Deep Reinforcement Learning from Human Preferences

Paul F Christiano
OpenAI
paul@openai.com
Alignment research center

Jan Leike
DeepMind
leike@google.com
Openai

Tom B Brown
nottombrown@gmail.com
Anthropic

Miljan Martic
DeepMind
miljanm@google.com
Kosen

Shane Legg
DeepMind
legg@google.com
DeepMind

Dario Amodei
OpenAI
damodei@openai.com
Anthropic

2017년도 논문

Member of Technical Staff
OpenAI
Dec 2018 - Dec 2020 · 2 yrs 1 mo

I led the engineering of GPT-3, and was responsible for the model-parallel distributed training infrastructure that scaled us from 1.5B parameters to 170B parameters.... ...see more

Language Models are Few-Shot Learners
Recent work has demonstrated substantial gains on many NLP tasks and benchmarks by pre-training on a large corpus of text followed by fine-tuning on a specific task. While...

Member Of Technical Staff
Google Brain
Aug 2017 - Nov 2018 · 1 yr 4 mos
San Francisco Bay Area

I worked with Ian Goodfellow's adversarial machine learning group to design attacks and defenses for ML systems.

nt Learning
eferences

Tom B Brown
nottombrown@gmail.com

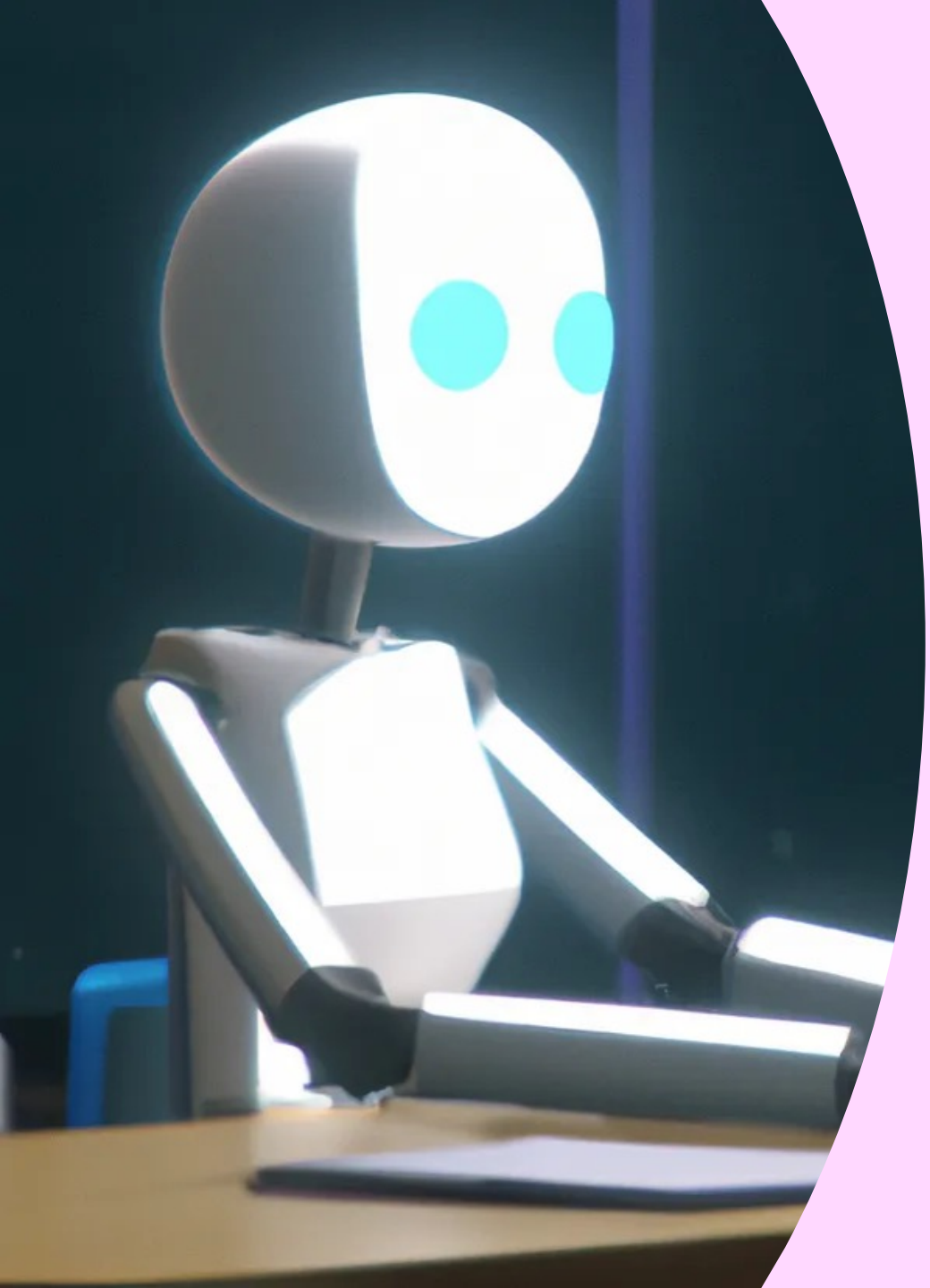
Language Models are Few-Shot Learners

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, Dario Amodei

Recent work has demonstrated substantial gains on many NLP tasks and benchmarks by pre-training on a large corpus of text followed by fine-tuning on a specific task. While typically task-agnostic in architecture, this method still requires task-specific fine-tuning datasets of thousands or tens of thousands of examples. By contrast, humans can generally perform a new language task from only a few examples or from simple instructions - something which current NLP systems still largely struggle to do. Here we show that scaling up language models greatly improves task-agnostic, few-shot performance, sometimes even reaching competitiveness with prior state-of-

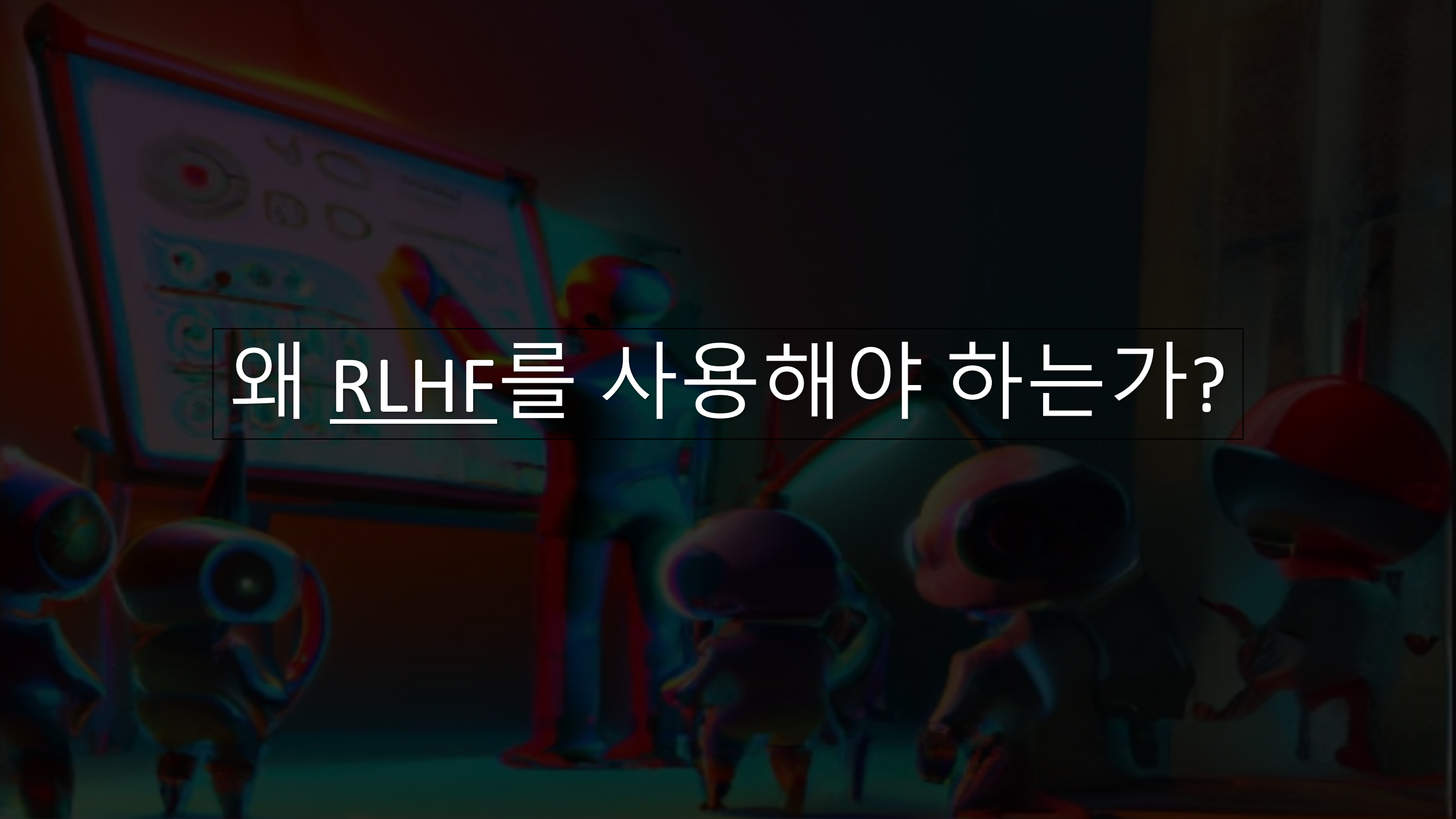
~

8656회 인용

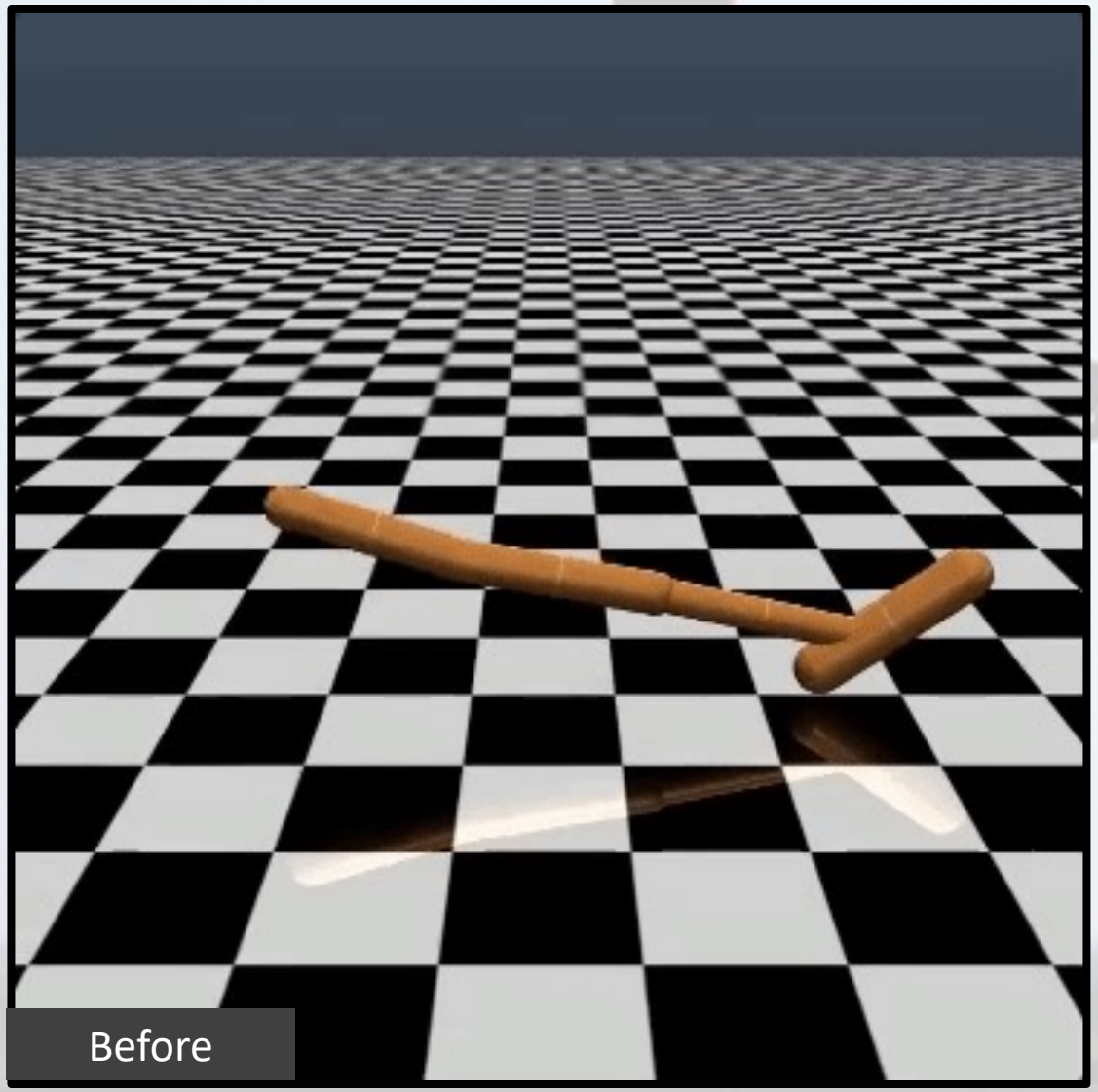


OPENAI's Planning for AGI

- RLHF는 Human feedback을 통해 더 정교한 학습을 다중 TASK에 적용할 수 있었다.
- RLHF가 GPT에서 Chatgpt로의 변화가 매우 주요했고, 인공지능은 점점 더 AGI에 가까워지고 있다.
- AGI를 위해선 RLHF와 같은 Aligning 기술이 더욱더 중요해질 것이다.


왜 RLHF를 사용해야 하는가?

복잡한 Task 학습



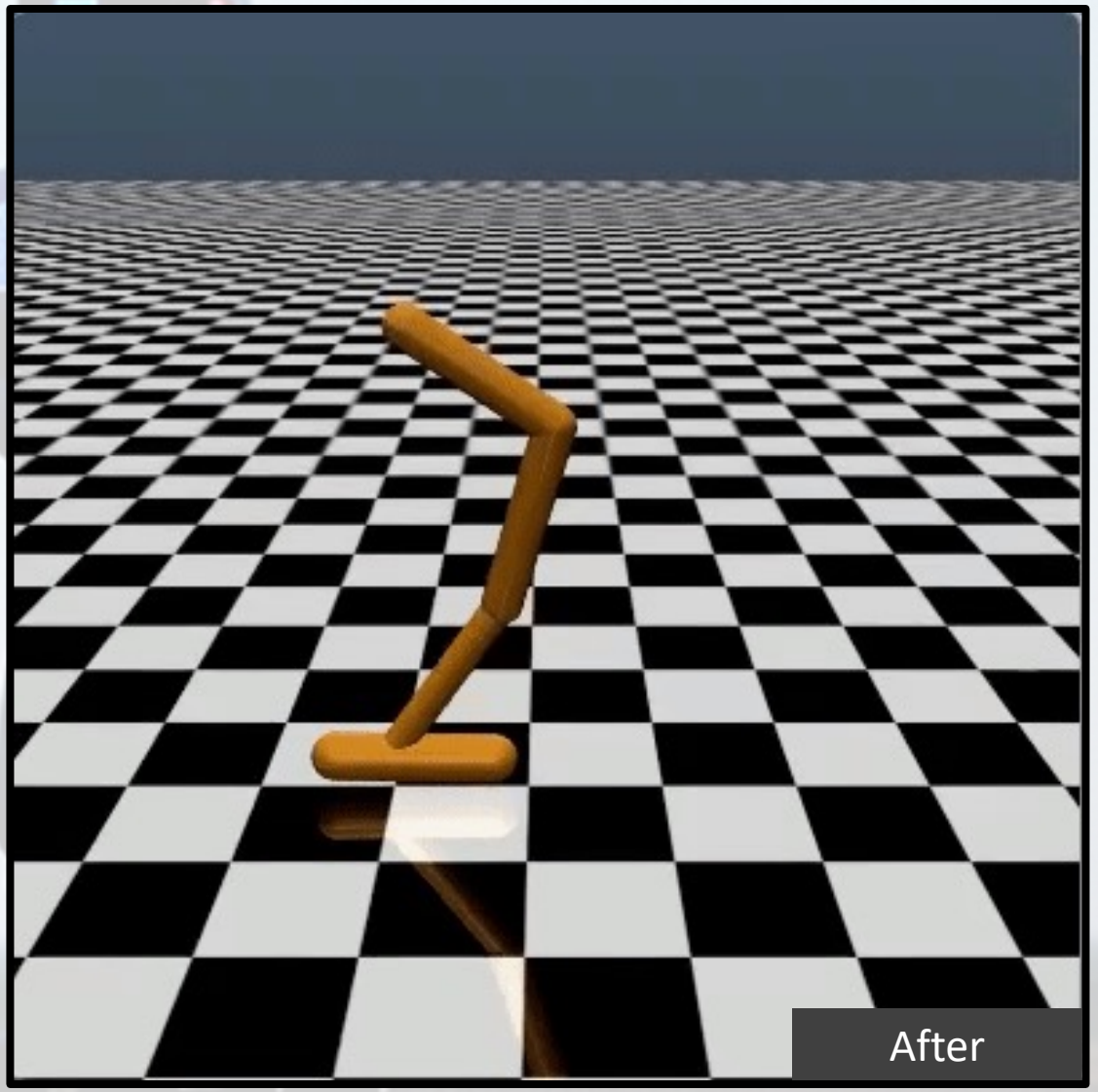


Figure. Hopper backflip (before, after)

w/o RLHF

Write an email to my mother wishing her Merry Christmas, and ask how dad is doing.

Read the book I got for Christmas: The Screwtape Letters by C. S. Lewis

Finish my Christmas shopping

Watch the Grinch Who Stole Christmas (good one)

Finish Christmas cards.

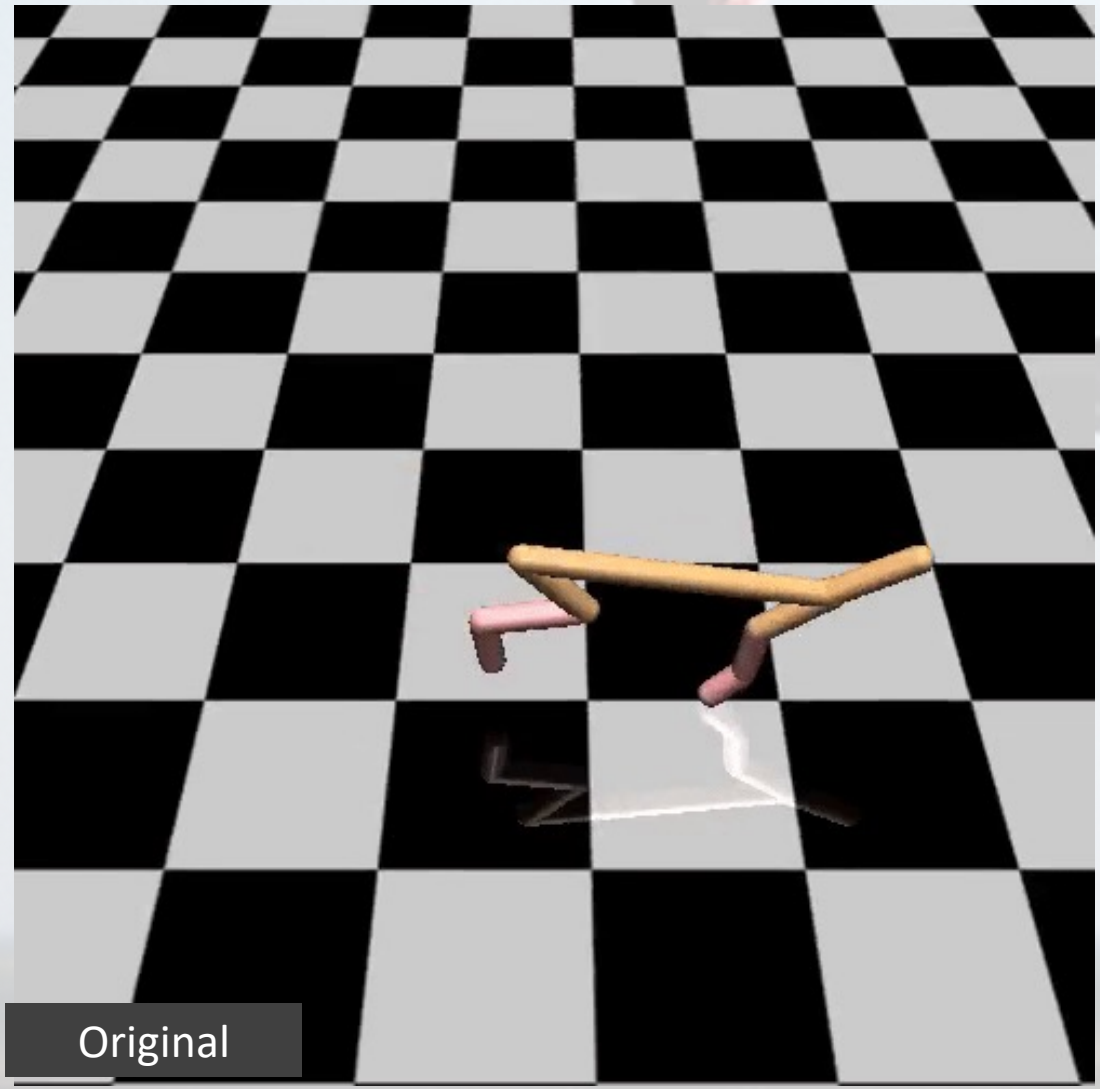
Go to church

w/ RLHF

Write an email to my mother wishing her Merry Christmas, and ask how dad is doing.

Hi Mom,
Merry Christmas! I hope you're doing well. I'm wondering how Dad is doing. I hope he's doing well. I hope to talk to you soon.
Take care!

다양한 Task 학습



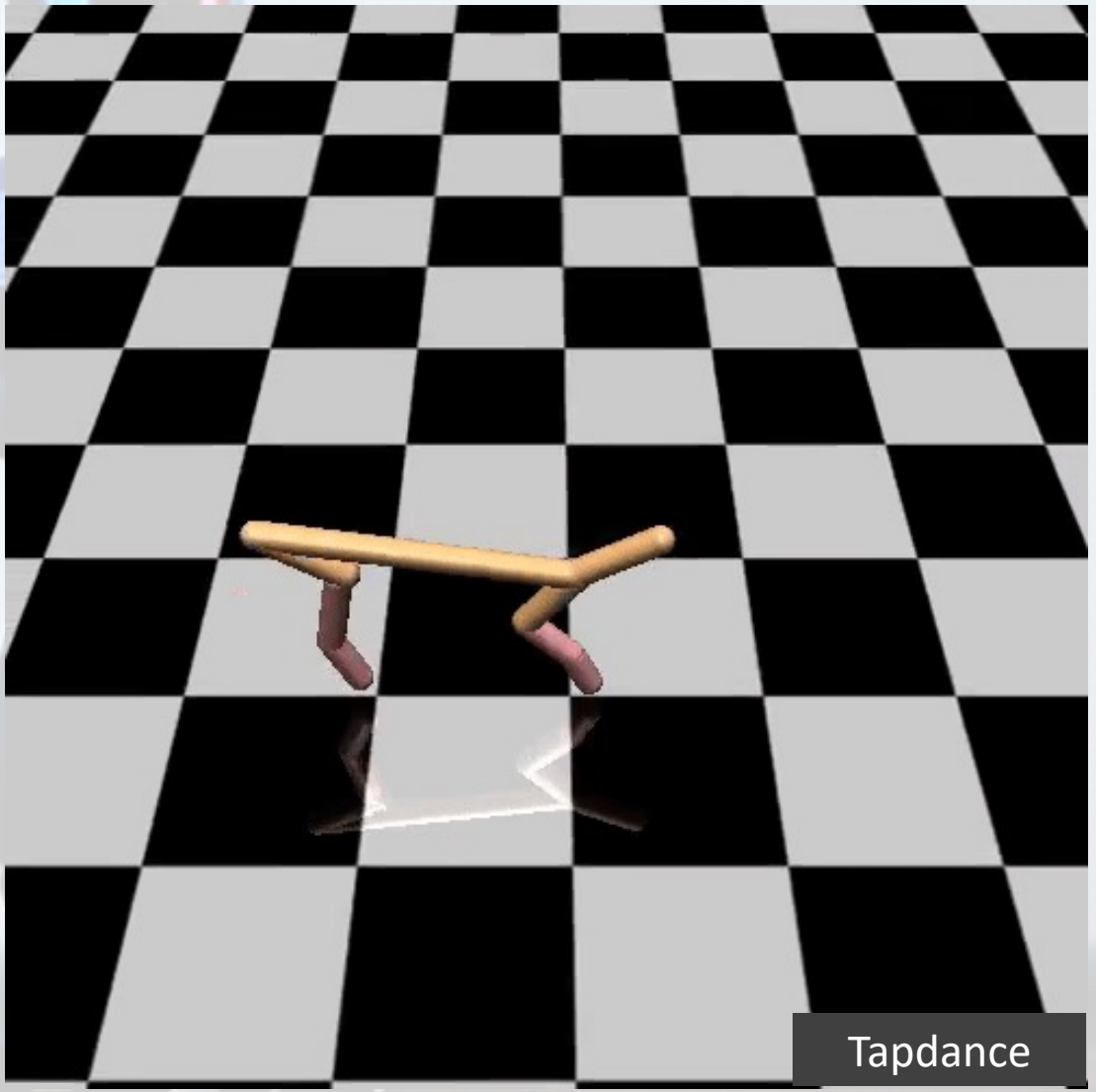
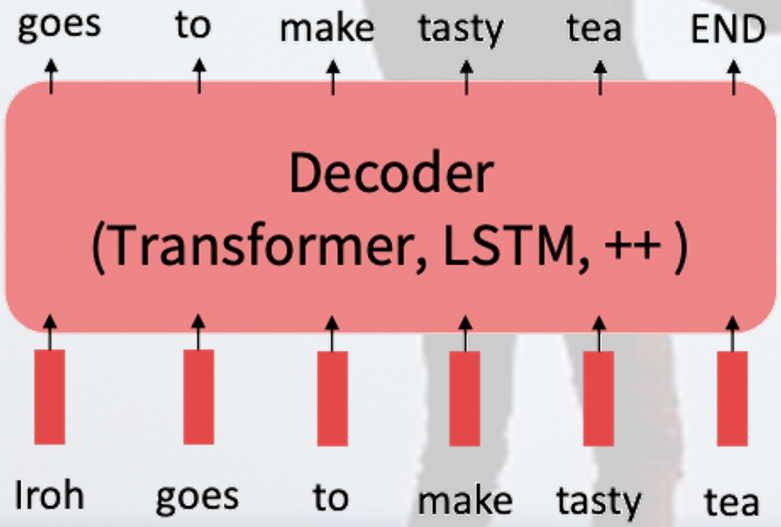


Figure. Cheetah (Original)

Aligning!

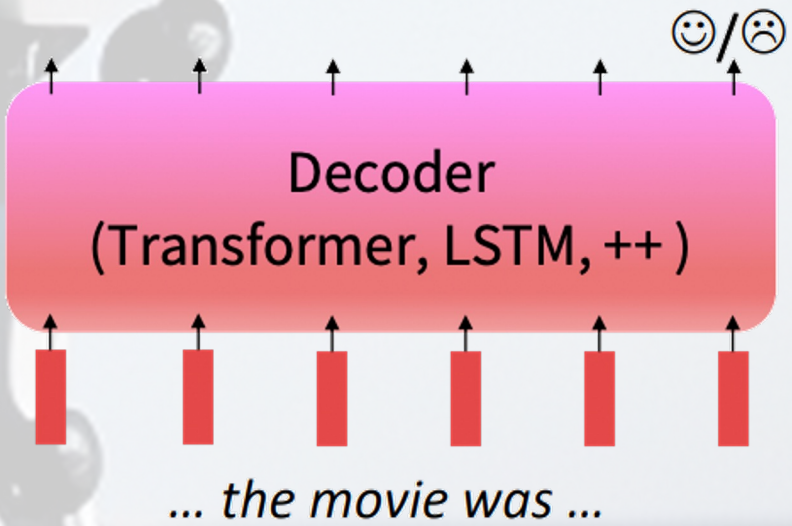
Step 1: Pretrain (on language modeling)

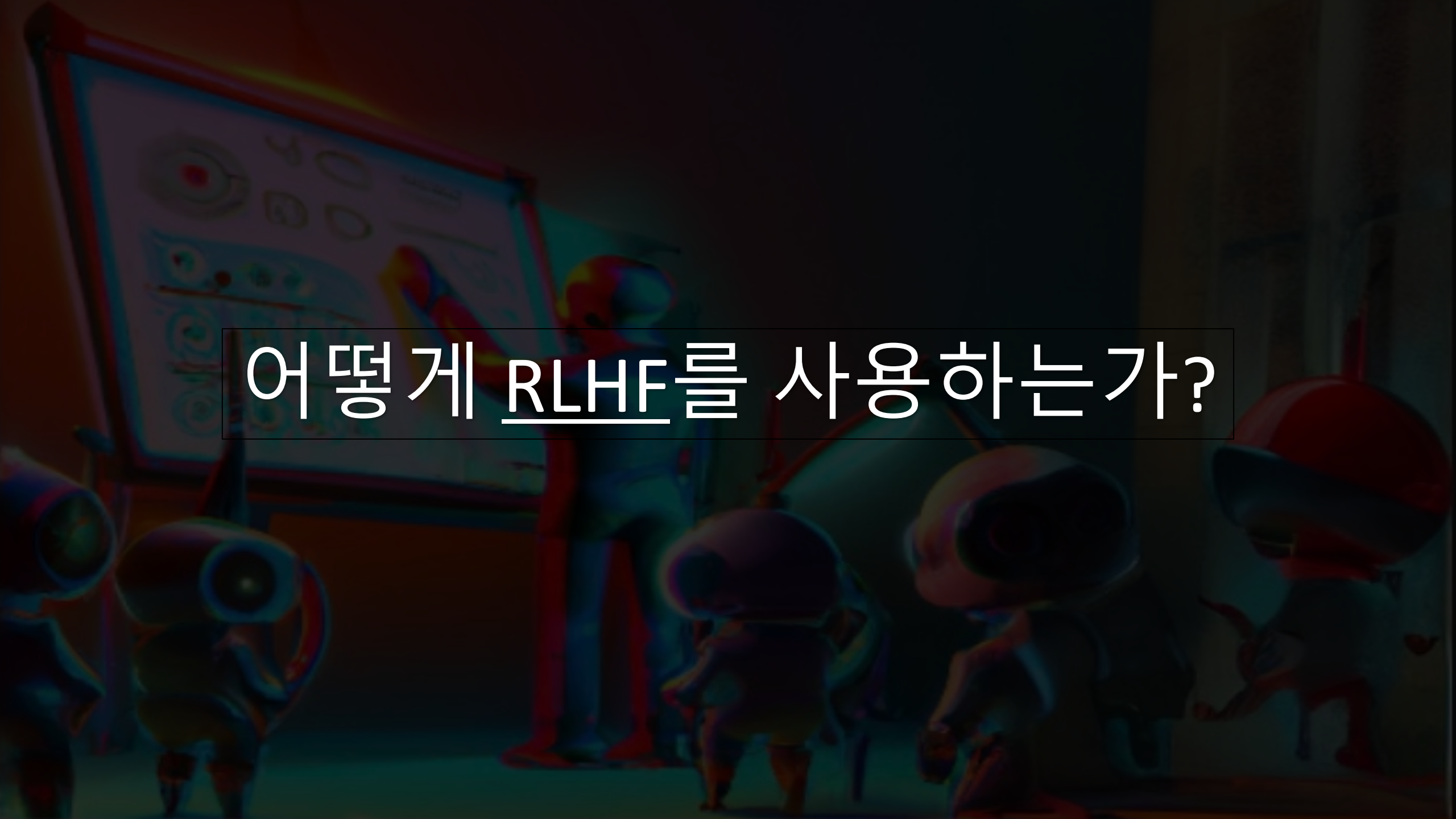
Lots of text; learn general things!



Step 2: Finetune (on many tasks)

~~Not~~ many labels; adapt to the tasks!




어떻게 RLHF를 사용하는가?

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-3.5 with supervised learning.

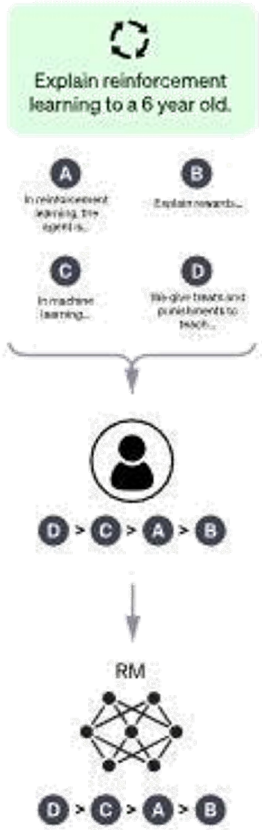
Step 2

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 the PPO reinforcement learning algorithm.

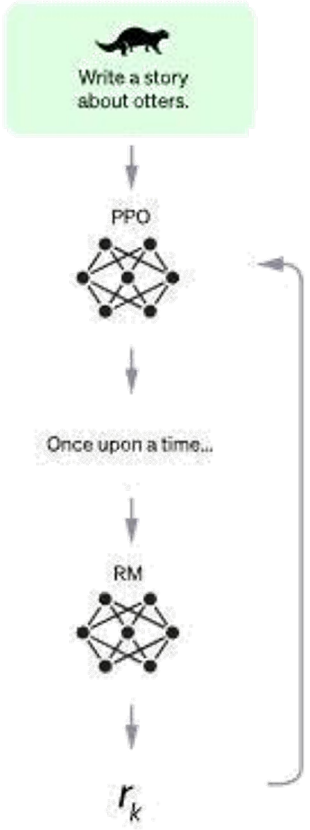
A new prompt is sampled from the dataset.

The PPO model is initialized from the supervised policy.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.



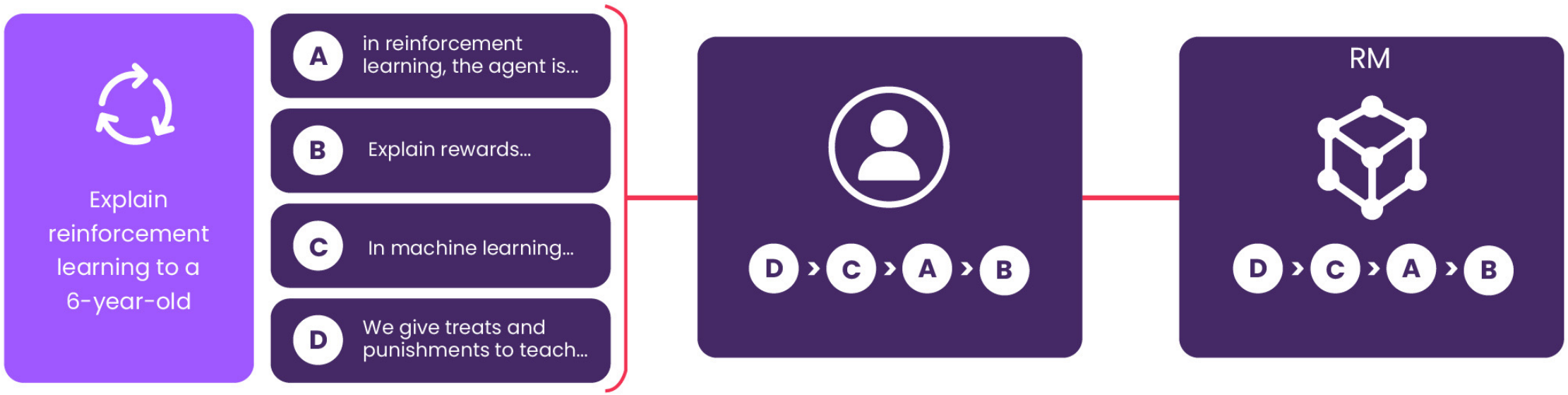
Step 2.

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 trained our reward model



Why Reward model?

- 저희는 모델을 사람이 선호하는 인공지능으로 학습하고 싶습니다. 예를들면, 착하게 말하는 그런 인공지능을 원하는거죠. 그래서, 사람의 선호(Human Preference)를 Reward로 구성하여 maximize하기 위해 강화학습을 사용합니다.
- **문제점** : 강화학습은 시뮬레이션을 이용해 매우 많은 Trial&Error로 학습하는 방법입니다. 그러나 **사람이 직접 시뮬레이션에 들어간다면 비용이 엄청날겁니다.**
- **해결책** : **사람의 선호를 인공지능 모델한테 학습**시켜 강화학습 시뮬레이션엔 사람대신 이 **모델을 사용**하는 겁니다. 이를 Reward Model이라 하겠습니다.

S1: 강화학습은 다양한 목적식(Objective)를 학습시키는데 유용하다.

S2: 강화학습은 완벽하다.

$R(s_1) = 8.0$

$R(s_2) = 1.6$

Why Reward model?

- 이번엔 데이터 구축과 관련된 얘기를 해보려 합니다. 방금 전에 아래와 같은 데이터를 보여드렸었죠.

S1: 강화학습은 컴퓨터 에이전트가 역동적인 환경에서 반복적인 시행착오 상호작용을 통해 작업 수행 방법을 학습하는 머신러닝 기법의 한 유형입니다. $R(s_1) = 8.0$

S2: 강화학습은 머신러닝 기법이다. $R(s_2) = 1.0$

- 그러면, 아래와 같은 문장을 한번 평가해보시겠습니까?

S3: 강화학습은 시행착오 상호작용을 통해 배우는 머신러닝 기법이다. $R(s_3) = ?$

- 이번엔 둘을 비교하는 방법으로 평가를 해보죠.

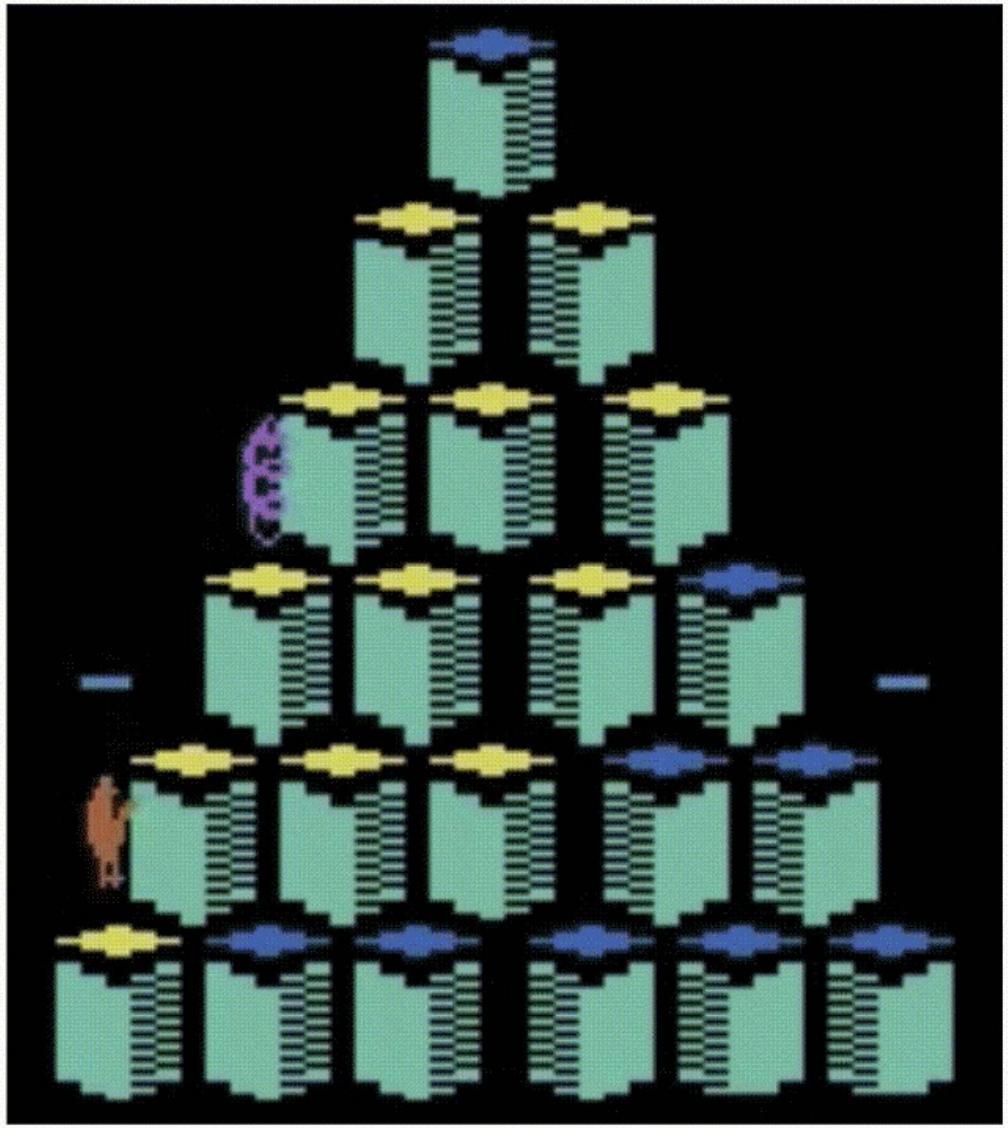
S2: 강화학습은 머신러닝 기법이다. vs S3: 강화학습은 시행착오 상호작용을 통해 배우는 머신러닝 기법이다. R(S1) < R(S2)

사실 이것은 사람의 평가 방식이 **절대적 평가**보다 **상대적 평가**에 특화되어 있다는 점을 이용한 겁니다.

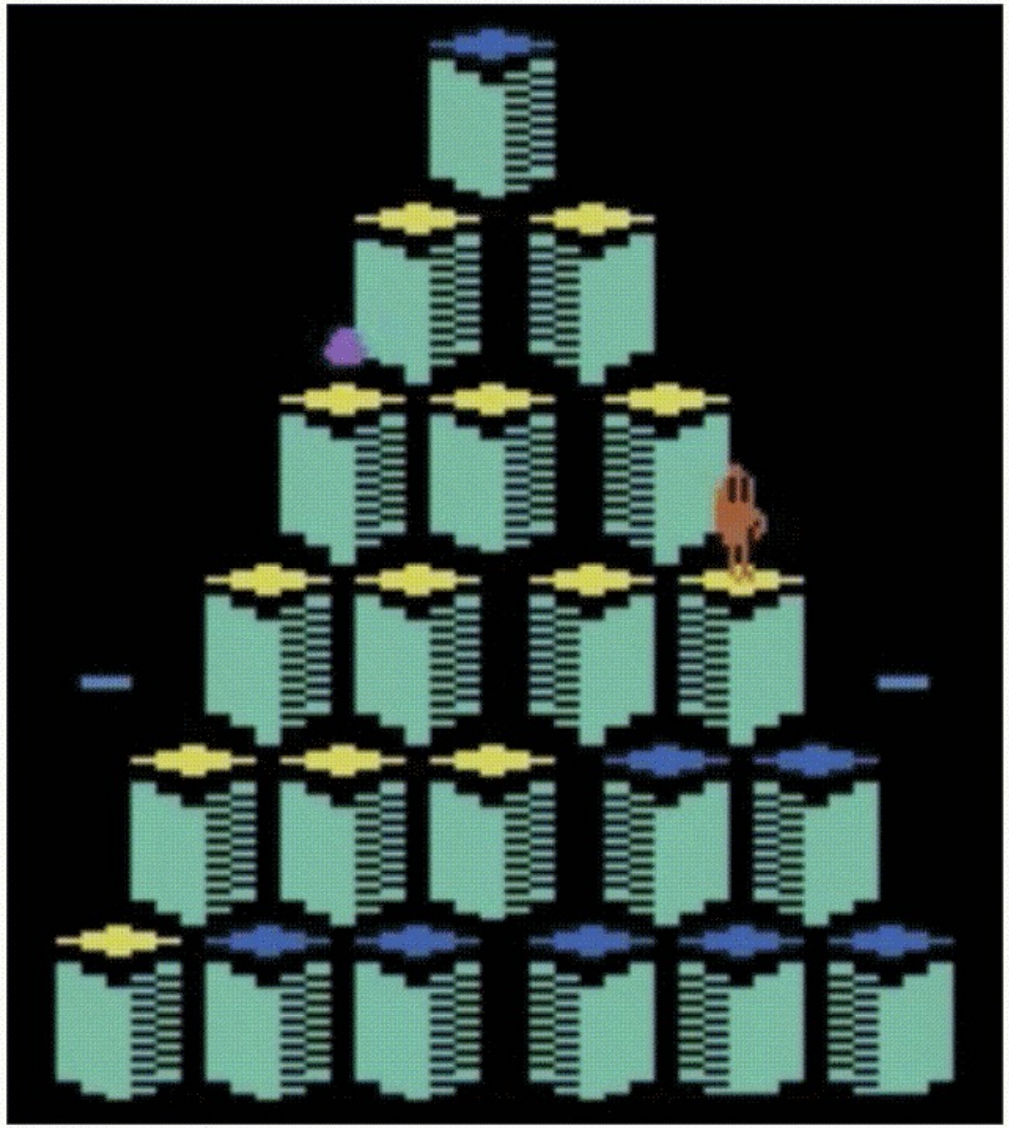
상대적 평가 방식을 이용했을 때 절대적 평가 방식을 이용한 Reward model보다 정교하게 구축할 수 있죠.

물론 절대적 평가를 이용하는게 잘못된 건 아닙니다, 꼭 두 문장을 비교해야 하는 것도 아닙니다.

Left



Right



It's a tie

Left is better

Can't tell

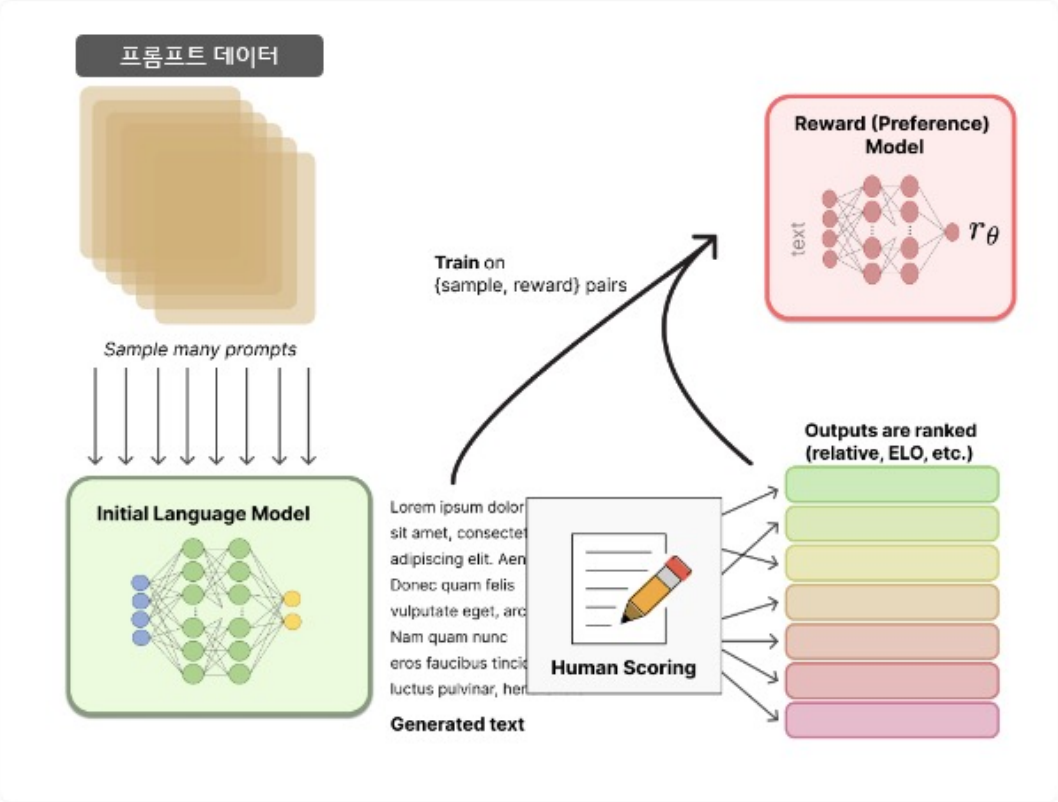
Right is better

데이터 구축


질문 (Prompt).
답변 1.
1: 백빵 한국
2. 반경 20km이내 초토화
3. 거의 없음.
...
165 ···
답변 2.
내가 진짜 간단하게 알려줄게
1. 우리가 이기는게 당연한거야
2. 북한 핵 쓰레기라서 그렇게 많이 못가
...
395 ···
답변 3.
너는집에서 메이플이나해라
형들이 총들고나간다
9,999+ ···
답변 3 > 답변 2 > 답변 1
총 3개의 comparison 데이터

Reward Model in Environment

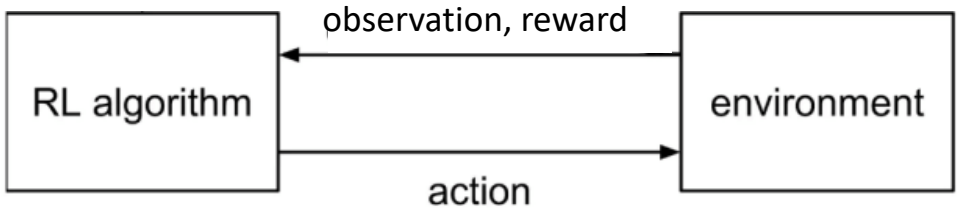
Reward model



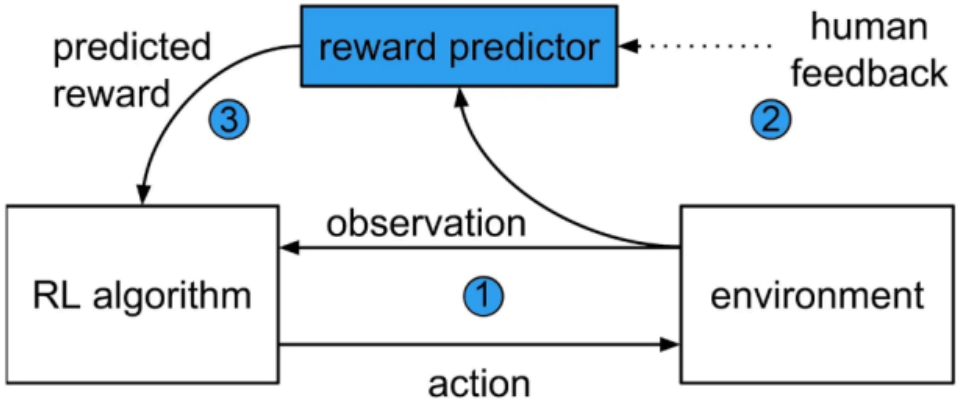
평가 결과 -> Supervised Learning

Environement

Reward model 적용 전

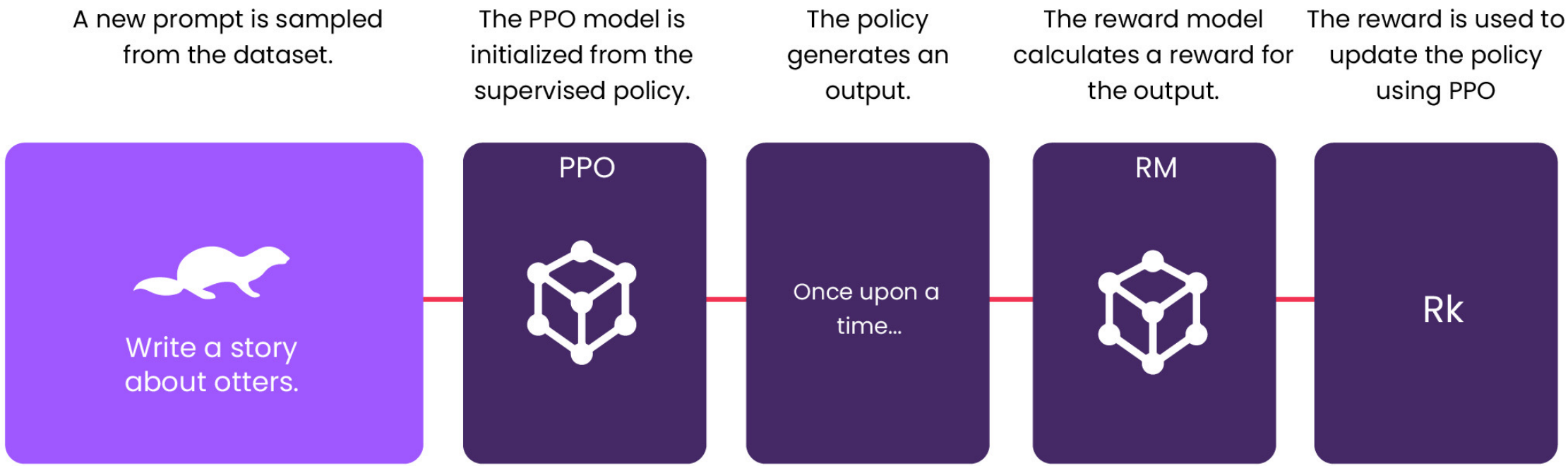


Reward model 적용 후



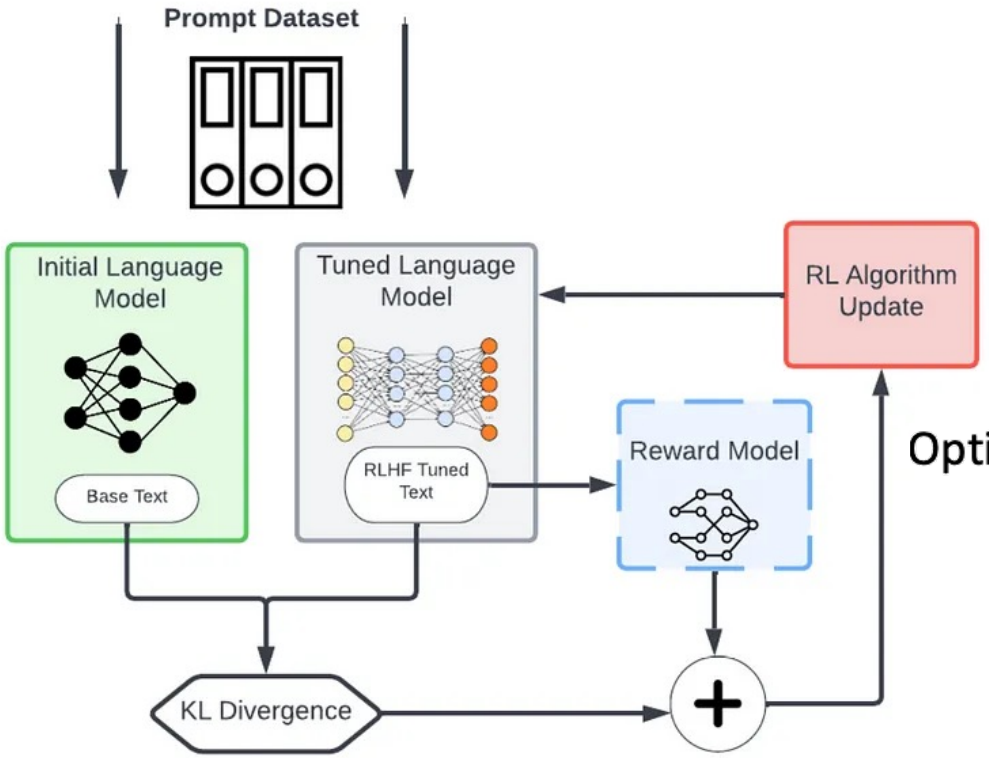
Step 3.

Optimize a policy against the reward model using the PPO reinforcement learning algorithm



Reward Model to RLHF Environment

Reward Model $RM_\phi(s)$:



$$\mathrm{R}(\mathrm{x},\mathrm{y}) = \mathrm{r}(\mathrm{x},\mathrm{y}) - \beta \log \left[\frac{\pi^{\mathrm{RL}}(\mathrm{y} \mid \mathrm{x})}{\pi^{\mathrm{SFT}}(\mathrm{y} \mid \mathrm{x})} \right]$$

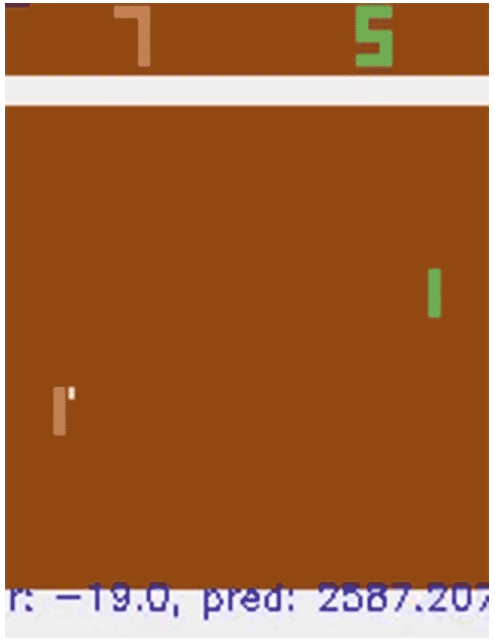
Optimize the following reward with RL: ppo-ptx

$$R(s) = RM_\phi(s) - \underbrace{\beta \log \left(\frac{p_\theta^{RL}(s)}{p^{PT}(s)} \right)}$$

Pay a price when $p_\theta^{RL}(s) > p^{PT}(s)$

This is a penalty which prevents us from diverging too far from the pretrained model. In expectation, it is known as the **Kullback-Leibler (KL)** divergence between $p_\theta^{RL}(s)$ and $p^{PT}(s)$.

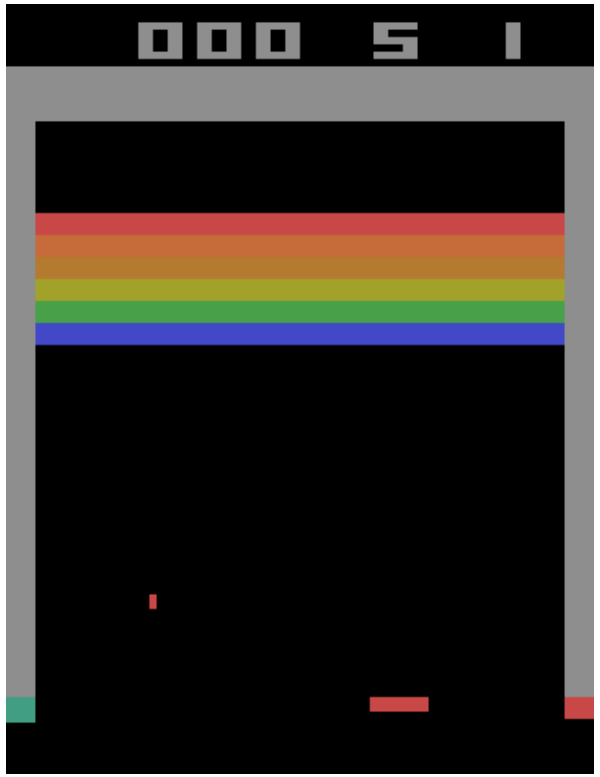
RLHF의 문제점



- 상대평가도 완벽하지 않습니다.
  - Ex) 가위 바위 보
- Reward model을 이용한 환경도 ppo-ptx도 비효율적이긴 하죠
  - Pretrain, Reward model 학습, RLHF 학습 (3중의 학습 구조)
  - KL divergence, Reward를 구하기 위해 Inference도 3중으로 이뤄짐.
- 학습된 Reward 모델 사용하는 것도 문제가 있습니다.

RLHF 성능

복잡한 게임



단순한 게임

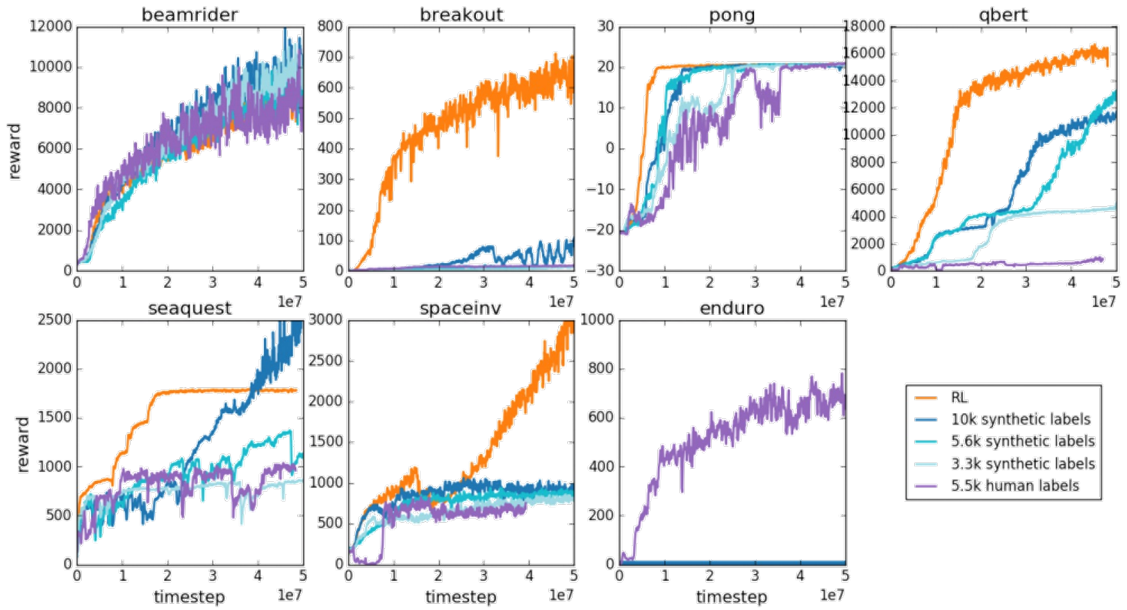


Figure 3: Results on Atari games as measured on the tasks' true reward. We compare our method using real human feedback (purple), our method using synthetic feedback provided by an oracle (shades of blue), and reinforcement learning using the true reward function (orange). All curves are the average of 3 runs, except for the real human feedback which is a single run, and each point is the average reward over about 150,000 consecutive frames.

결론

- RLHF는 매우 유용한 기술입니다.
  - 특히 복잡하거나 어려운 학습을 한다거나
  - Multi-task (Alignment) 등을 위한 Fine tuning에 유용하죠.
- 물론 한계도 있겠지만 이미 chat-gpt를 통해 유용함은 증명이 된 것 같습니다.
  - 특히 이번을 계기로 reward를 이용하면 많은 문제의 설계를 할 수 있다는 장점이 증명된 것 같습니다.

REWARD IS ENOUGH