Directly Fine-Tuning Diffusion Models on Differentiable Rewards

Kevin Clark* Paul Vicol* Kevin Swersky David Fleet

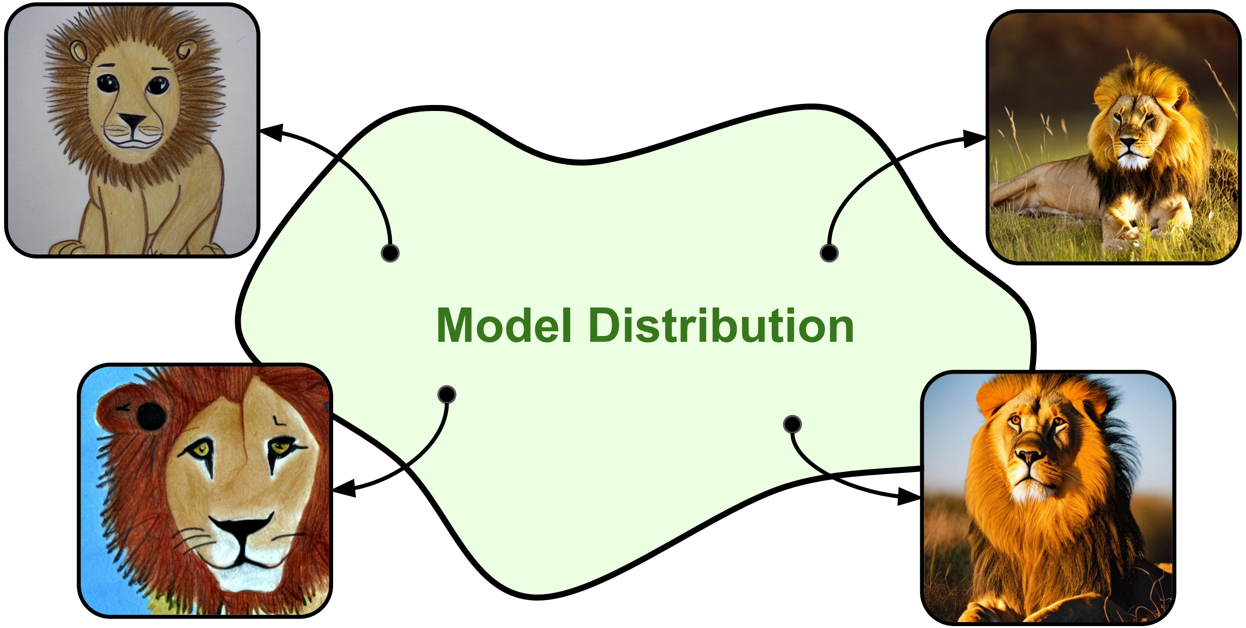
ICLR 2024

Google DeepMind

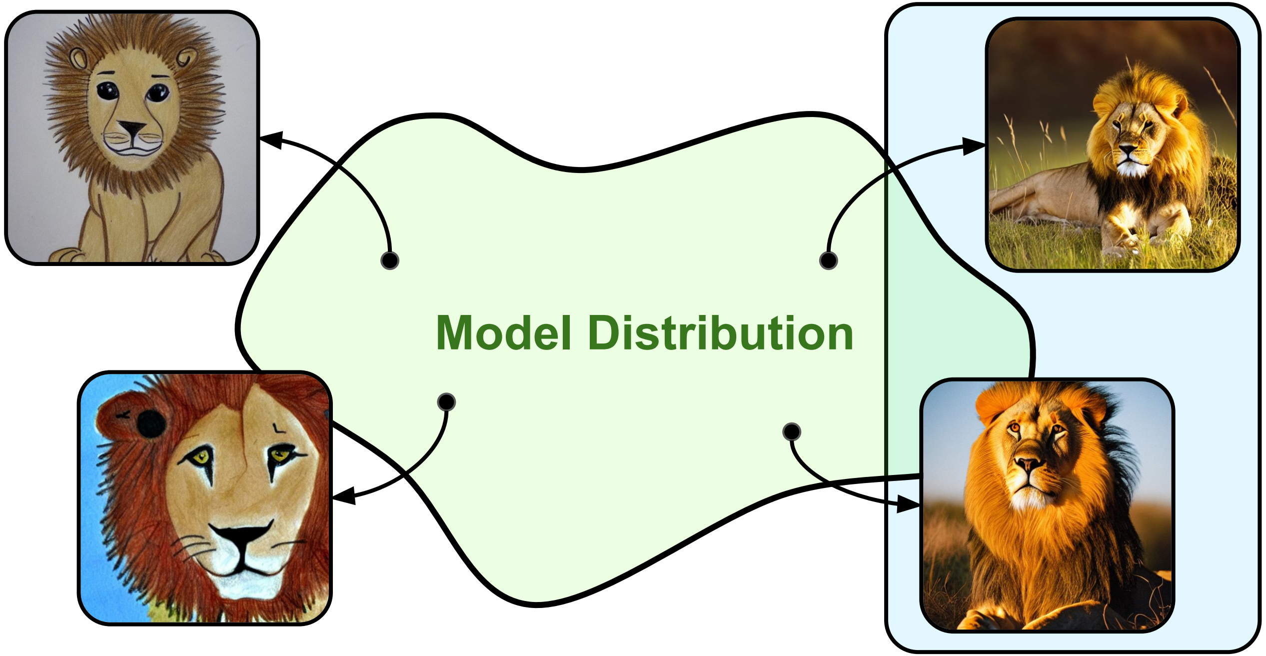
Motivation: Fine-Tuning Diffusion Models

- *Diffusion models are pre-trained to model the data distribution (e.g., diverse web images)*



Motivation: Fine-Tuning Diffusion Models

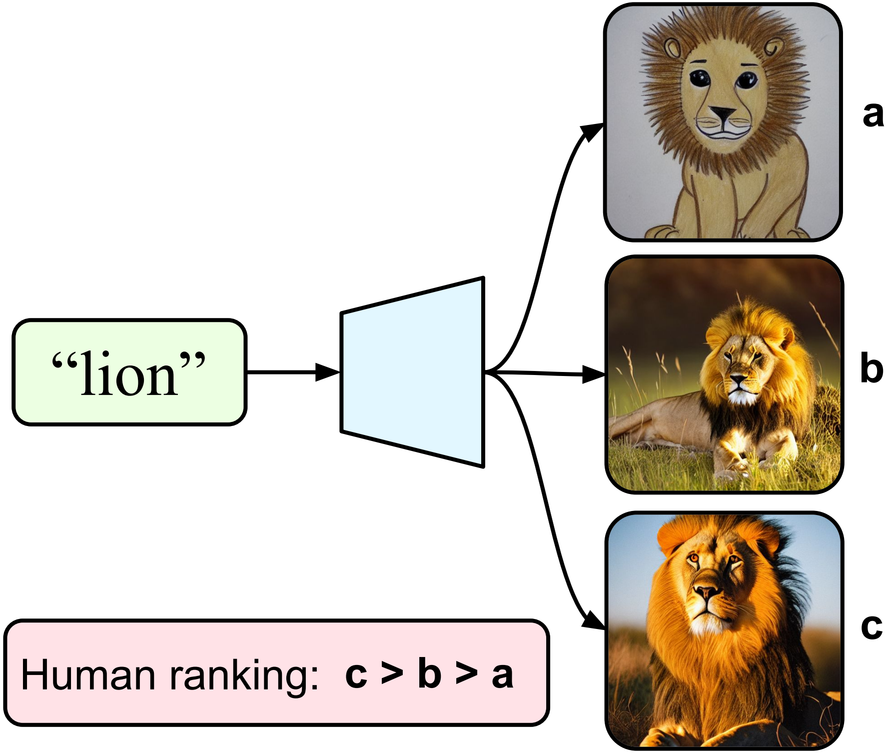
- *Diffusion models are pre-trained to model the data distribution (e.g., diverse web images)*



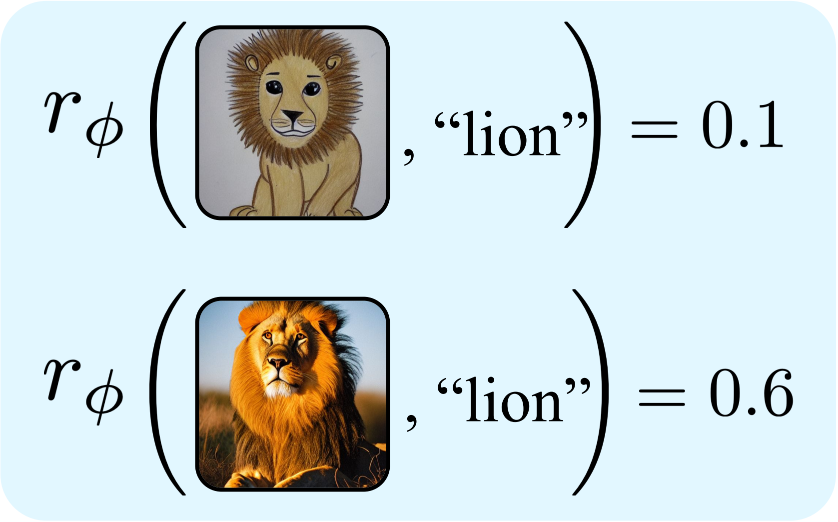
- But modeling the data distribution exactly often does not align with desired behavior
- E.g., *generating images with certain aesthetic qualitites*

Human Preference Datasets

Human Preference Data



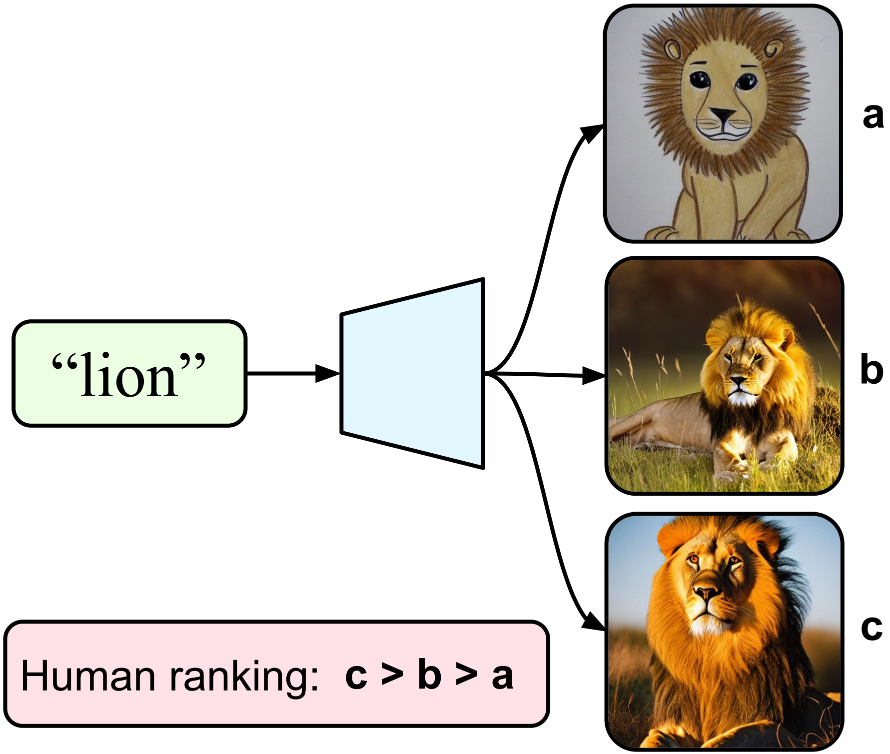
Reward Models



Human Preference Dataset (HPDv2) (Wu et al., 2023)
Pick-a-Pic Dataset (Kirstain et al., 2023)

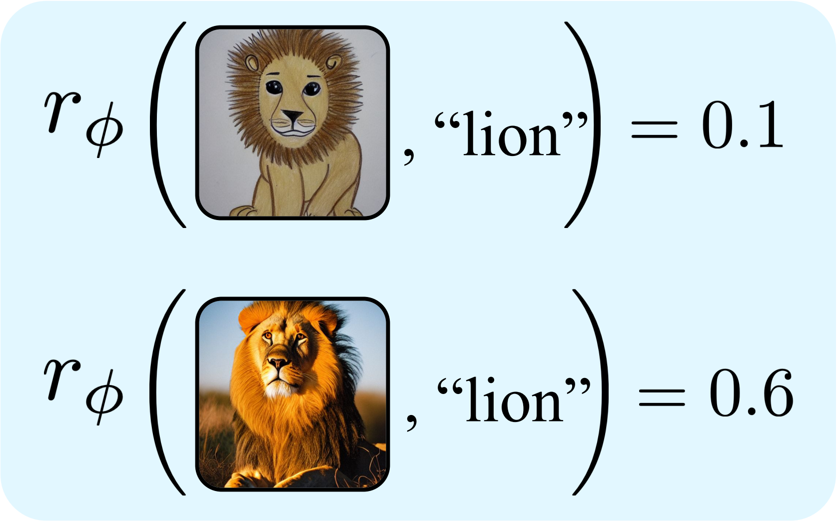
Human Preference Datasets

Human Preference Data



Human Preference Dataset (HPDv2) (Wu et al., 2023)
Pick-a-Pic Dataset (Kirstain et al., 2023)

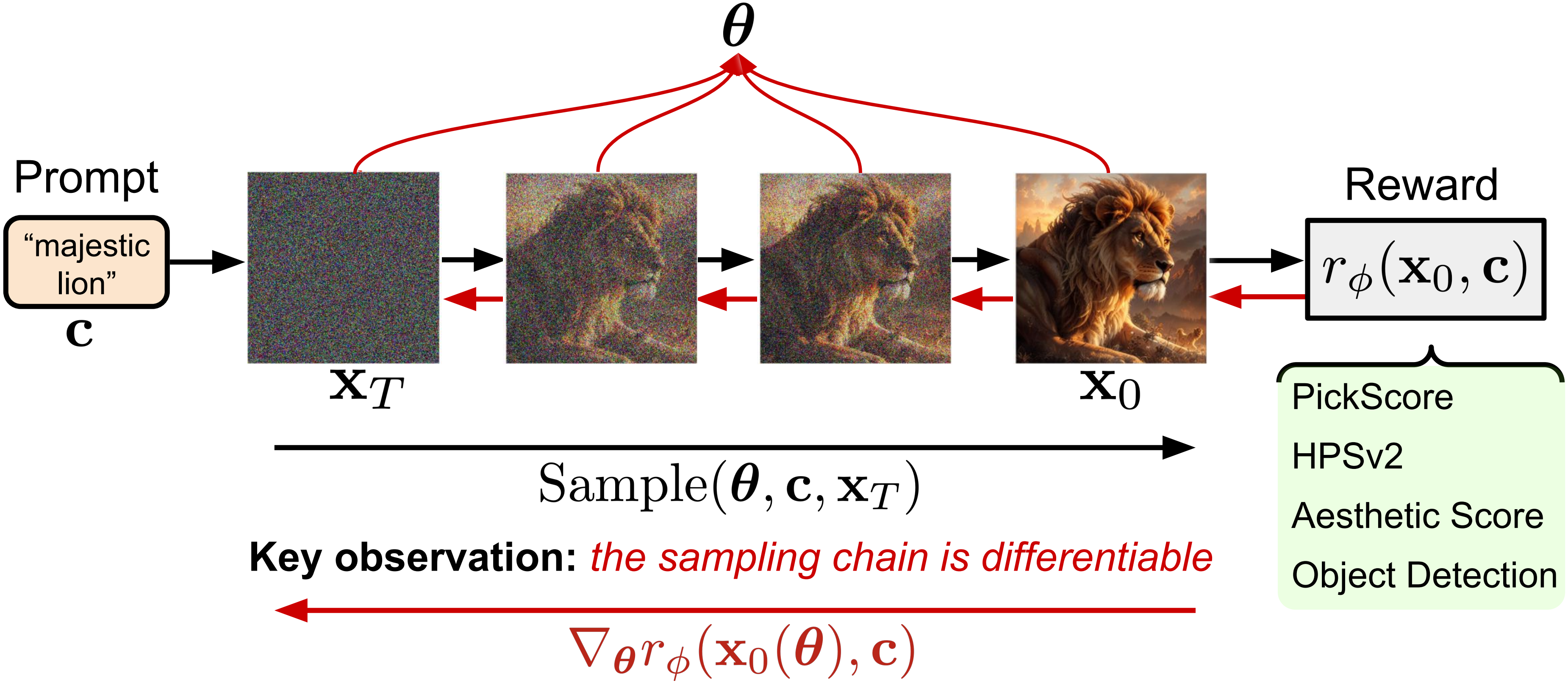
Reward Models



Prior work on diffusion fine-tuning used *RL-based techniques* (Black et al., 2023, Fan et al., 2023) → Promising results, but *sample-inefficient*

Direct Reward Fine-Tuning (DRaFT)

Objective: $J(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{c} \sim p_{\mathbf{c}}, \mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \left[r_\phi(\mathrm{Sample}(\boldsymbol{\theta}, \mathbf{c}, \mathbf{x}_T), \mathbf{c}) \right]$

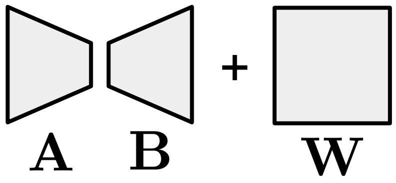


Memory Challenges

- **Main challenge:** *storing intermediate activations* through the unrolled sampling chain for use in backprop is expensive
- DRaFT uses two simple techniques to *keep memory usage tractable:*

(1) DRaFT fine-tunes *LoRA parameters* (Hu et al., 2022)

- *Reduces memory usage, yields smaller checkpoints*
- **Extra benefit:** Can interpolate between the original and fine-tuned model by re-scaling the LoRA parameters

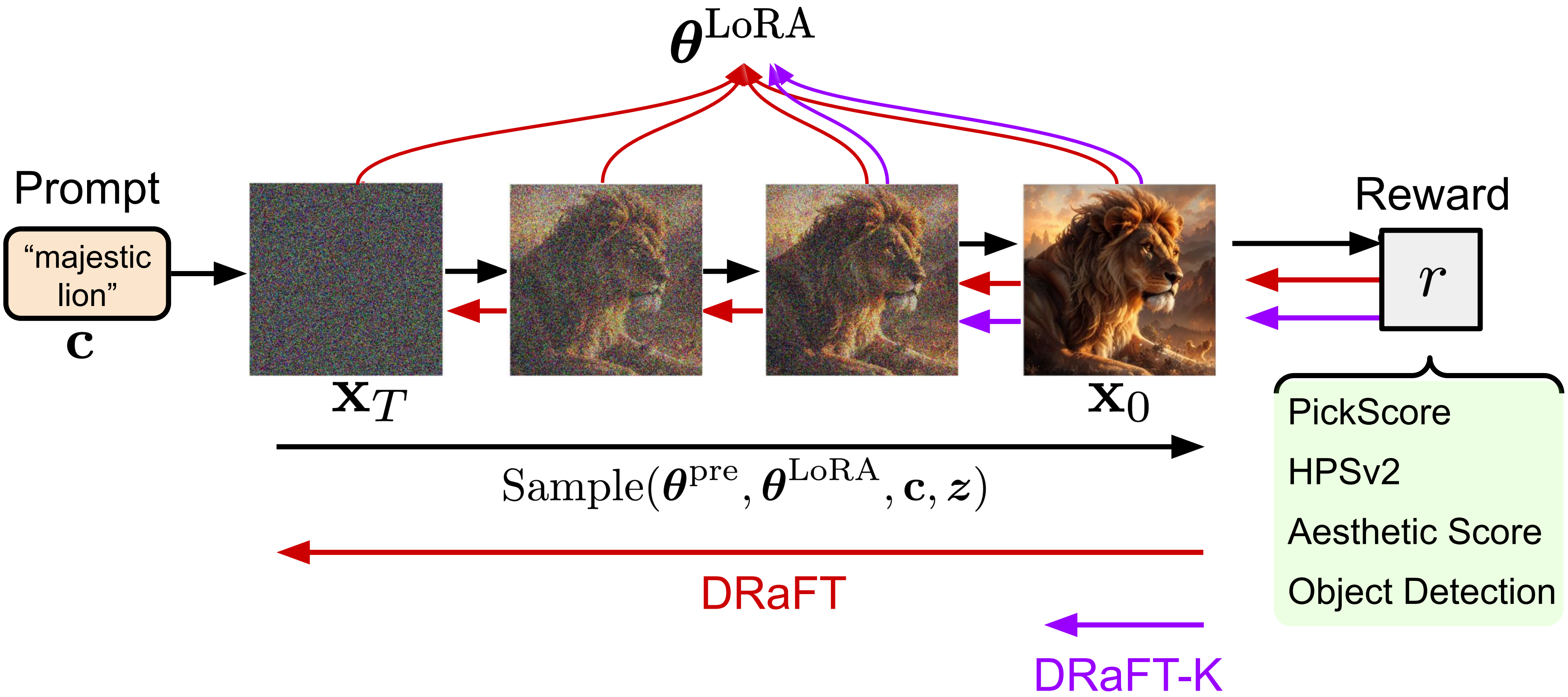


(2) *Gradient Checkpointing*

- Stores a subset of the intermediate activations in memory, and *recomputes non-stored ones on-the-fly during backprop*
- Only need to add one `@jax.checkpoint` to be able to compute the gradient

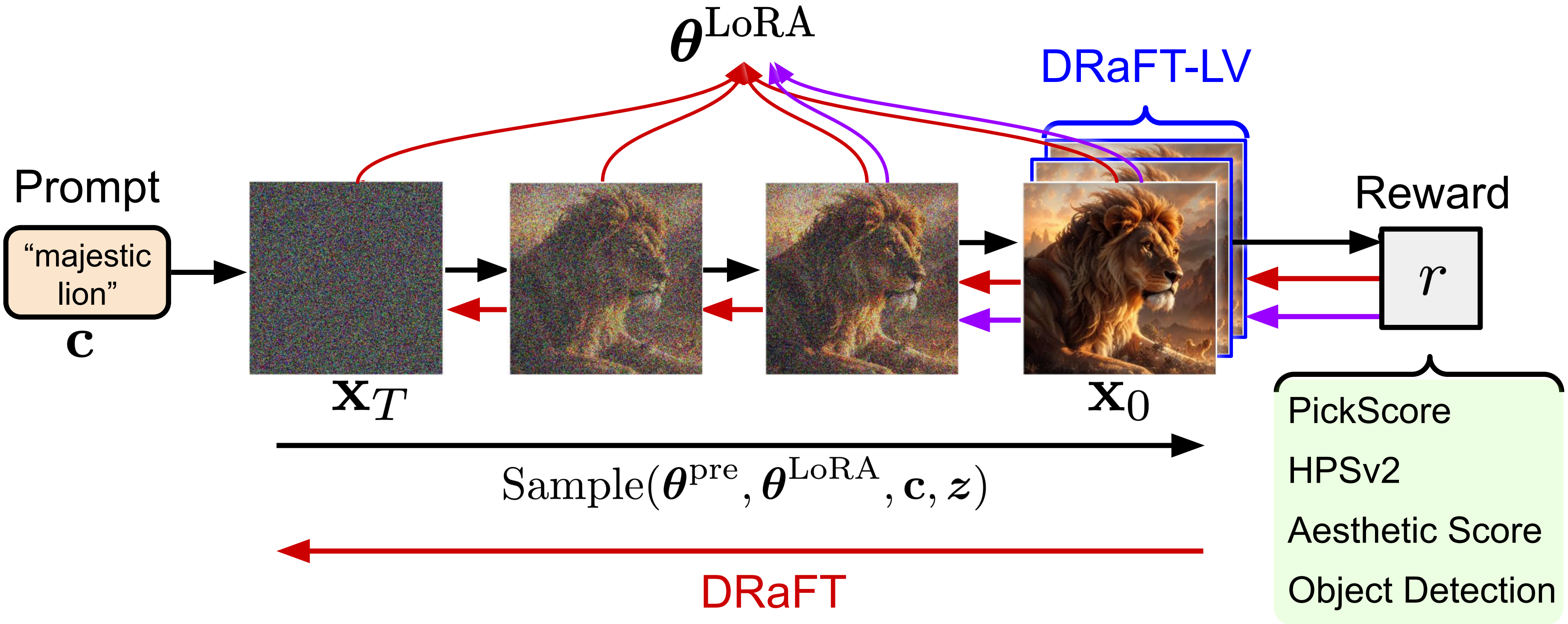
DRaFT + Efficiency Improvements

- **DRaFT-K:** Truncates backpropagation through *only the last K sampling steps*



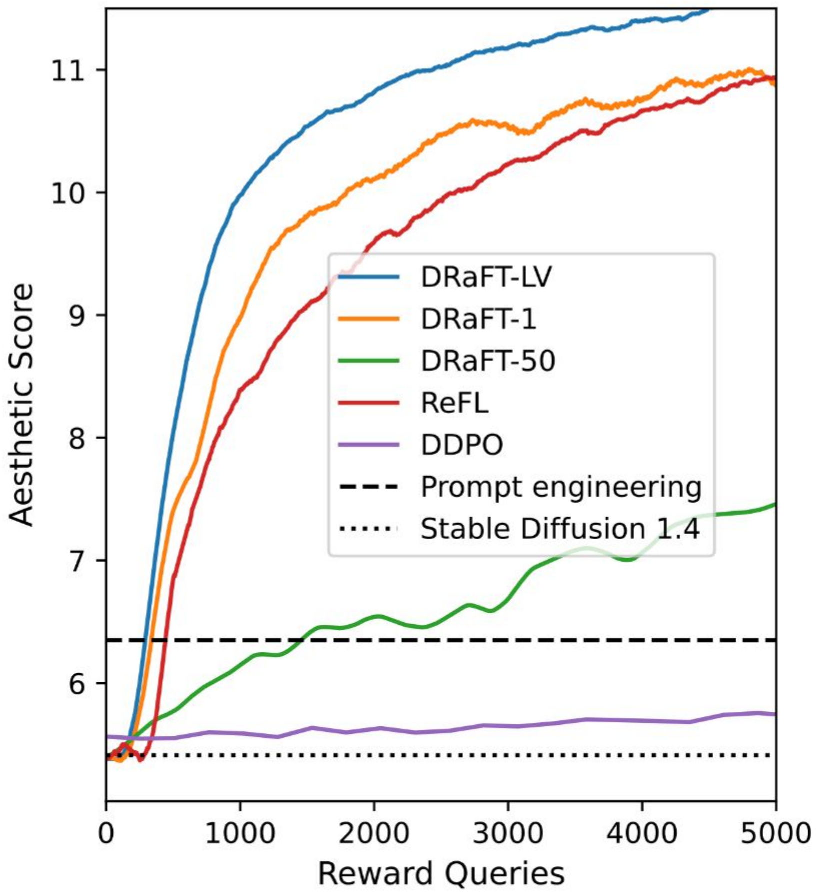
DRaFT + Efficiency Improvements

- **DRaFT-LV:** lower-variance gradient estimator. *Noise the generated image n times, and use the average reward gradient over these examples.*
- Using n=2 is around *2× more efficient than DRaFT-1* while adding around 10% overhead



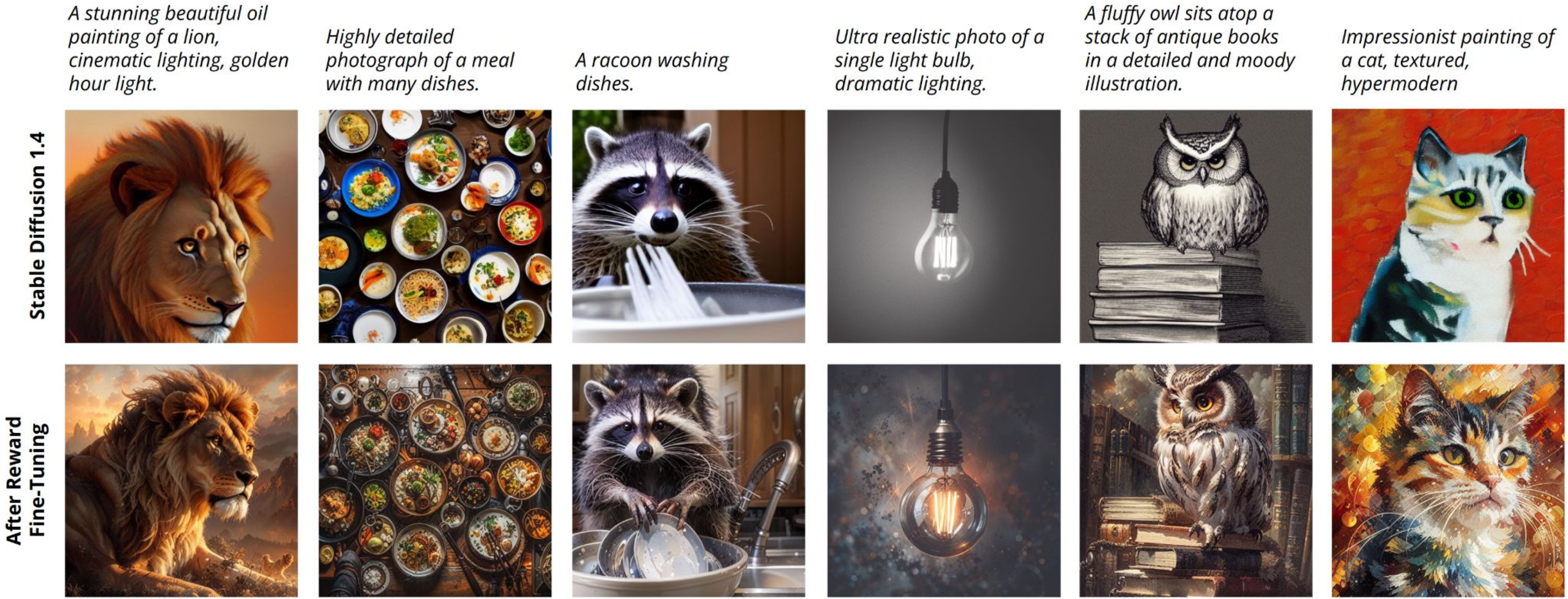
Quantitative Reward Optimization Comparison

- **Goal:** *optimize aesthetic quality scores of the LAION aesthetic predictor.*
- We compare against DDPO (Black et al., 2023), ReFL (Xu et al., 2023), and a prompt engineering baseline.
- *DRaFT is much more sample-efficient than RL*, as it leverages gradient information
- Because of its low-variance gradient estimate, *DRaFT-LV further improves training efficiency.*



DRaFT: Fine-Tuning for Human Preferences

- DRaFT using human preference reward models yields *more detailed and stylized images than baseline Stable Diffusion*



Scaling LoRA Parameters

Can *interpolate* between the original pre-trained model and the LoRA-adapted model by *scaling the LoRA weights*

Pre-trained model

$$\underbrace{\alpha = 0.6 \quad \alpha = 0.8 \quad \alpha = 0.9}_{\boldsymbol{\theta}^{\text{pre}} + \alpha\boldsymbol{\theta}^{\text{LoRA}}}$$

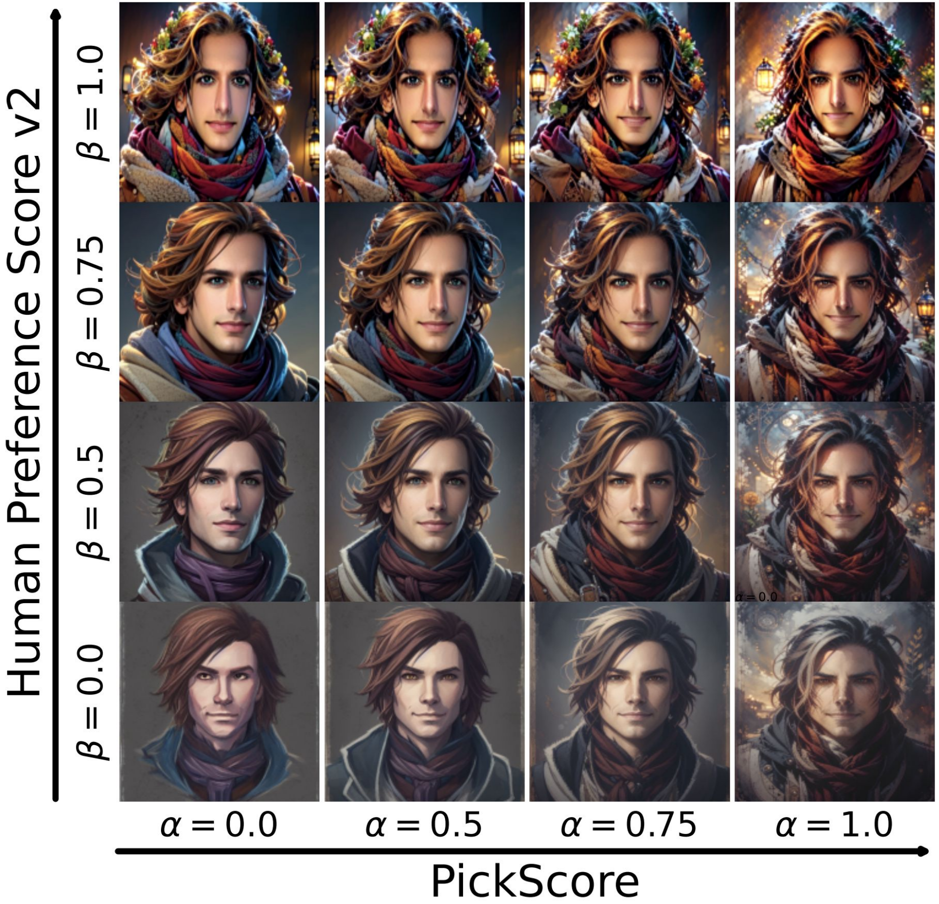
Fine-tuned model

$\alpha = 0.0$ | $\alpha = 0.6$ | $\alpha = 0.8$ | $\alpha = 0.9$ | $\alpha = 1.0$

Mixing LoRA Parameters

- LoRA parameters *fine-tuned independently for different rewards* can be combined post-hoc without additional training
- Taking *linear combinations of LoRA parameters*:

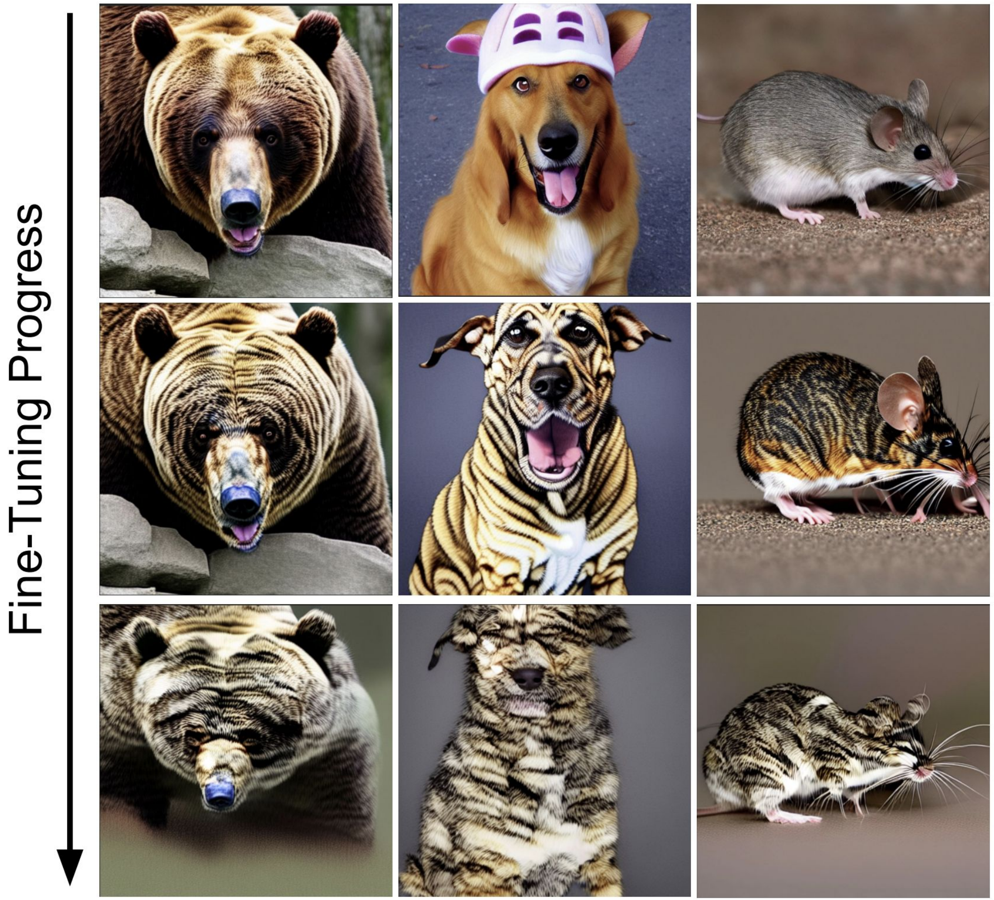
$$\boldsymbol{\theta}^{\text{pre}} + \alpha\boldsymbol{\theta}^{\text{LoRA}}_{\text{PickScore}} + \beta\boldsymbol{\theta}^{\text{LoRA}}_{\text{HPSv2}}$$



Object Detection for Addition and Removal

- Well-crafted reward functions can allow us to specify what kinds of images we wish to generate *even when no real images exist that satisfy the criteria*

Diffusion Adversarial Examples



- Fine-tuning a diffusion model such that *images generated based on prompts {“bear”, “dog”, “mouse”} are classified as target class “cat”* by a ResNet-50 classifier.
- This classifier is *texture-biased*, as the fine-tuned images have cat-like textures while keeping the animal shapes mostly unchanged.

Conclusion

- DRaFT is an efficient framework for *fine-tuning diffusion models on differentiable rewards by leveraging reward gradients.*
    - DRaFT is substantially *more efficient than RL-based fine-tuning approaches*
- We applied DRaFT to a diverse array of reward functions
    - Human preference rewards, object detection, classification, and more
- Just as RLHF has become crucial for deploying LLMs, *reward fine-tuning may become a key step for improving image generation models*.

Thank you!