Adding Conditional Control to Pre-trained Diffusion Models: A Reinforcement Learning Approach

Yulai Zhao
Intern at DELTA, BRAID
Mentored by Ehsan Hajiramezanali, Masatoshi Uehara, Gabriele Scalia

July 29th, 2024

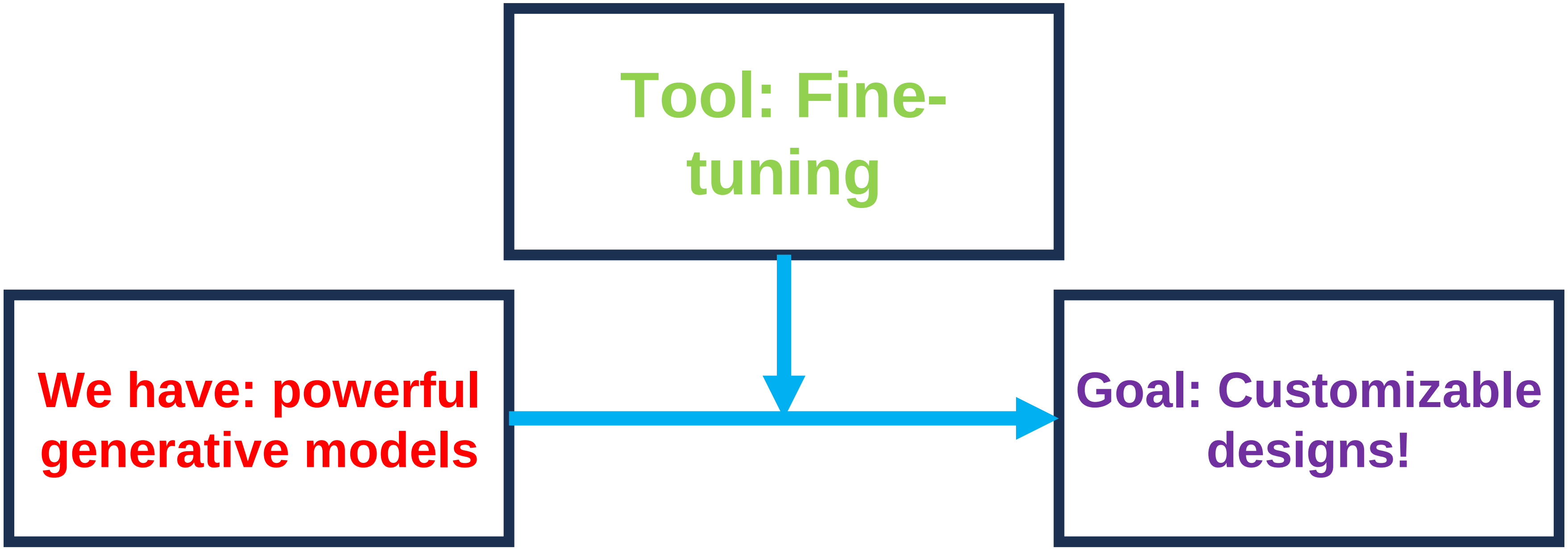
Acknowledgements

- A preprint paper is released at https://arxiv.org/abs/2406.12120

  "Adding Conditional Control to Diffusion Models with Reinforcement Learning"

- Authors: Yulai Zhao*, Masatoshi Uehara*, Gabriele Scalia, Tommaso Biancalani, Sergey Levine, Ehsan Hajiramezanali

General roadmap: AI-aided design



Examples - Images

Pre-trained generative models

Fine-tuning

Images customized on their aestheticness, layout or compressibility

In this work, our objective is different from merely optimizing towards certain rewards!

Standard fine-tuning: towards reward models

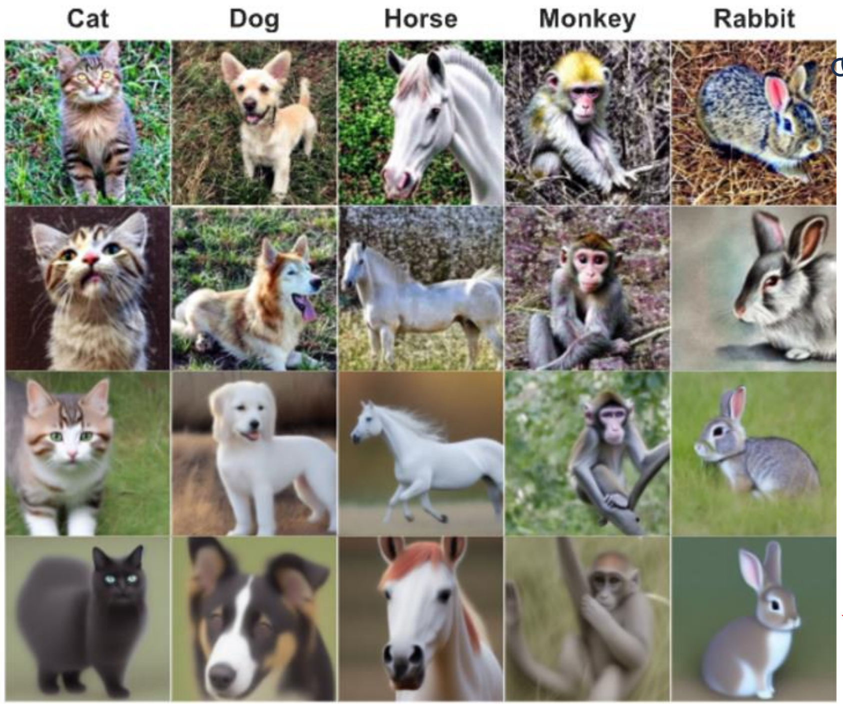
Pre-trained generative models

Improved aestheticness

Fine-tune towards aesthetic socres

Our task is different, and harder

Adding a new condition!

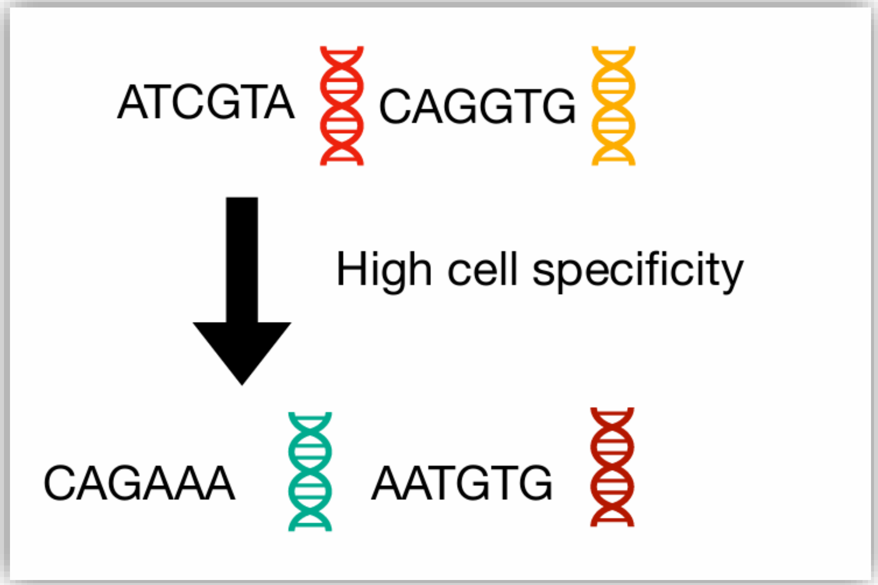


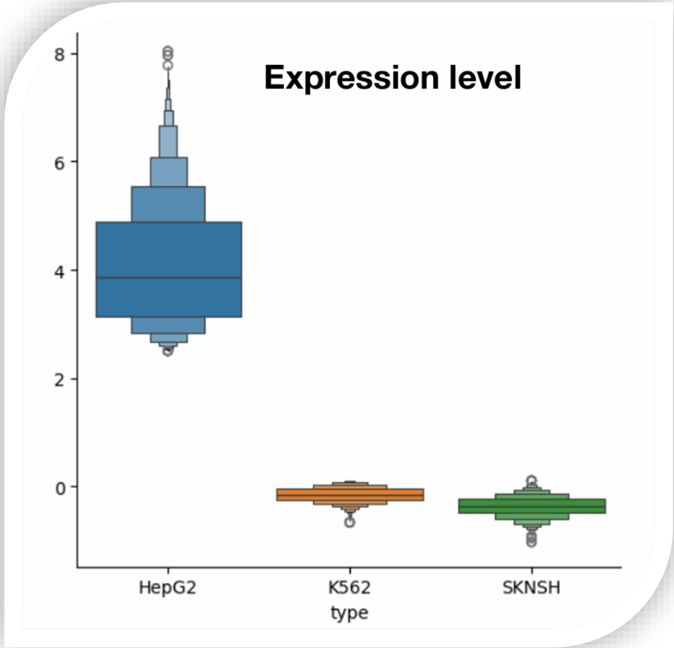
All generated by one model!

Arbitrary compressible levels

Examples – Sequence design

DNA enhancers

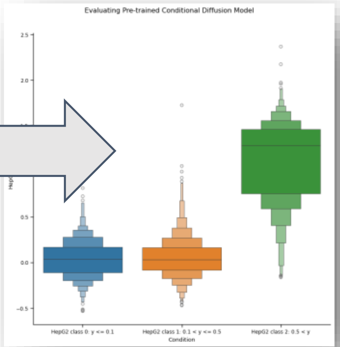


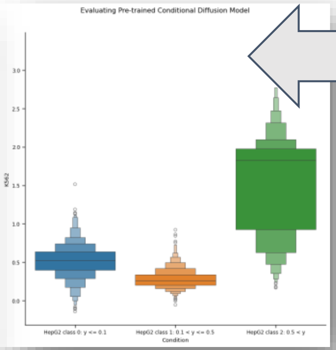


<u>Goal: Highly active in just one cell line</u>

Why our task is important?

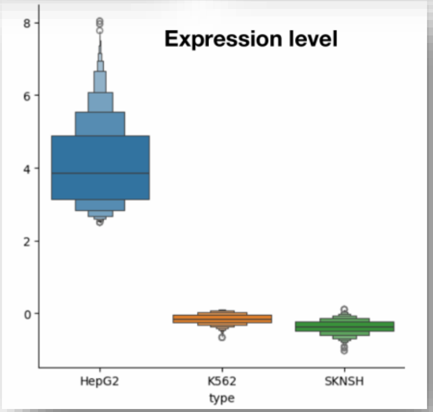
Condition on high activity level in HepG2





Also active in other cell lines, no specificity

If we can add "conditional control"

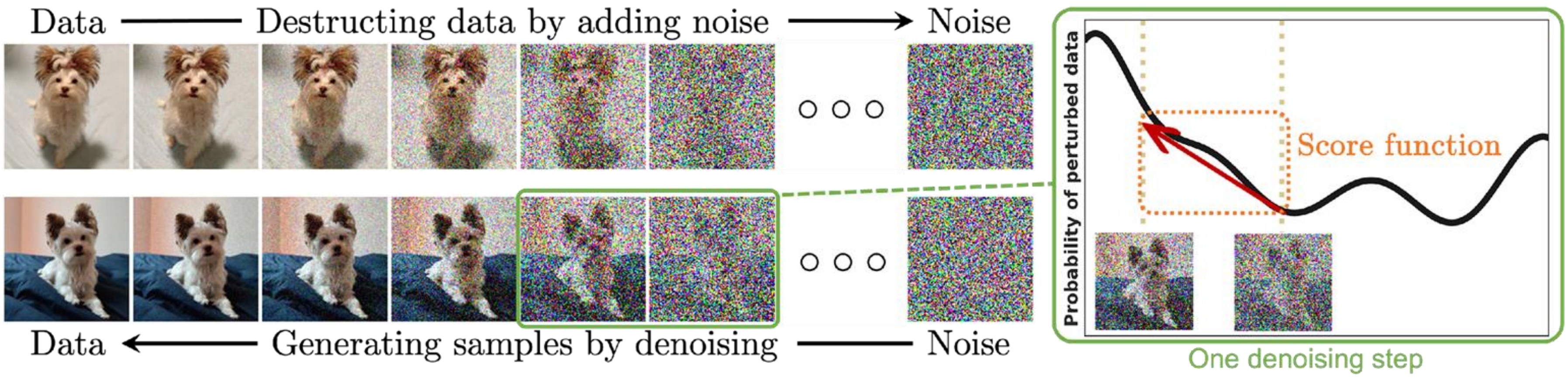


Can achieve specificity!

Contents

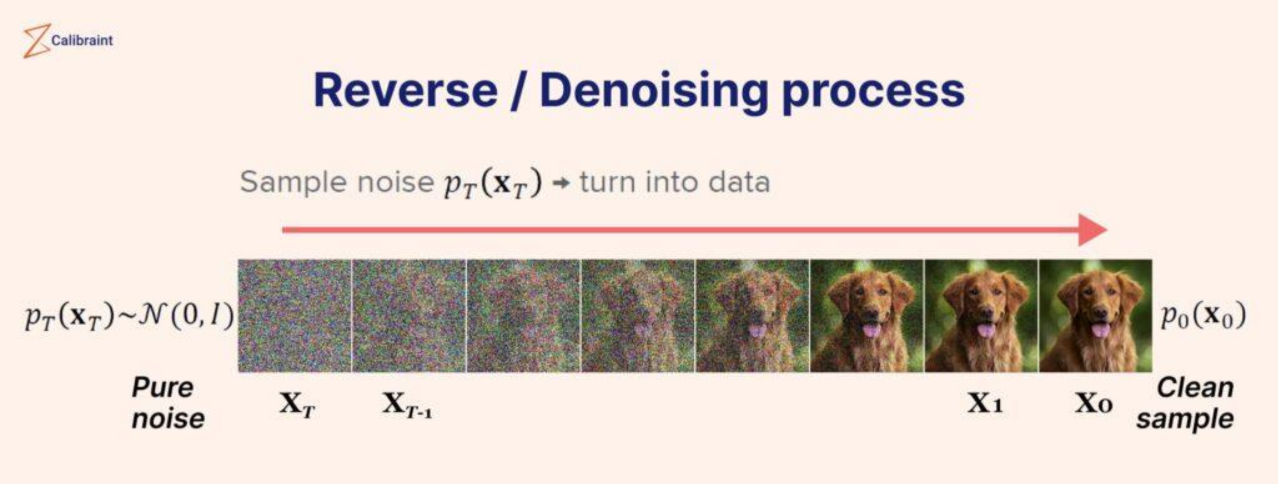
Diffusion Models

Forward process: adding noise



Denoising process: generation

How to train diffusion models?



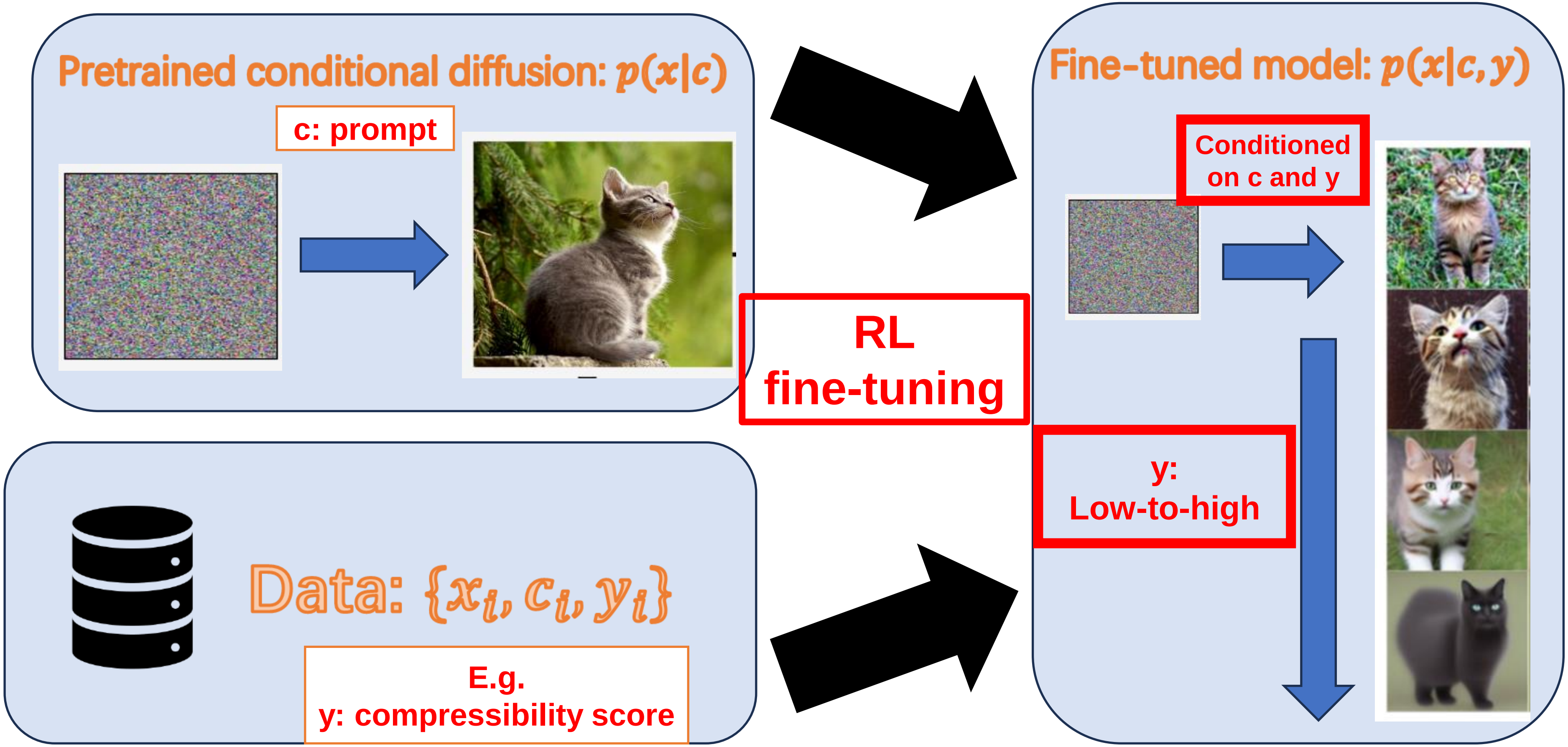
Data:
Many x_i, following $p^{pre}(x)$

Goal:
Learn θ from data, such that $x_T \sim p^{pre}$

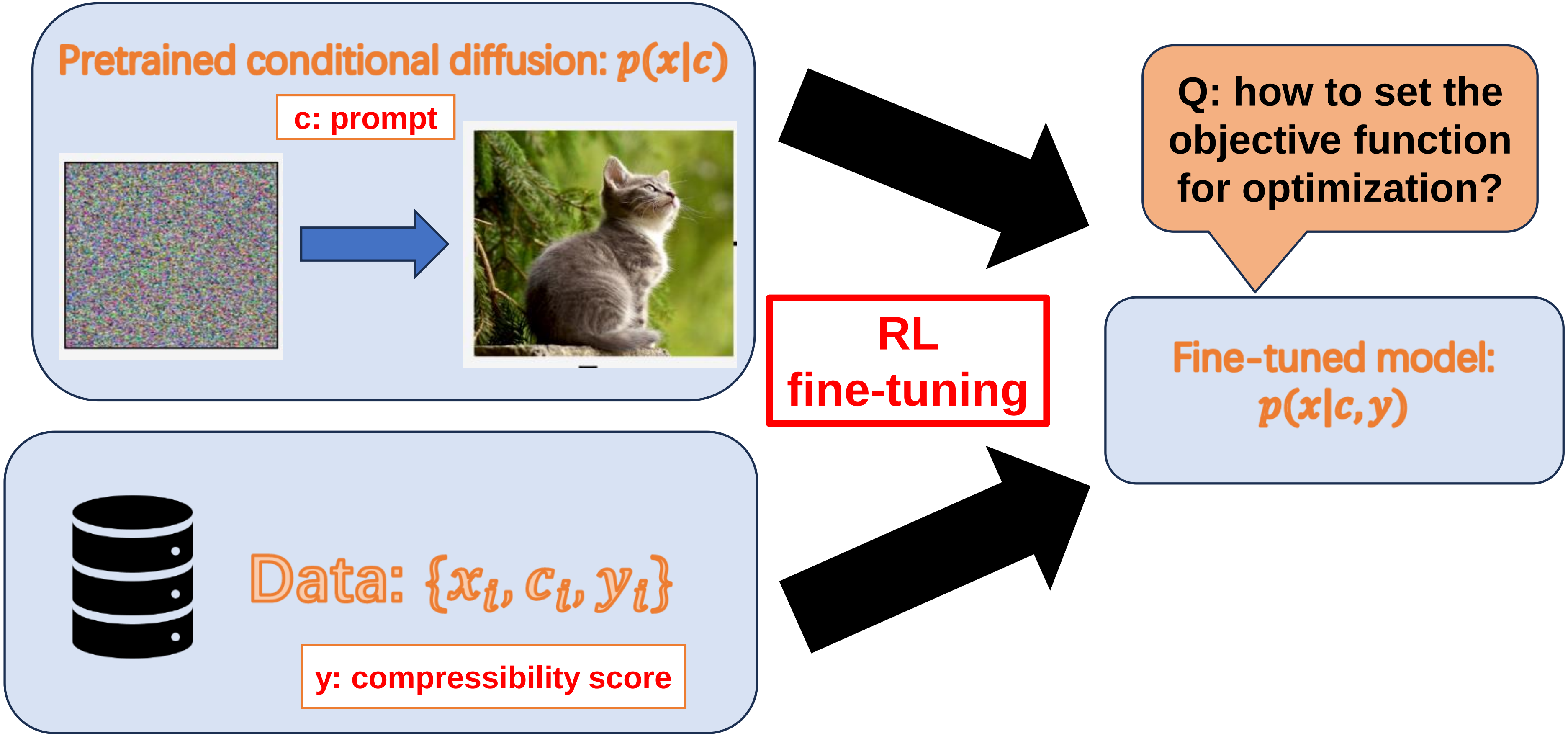
Denoising process:

$$d\,x_t = f(t, x_t, \theta) + \sigma\, dw_t$$

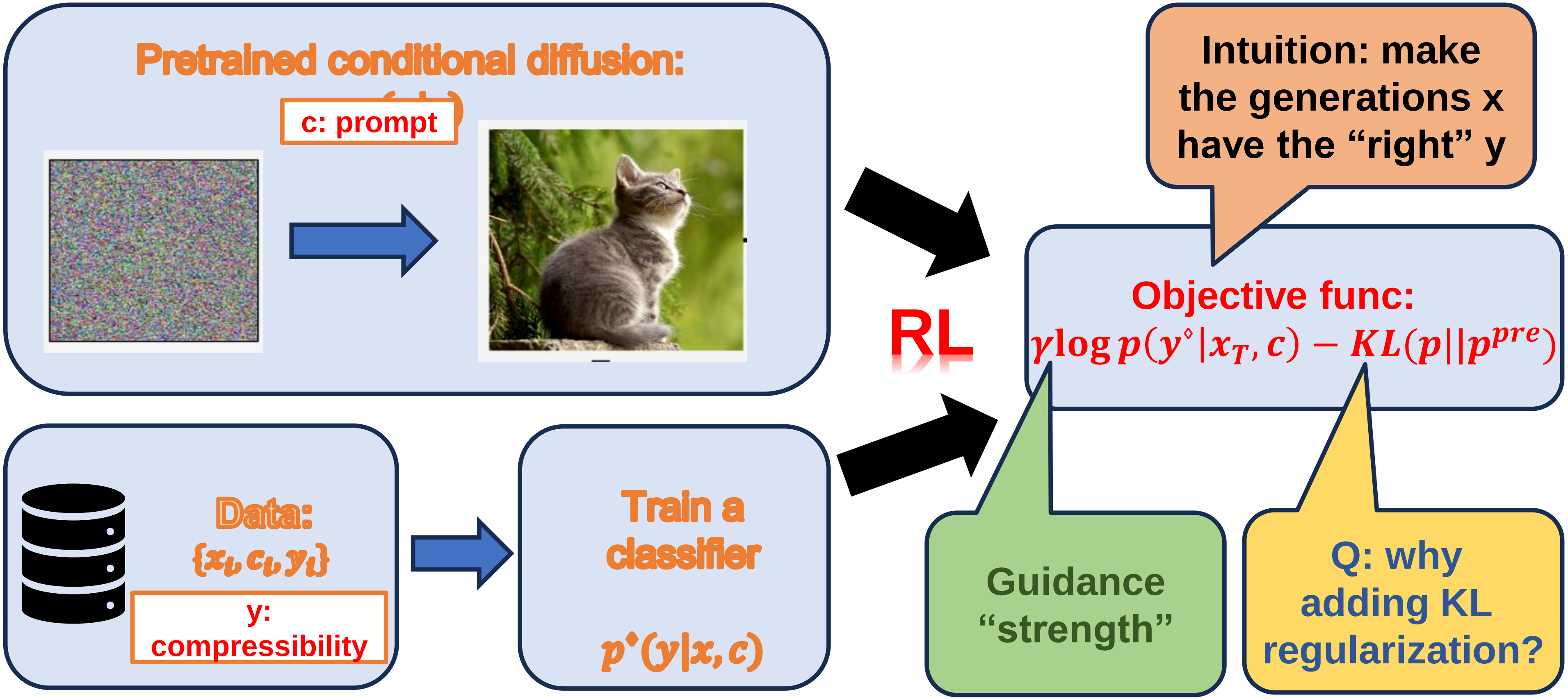
Our goal: adding control via fine-tuning



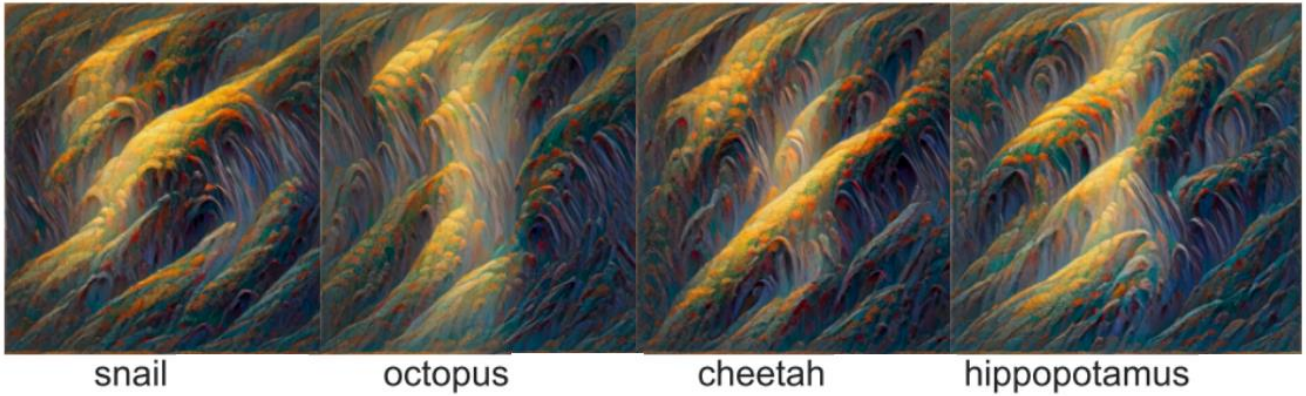
Our goal: adding control via fine-tuning



Methodology



Reward collapse

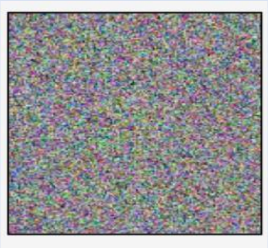


- Very common in generative systems (DMs, LLMs, GANs)
- Generations have low diversity
- Because the oracle is "over-optimized"

Methodology – Cont.

Pretrained conditional diffusion:

c: prompt



Data:
$\{x_i, c_i, y_i\}$

y: compressibility

Train a classifier:
$p^{\diamond}(y|x, c)$

RL

Objective func:
$\log p^{\diamond}(y|x_T, c) - KL(p||p^{pre})$

KL avoids being too far from the pre-trained model!

Theoretical justification (incomplete)

In fine-tuning

Objective func:

$$\gamma \log p(y|x_T, c) - KL(p||p^{pre})$$

Fine-tuned model

$p(x|y,c)$

$$p(x|y,c) \propto \left(p^{\diamond}(y|x,c)\right)^{\gamma} \cdot p^{pre}(x|c)$$

Additional guidance

Represents 'deviation' from pre-trained model

Advantage: leverage conditional independency

Example 1: $Y \perp C \mid X$

$p^{\diamond}(y|x,c) = p^{\diamond}(y|x)$

We only need (x, y) data.

extend to multi-task

Example 2:
$Y_1 \perp Y_2 \mid C, X$ and $Y_1 \perp C \mid X$ and $Y_2 \perp C \mid X$

$\log p^{\diamond}(y_1, y_2|x,c) = \log p^{\diamond}(y_1|x) + \log p^{\diamond}(y_2|x)$

We only need (x, y_1) and (x, y_2) data.

Classifier-free baseline needs (x,c,y) and (x,c,y1, y2) !

Experiments

Example 1: $Y \perp C \mid X$

1. Make 4 levels of **compressibility**
2. Add compressibility as an additional control to Stable Diffusion

We validate both examples!

Example 2:
$Y_1 \perp Y_2 \mid C, X$ and $Y_1 \perp C \mid X$ and $Y_2 \perp C \mid X$

Y_1: compressibility, Y_2: aesthetic scores

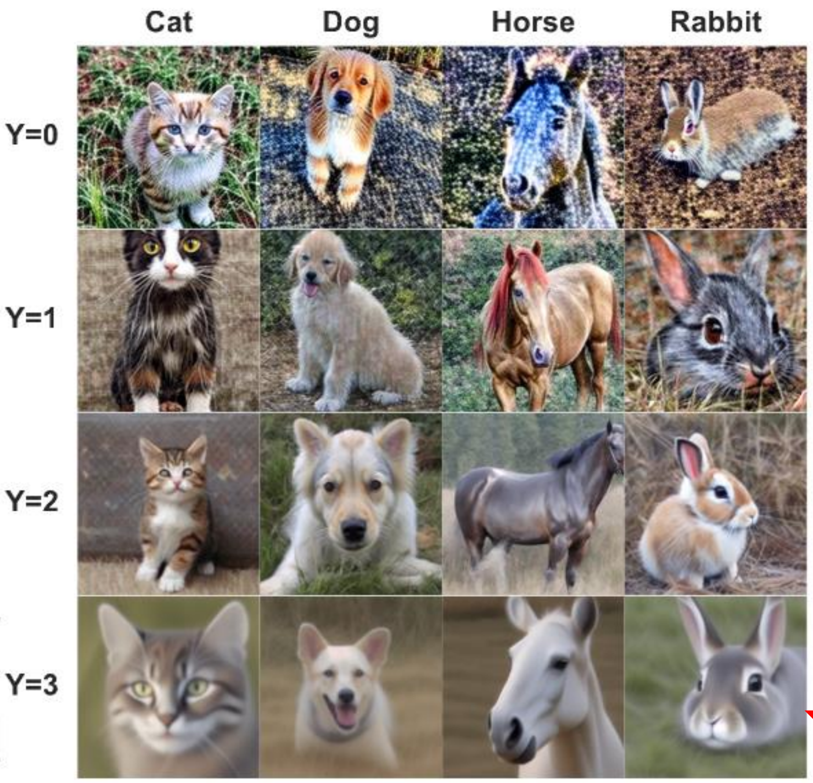
1. Both scores are divided into 2 levels.
2. Together it reduces to a 2X2 multi-task generation.

Example 1: Compressibility

(a) Training curve

	Accuracy ↑	Macro F1 score ↑
DPS	0.45	0.44
CTRL (Ours)	**1.0**	**1.0**

(b) Evaluation of conditional generations



(c) Generated images

more compressible

More Qualitative Results

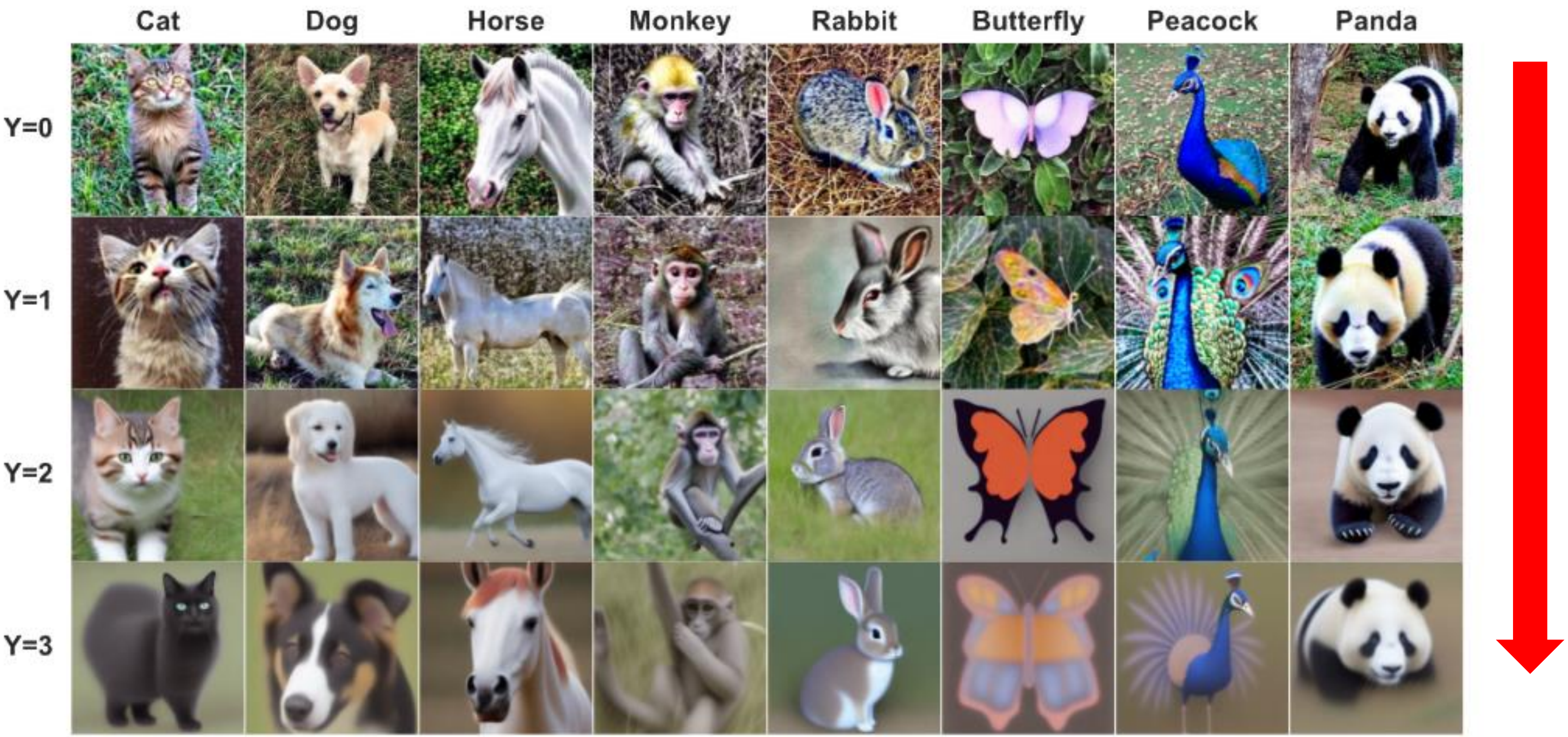
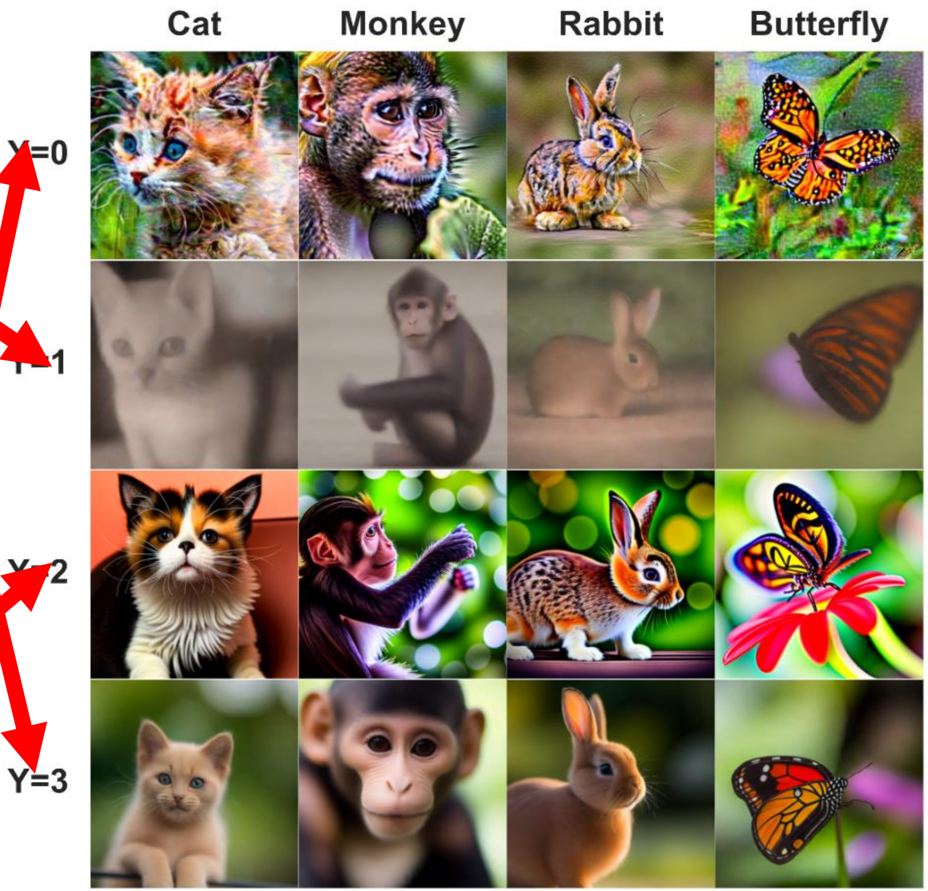


Figure 4: More images generated by **CTRL** in the compressibility task.

Example 2: Compressibility & Aestheticness

High compressibility

Low compressibility



Low aestheticness

High aestheticness

More Qualitative Results

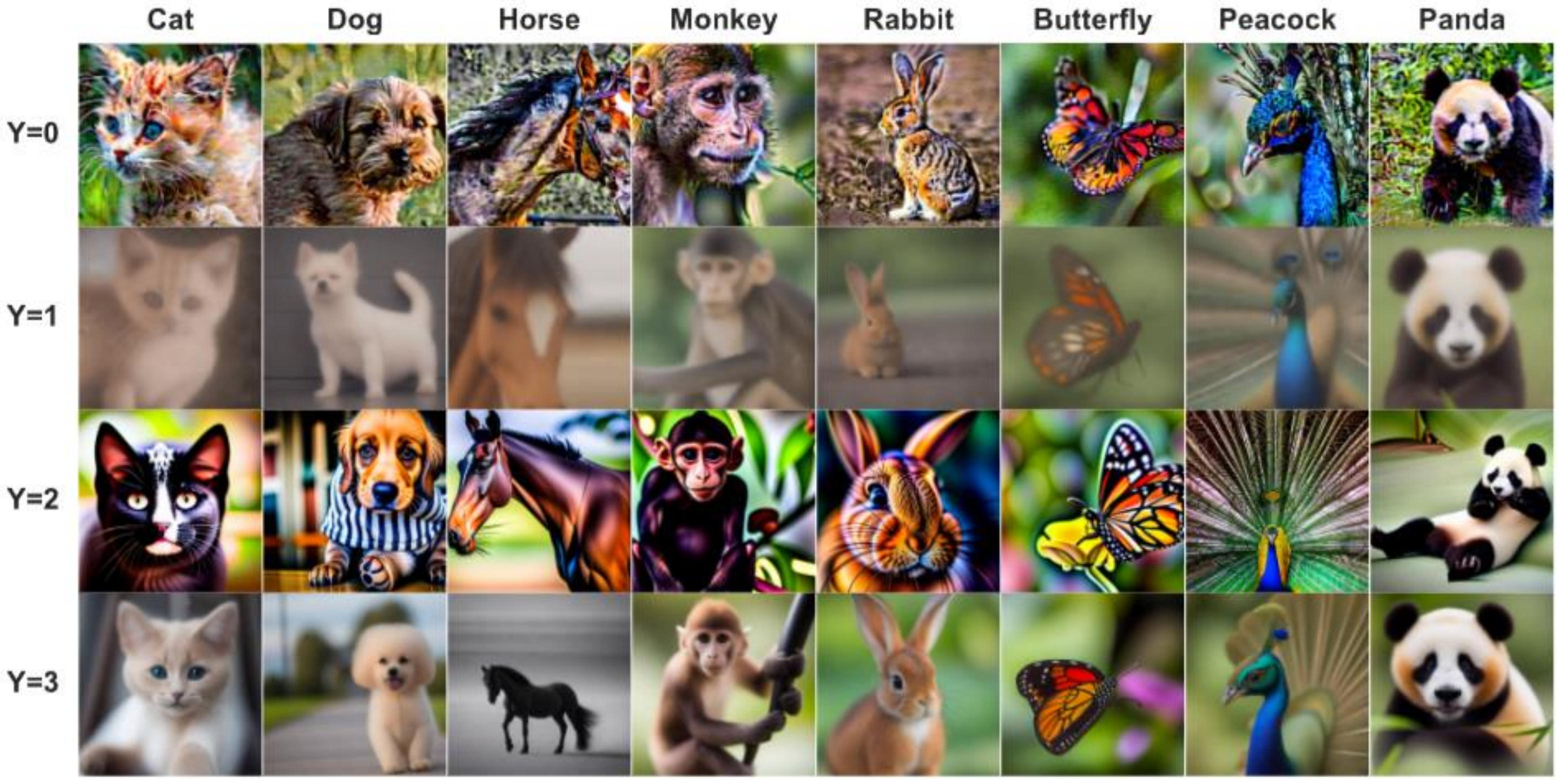
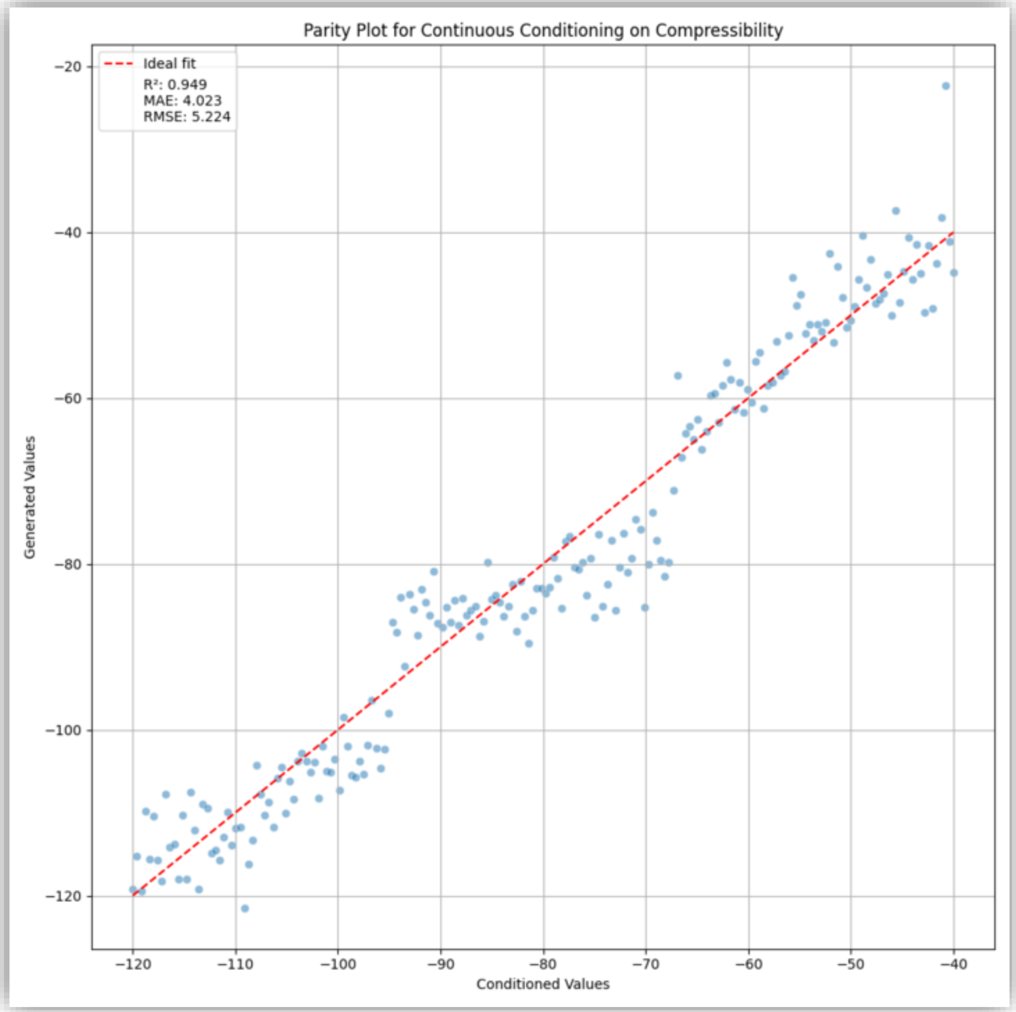


Figure 5: More images generated by **CTRL** in the multi-task conditional generation.

Extension: adding condition on a continuous $\boldsymbol{y}$



Compare with baselines

- **Classifier guidance**: training a classifier and incorporating its gradients to guide inference (while freezing pre-trained models)
- **Classifier-free guidance**: directly conditions the generative process on both data and context, bypassing the need for explicit classifiers.

Table 1: Comparison between our proposal and existing approaches. In contrast to classifier guidance or its variations, our method entails re-training the models directly on top of pre-trained models (i.e., fine-tuning). Additionally, we circumvent the necessity of learning a mapping $x_t \to y$ or employing heuristic approximation techniques to address this issue. Compared to classifier-free guidance which always demands triplets $\{c, x, y\}$, our method can leverage conditional independence and only necessitate pairs $\{x, y\}$ by leveraging if $y \perp c | x$ holds. This simplifies the construction of the offline dataset.

Methods	Fine-tuning	Need to learn $x_t \to y$	Leveraging conditional independence
Classifier guidance (Dhariwal and Nichol, 2021)	No	Yes	Yes
Reconstruction guidance (e.g. (Ho et al., 2022), (Chung et al., 2022), (Han et al., 2022))	No	No	Yes
Classifier-free guidance (Ho and Salimans, 2022)	Yes	No	No
CTRL (Ours)	Yes	No	Yes

Conclusion

- We introduce an **RL-based fine-tuning** approach for conditioning pre-trained diffusion models on new additional labels.
- Compared to **classifier-free guidance**, our proposed method uses the offline dataset more efficiently and allows for **leveraging the conditional independence** assumption, thereby greatly simplifying the construction of the offline dataset.
- We also theoretically justify our approach and **build the connection with classifier-based guidance**.

Thank you!

ArXiv