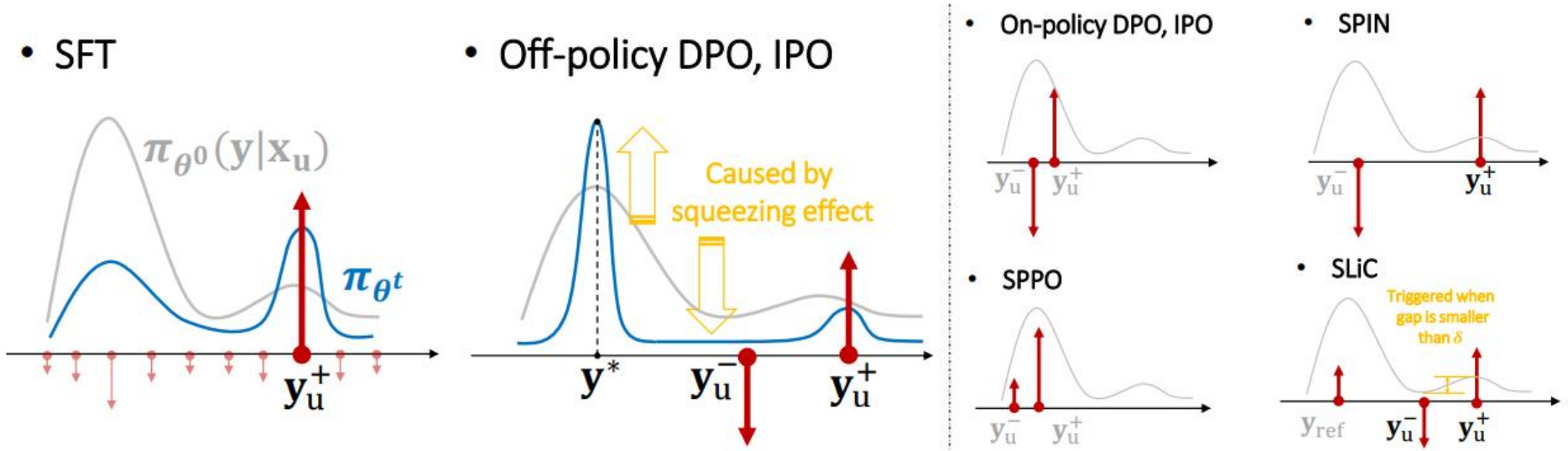
  - 分解

$$[\Delta \log \pi^t(\mathbf{y} \mid \boldsymbol{\chi}_o)]_m = -\sum_{l=1}^{L} \eta [\mathcal{A}^t(\boldsymbol{\chi}_o)]_m \left([\mathcal{K}^t(\boldsymbol{\chi}_o, \boldsymbol{\chi}_u^+)]_l [\mathcal{G}^t_{\text{DPO+}}]_l - [\mathcal{K}^t(\boldsymbol{\chi}_o, \boldsymbol{\chi}_u^-)]_l [\mathcal{G}^t_{\text{DPO-}}]_l \right) + \mathcal{O}(\eta^2)$$

$$\mathcal{G}^t_{\text{DPO+}} = \beta(1-a)\left(\pi_{\theta^t}(\mathbf{y} \mid \boldsymbol{\chi}_u^+) - \mathbf{y}_u^+\right); \qquad \mathcal{G}^t_{\text{DPO-}} = \beta(1-a)\left(\pi_{\theta^t}(\mathbf{y} \mid \boldsymbol{\chi}_u^-) - \mathbf{y}_u^-\right),$$

研究内容

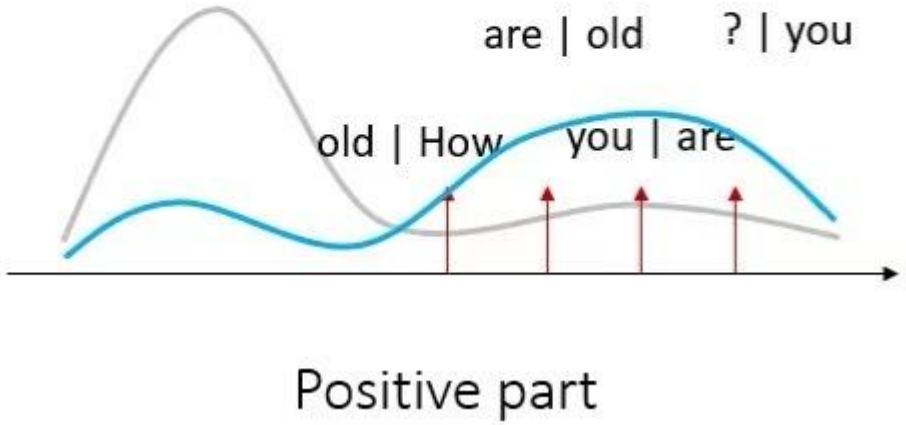
- 负梯度导致的挤压效应（off-policy）
  - **结论**：正样本和模型分布概率最高位置的值不一致时，几乎所有输出的confidence都会降低，而所有减少的probability都被挤压到模型分布概率高的地方。
  - **解释现象1**：这可能会使模型不断生成重复的短语。DPO 算法的各种变体通常会通过限制负梯度的强度或负样本的分布位置来无意中减轻这种挤压效应，这在一定程度上解释了它们的优势所在。



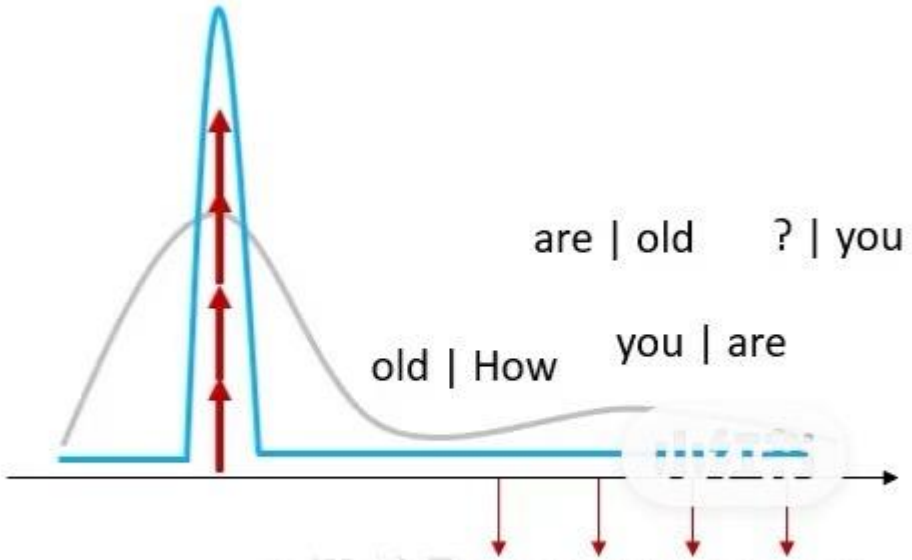
研究内容

- 负梯度导致的挤压效应（off-policy）

- Consider token-wise modeling



Positive part



Negative part

提纲

- 研究背景
- 问题定义
- 研究内容
- 实验
- 总结

中科院计算所

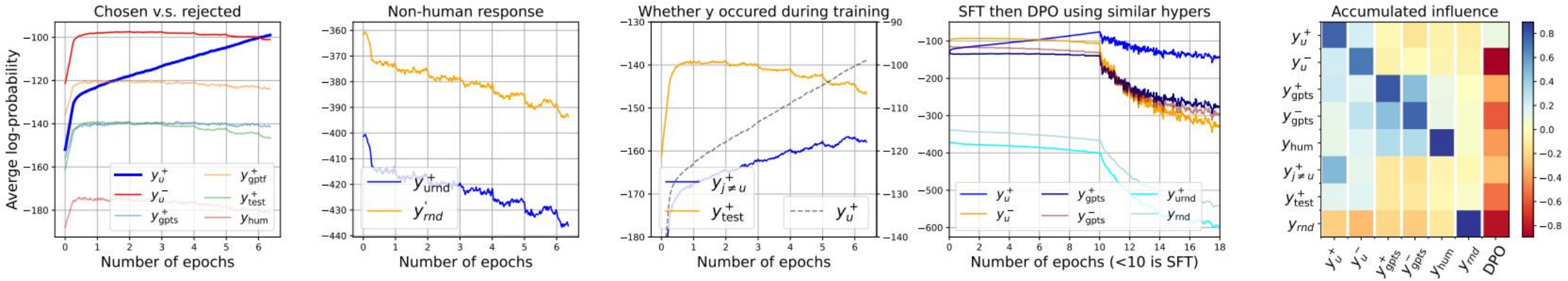
实验

- 实验设置
  - 训练数据集 D_{train}，从数据集训练集中采样5000条样本，包含输入prompt，正样本、负样本。
  - Probing数据集 D_{prob}，从训练数据集采样500而来，并且生成了一些response。
  - Probing数据集2 D_{prob2}从测试集中采样，作为消融实验。

实验

Learning Dynamics of SFT

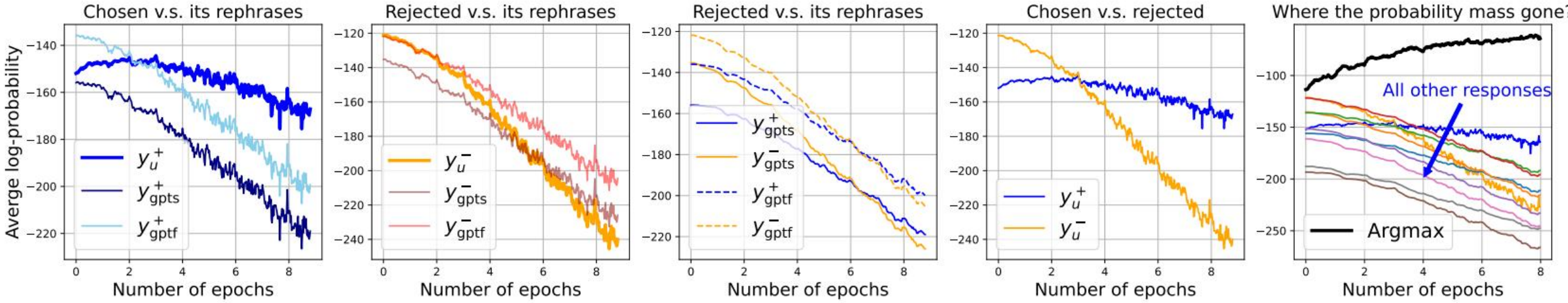
- 模型对y_u^+的信心在整个学习过程中不断增加。
- 其他response样本的model perdiction confidence在训练初期都有轻微的增加，随后随着训练的进行而逐渐减少。
- **解释现象2**：由于训练集存在u和j两组数据，都是在持续学习的，因此这种学习带来的提升超过了下降，可能导致幻觉。
- **解释现象3**：有一个有趣的发现是，所有由ChatGPT生成的response在模型看来都是非常相似的，无论它们在语义上有多么不同。可能是因为LLM有其偏好的习惯语或短语，这可以被认为是一种“指纹”。



实验

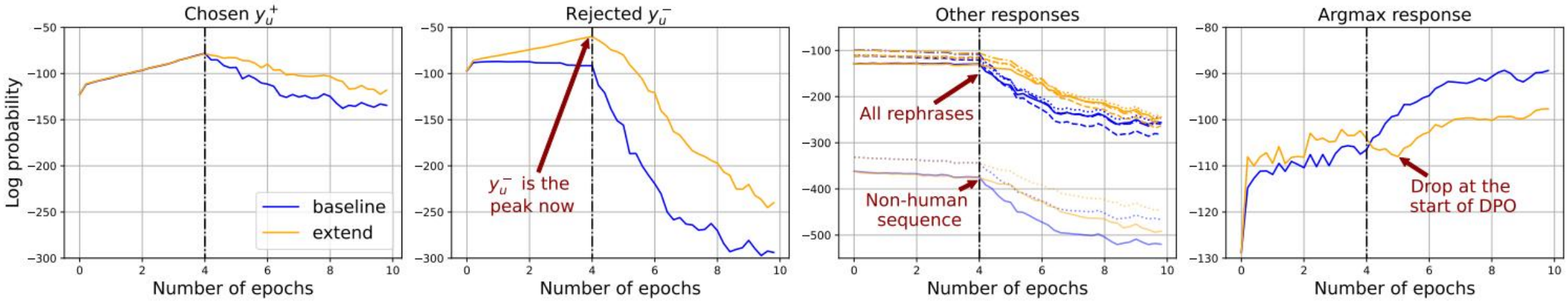
Learning Dynamics of Off-Policy DPO

- 尽管G^{t}_{dpo+}施加了对正样本的pull up，但log prob的增加并不像在SFT中那样显著。
- 这种不那么强烈的下降，是因为“挤压效应”。DPO阶段，其值迅速拉高，说明了DPO这种挤压效应的存在。然而，信心最高的token不一定是一个好的响应：它会强化原本LLM中的先验偏差（**解释现象1**）。
- 第四个图可以看出，正负样本下的差距不断增加，这就说明模型逐渐获得了区分正负样本的能力。



实验

- 通过增加训练集来缓解SFT中的挤压效应
  - 在DPO过程中，对不太可能的预测施加的负梯度导致的“挤压效应”会损害模型的表现，因此，我们可以在SFT阶段同时训练模型使用正负样本，使得负响应的概率被拉高pull up，然后再进行常规的DPO。
  - 在此之后，DPO阶段施加的“push down”压力可以有效地降低负样本及其相似response的confidence，一定程度上缓解挤压效应。



提纲

- 研究背景
- 问题定义
- 研究内容
- 实验
- 总结

总结

- 通过增加训练集来缓解SFT中的挤压效应
  - 提出AKG分解理论。
  - SFT下，learning dynamic相对直观，就是对正样本的概率增加，对特征相似的样本概率也会增加。
  - DPO下，正样本的概率会被拉升，负样本及其附近的样本概率会被挤压，会挤压到argmax。
  - 通过上述两方面的分析，作者解释了LLM微调中的三种现象，包含：
    DPO下的挤压效应导致模型“自负”，容易生成重复短语；
    LLM输出其他问题答案，因为训练数据出现的response的log prob都会有上涨现象；
    LLM似乎有起偏好的习惯语，类似指纹。
  - 作者也新增了实验验证，通过使用对比学习降低SFT下的学习效果，使得分布更加平缓，来提升性能。（先sft再dpo）

中科院计算所

谢谢大家！

Q&A