Fine-tuning Language Models with Generative Adversarial Feedback

Lau Jia Jaw, Wong Qin Jiang, Zhang Hui, Zhang Ze Yu

National University of Singapore, School of Computing

MOTIVATION & OBJECTIVE

Large Language Models (LLMs) offer immense potential for enhancing productivity across various use cases, but they can also produce undesirable outputs like inaccurate facts and harmful suggestions. OpenAI uses reinforcement learning with human feedback (RLHF) to align LLMs with human preferences, but this method can be time and resource-intensive. In this work, we explore using generative adversarial training as an alternative for scoring LLMs.

In Generative Adversarial Networks (GANs), a generator and discriminator are used to improve the generator's output quality. The discriminator learns scoring functions automatically while being trained simultaneously with the generator. This approach provides learning signals without explicit scoring for output quality, making it a potential alternative to RLHF.

OVERALL APPROACH

GAN has been used extensively in natural image generation fields but not for sequence generation since GAN is designed for generating continuous data instead of sequences of discrete tokens and discriminator can only provide reward after the whole sentence has been generated, intermediate score does not represent the performance of the whole sequence. Given these challenges, in this project, we are working on finetuning LLMs with GAN for question answering task.

We formulate the question answering task as sequence generation task. We used the Stanford Question Answering Dataset (SQuAD) which is a reading comprehension dataset comprising questions posted by crowd-workers on a set of Wikipedia articles – the piece of Wikipedia article as context, and the answer to every question is a segment of text from the context.

APPROACH I: POLICY GRADIENT

The objective of GAN training is given by:

$$\min_G \max_D V(D,G) = E_x[\log D(x)] + E_z[\log(1 - D(G(z)))]$$

where z here is the input prompt to the generator instead of noise vector.

The training of discriminator still follows:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right]$$

where $G\left(\boldsymbol{z}^{(i)}\right)$ is the i-th output token sequence from the generator.

However, since the output sequences from the generator is obtained from sampling, native backpropagation does not work. Instead, we attempted various methods to address this issue.

1. Monte Carlo policy gradient: we treat the generator as a stochastic policy and use REINFORCE algorithm to estimate the policy gradient according to:

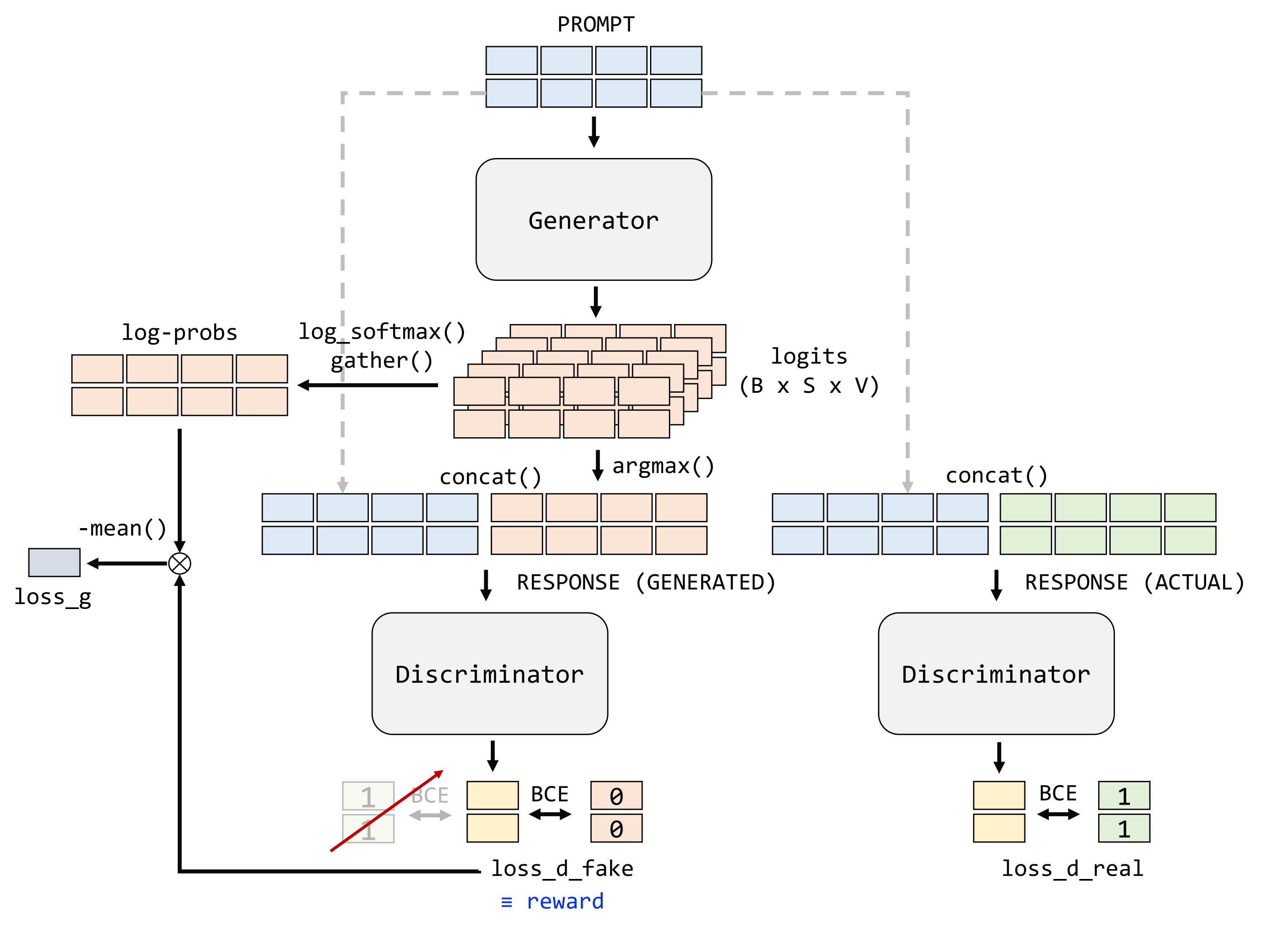
$$\nabla J = \nabla \mathbb{E}_{\pi_\theta}[r(\tau)] = \mathbb{E}_{\pi_\theta}\left[\left(\sum_{t=1}^{T}(G_t - b)\nabla \log \pi_\theta(a_t|s_t)\right)\right] \quad \theta_{t+1} = \theta_t + \alpha \nabla J(\theta_t)$$

2. Proximal policy gradient: this is the same method that OpenAI used to align InstructGPT. The objective is given by:

$$\text{objective}(\phi) = E_{(x,y)\sim D_{\pi_\phi^{\mathrm{RL}}}}\left[r_\theta(x,y) - \beta \log\left(\pi_\phi^{\mathrm{RL}}(y \mid x)/\pi^{\mathrm{SFT}}(y \mid x)\right)\right] + \gamma E_{x\sim D_{\text{pretrain}}}\left[\log(\pi_\phi^{\mathrm{RL}}(x))\right]$$

These two methods directly model the generator as a reinforcement learning (RL) agent, treat the different possible output sequences given some particular prompt as modeled by the agent's stochastic policy, and optimize the policy in accordance with the scoring given by the discriminator. The discriminator score acts as the reward signal for the RL agent.

MODEL ARCHITECTURE



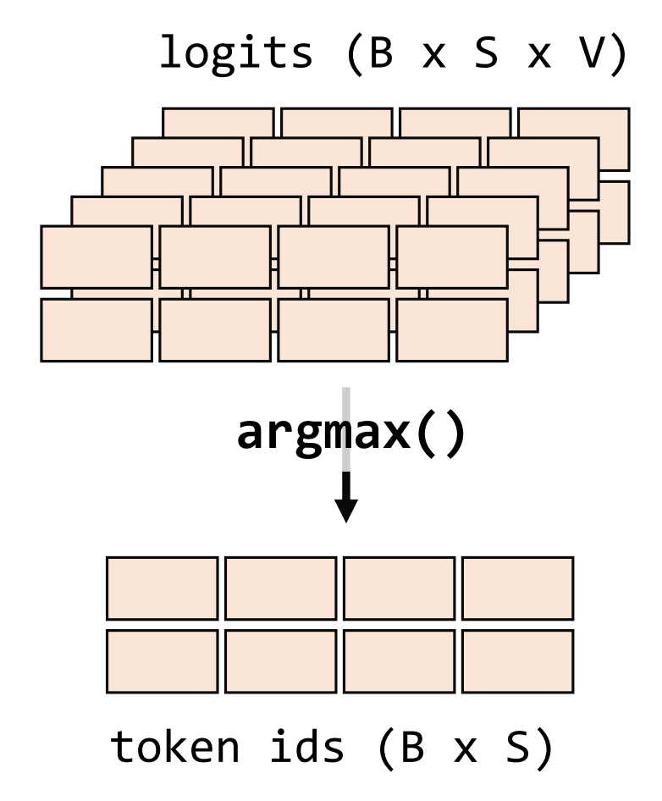
APPROACH II: GUMBEL-SOFTMAX

The GAN approach generally works for domains with continuous values such as image-related tasks. In our context, the response produced by the generator is in the form of discrete tokens (obtained using an argmax function on the model's output logits), which prevents the back-propagation of gradients from the loss to the generator via the discriminator.

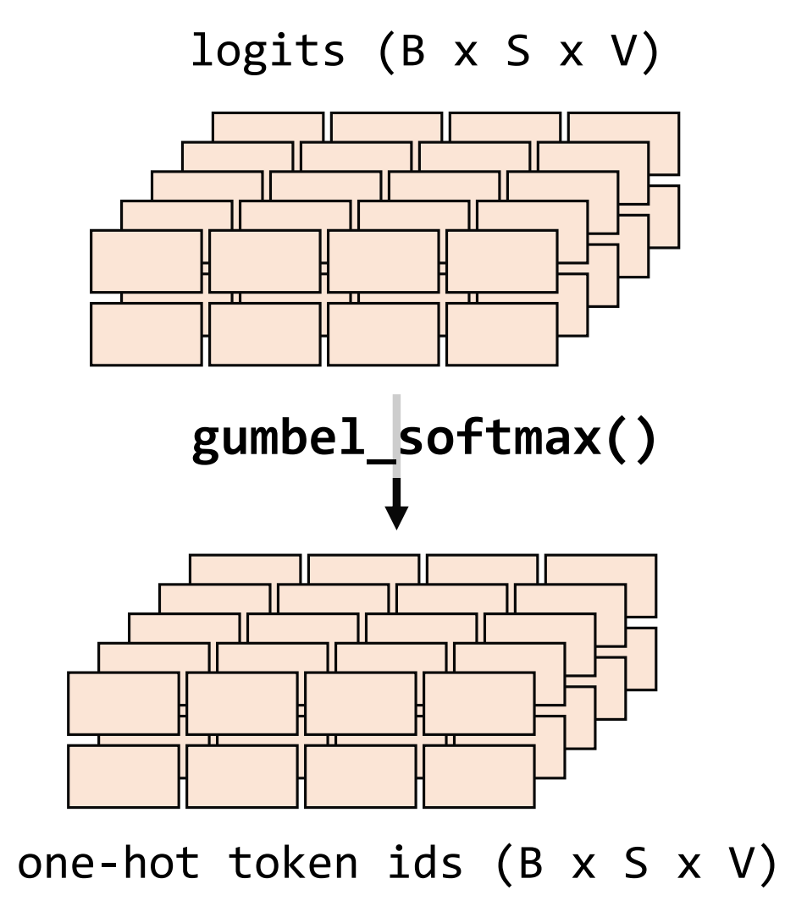
We applied the Gumbel-Softmax technique on the output logits of the generator. This technique makes use of a softmax function in place of the non-differentiable argmax function to convert the logits (continuous categorical densities) into one-hot encoded categorical distributions.

While this approach resolved the issue of back-propagation, it introduced new problems to the modelling. The discriminator's transformer embeddings layer had to be modified to handle the one-hot encoding input compared to the previous input token IDs. We implemented a custom embedding module to perform matrix multiplication, instead of the look-up operation of the default Embedding module. In addition, the prompt and ground truth response also had to be converted to one-hot encodings to compute the real data loss for the discriminator.

ARGMAX



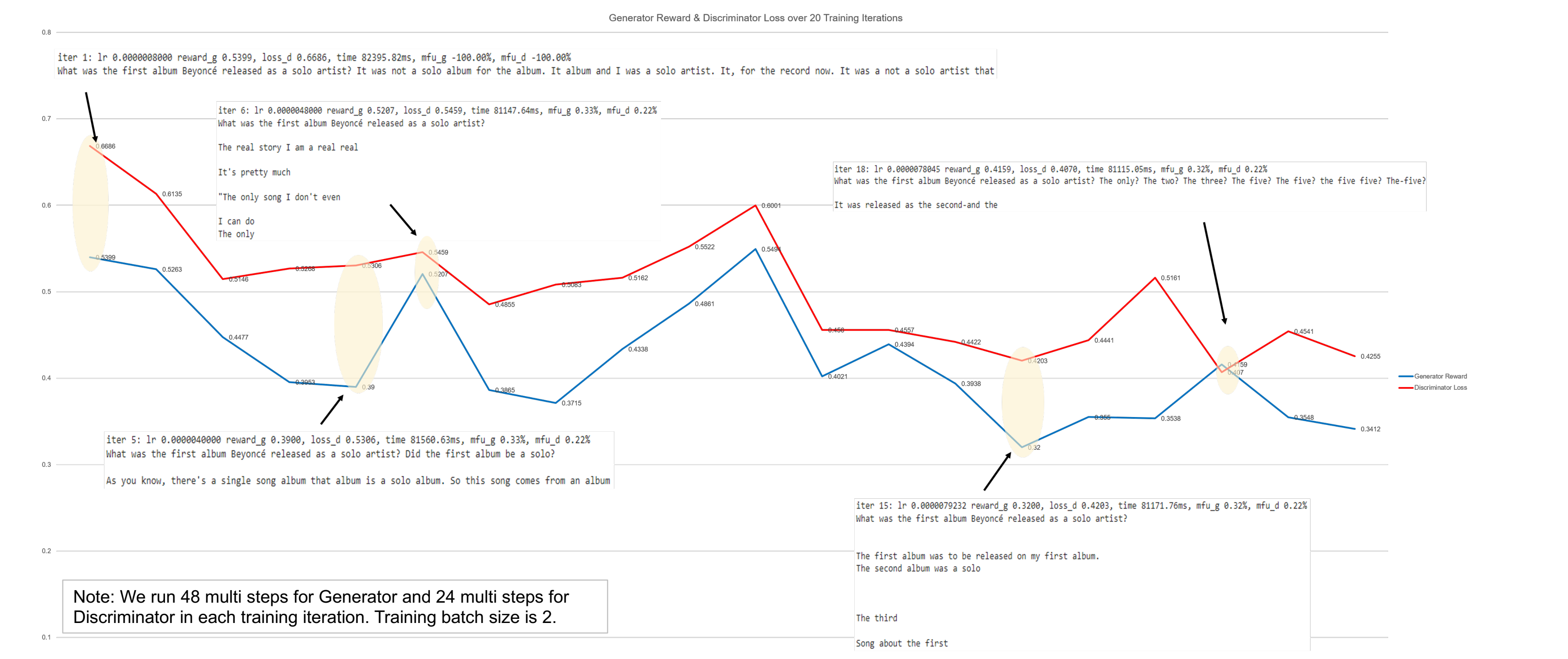
GUMBEL-SOFTMAX



RESULTS & ANALYSIS

Our hypothesis is that fine-tuning a language model will primarily align the model with the specific domain it is being trained on, rather than significantly improving its overall performance. Since we are using GPT2 as our base model, we do not anticipate any major performance improvements, but we aim to achieve our desired outcomes through GAN training. Specifically, by training our model on the SQUAD 2.0 dataset, we expect it to generate very short sentences that align with the training data.

Policy Gradient Approach



```
iter 1: lr 0.0000008000 reward_g 0.5399, loss_d 0.6686, time 82395.82ms, mfu_g -100.00%, mfu_d -100.00%
What was the first album Beyoncé released as a solo artist? It was not a solo album for the album. It album and I was a solo artist. It, for the record now. It was a not a solo artist that
```

```
iter 6: lr 0.0000048000 reward_g 0.5207, loss_d 0.5459, time 81147.64ms, mfu_g 0.33%, mfu_d 0.22%
What was the first album Beyoncé released as a solo artist?

The real story I am a real real

It's pretty much

"The only song I don't even

I can do
The only
```

```
iter 18: lr 0.0000078045 reward_g 0.4159, loss_d 0.4070, time 81115.05ms, mfu_g 0.32%, mfu_d 0.22%
What was the first album Beyoncé released as a solo artist? The only? The two? The three? The five? The five? the five five? The-five?

It was released as the second-and the
```

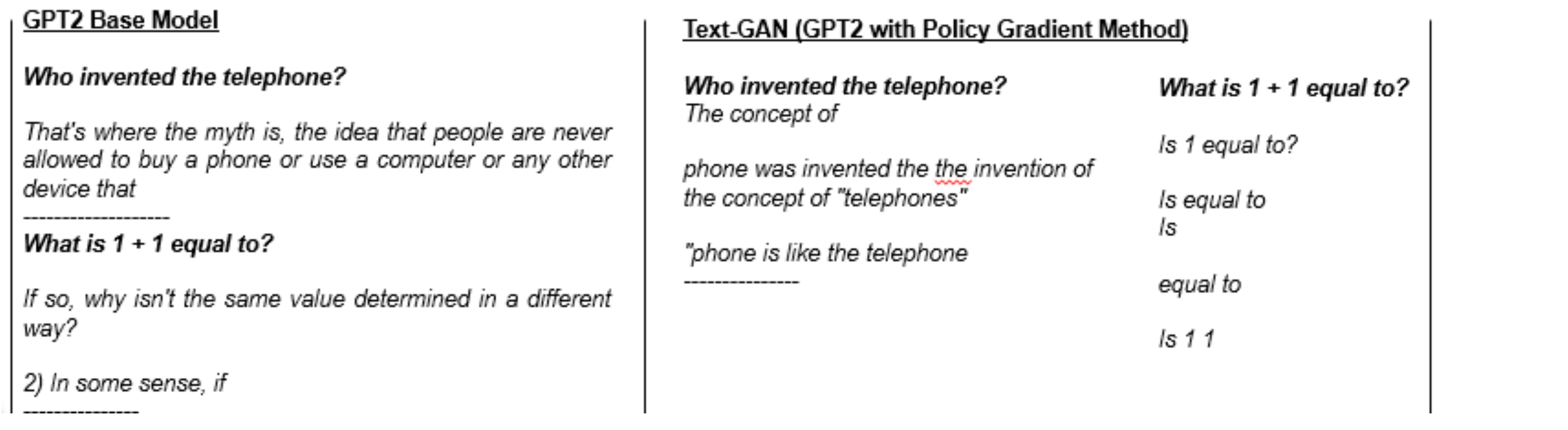
```
iter 5: lr 0.0000040000 reward_g 0.3900, loss_d 0.5306, time 81560.63ms, mfu_g 0.33%, mfu_d 0.22%
What was the first album Beyoncé released as a solo artist? Did the first album be a solo?

As you know, there's a single song album that album is a solo album. So this song comes from an album
```

```
iter 15: lr 0.0000079232 reward_g 0.3200, loss_d 0.4203, time 81171.76ms, mfu_g 0.32%, mfu_d 0.22%
What was the first album Beyoncé released as a solo artist?


The first album was to be released on my first album.
The second album was a solo


The third

Song about the first
```

Note: We run 48 multi steps for Generator and 24 multi steps for Discriminator in each training iteration. Training batch size is 2.

Training observation: We observed a significant drop in the Generator reward when the Discriminator started learning and its loss decreased. By iteration 6, the Generator output a sentence that was structurally very similar to the training data, resulting in a spike in reward. However, as the Discriminator continued to improve (as evidenced by the downward trend in its loss), the Generator also began generating sentences with different structures in order to get higher reward.

We also compared the outputs from both our Policy Gradient Method model and GPT2 Base model using the same hyperparameters, including Temperature = 0.8, Top-k = 100. While the base model generated longer and more complete sentences, as expected from a sentence completion model, our model generated shorter sentences in multiple lines. This behavior is consistent with our hypothesis.



GPT2 Base Model

Who invented the telephone?

That's where the myth is, the idea that people are never allowed to buy a phone or use a computer or any other device that

----

What is 1 + 1 equal to?

If so, why isn't the same value determined in a different way?

2) In some sense, if

Text-GAN (GPT2 with Policy Gradient Method)

Who invented the telephone?
The concept of

phone was invented the the invention of the concept of "telephones"

"phone is like the telephone

What is 1 + 1 equal to?

Is 1 equal to?

Is equal to
Is

equal to

Is 1 1

Degeneration and Collapse of Model Output (Gumbel-Softmax Approach)

```
[QUESTION] Can an AI model have consciousness?

[ANSWER] What are you trying to determine should could could should could could could might could could
could were could could could could could could could could could could might could could may could could
should could could should might could could could could could could could might could could could could
```

LIMITATIONS

1. High Variance of Policy Gradient Method
Monte Carlo estimates rely on random samples of the environment and are therefore inherently noisy, especially for highly non-linear objective functions or highly stochastic policies. Gradient estimates are also sensitive to hyperparameter tuning and can have high variance, leading to slow convergence or even divergence.

2. Instability of GAN Training
GAN training is notoriously difficult and unstable, as it involves two competing neural networks. Common failure modes include vanishing gradients (the discriminator is too good) and mode collapse (the generator always produces the same plausible output). Hyperparameter tuning and regularization are needed in order to successfully train GAN, which requires significant time and iterative effort.

3. Limited Computing Resources
Training GAN-based language models requires significant computing resources. Given the project limitations, we used smaller models (e.g. GPT-2 small and DistilGPT) in our experiments, along with reduced sequence length and batch size to minimize CUDA out-of-memory issues. These limit the learning capacity and expressiveness of our GAN models to produce high quality text.