Extracting Reward Functions from Diffusion Models

Felipe Nuti*

Tim Franzmeyer*

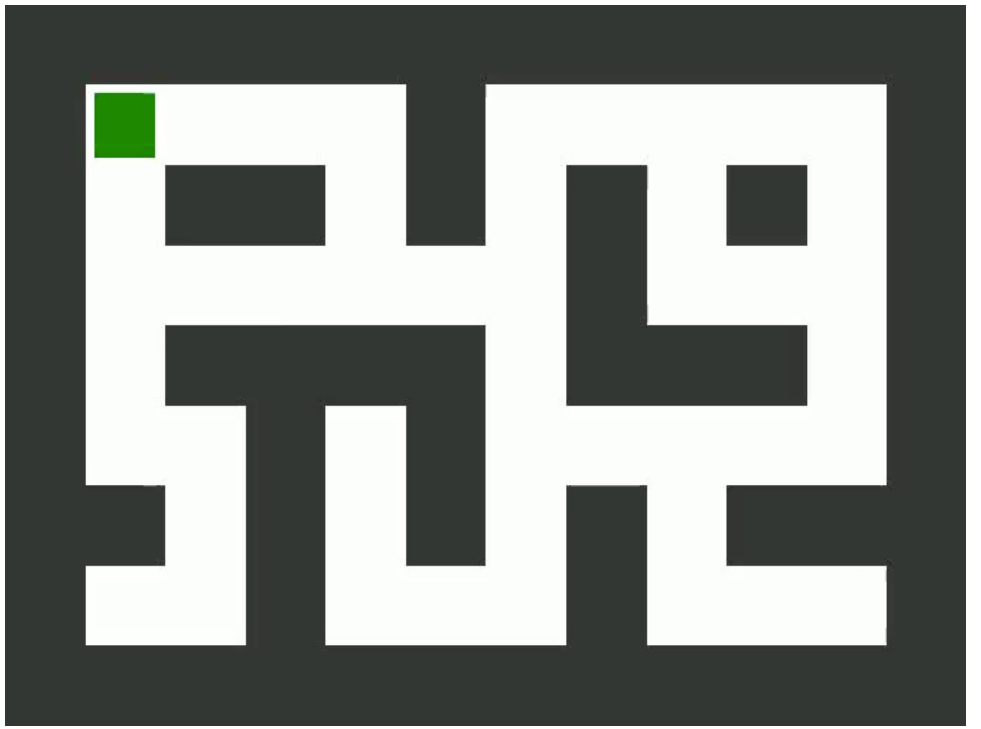
João F. Henriques

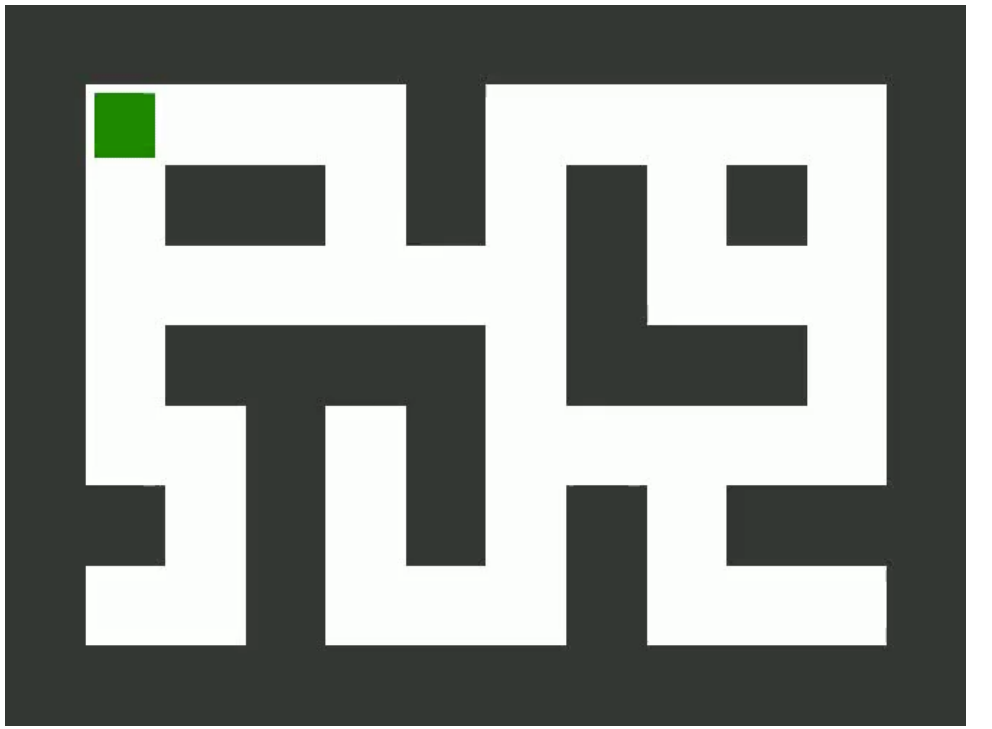
UNIVERSITY OF OXFORD

*equal contribution

NEURAL INFORMATION PROCESSING SYSTEMS

Running example: Maze2D

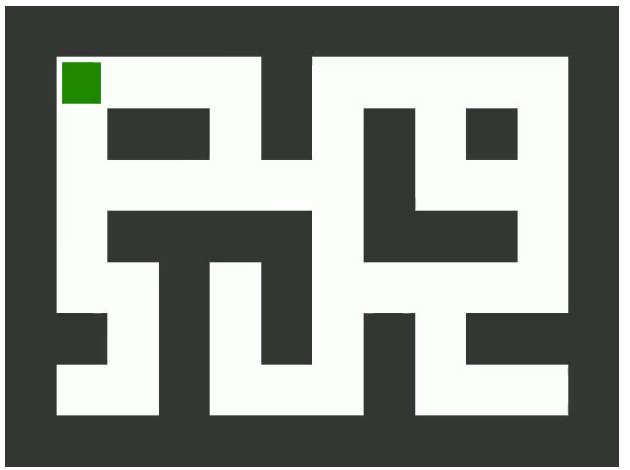




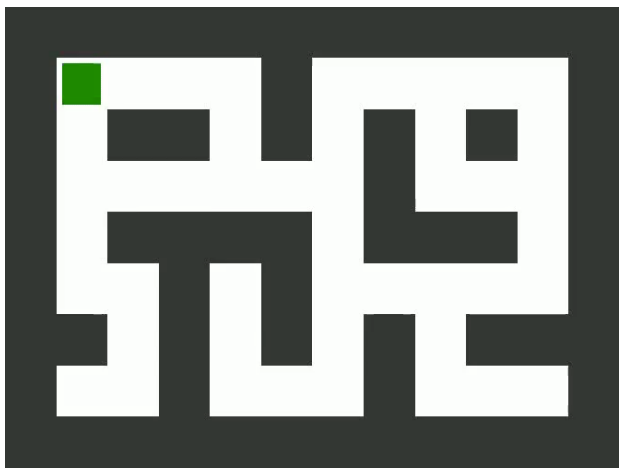
From *Fu, Justin, et al. "D4rl: Datasets for deep data-driven reinforcement learning." arXiv preprint arXiv:2004.07219 (2020).*

Diffusing trajectories: Training

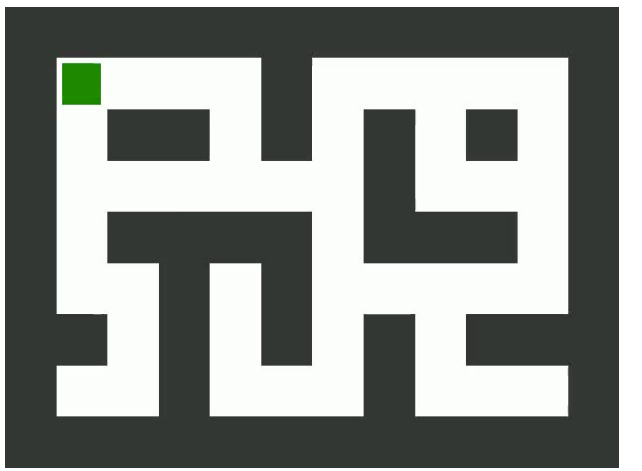
Method of Janner et al. 2022



Diffusion step t=20



Diffusion step t=5

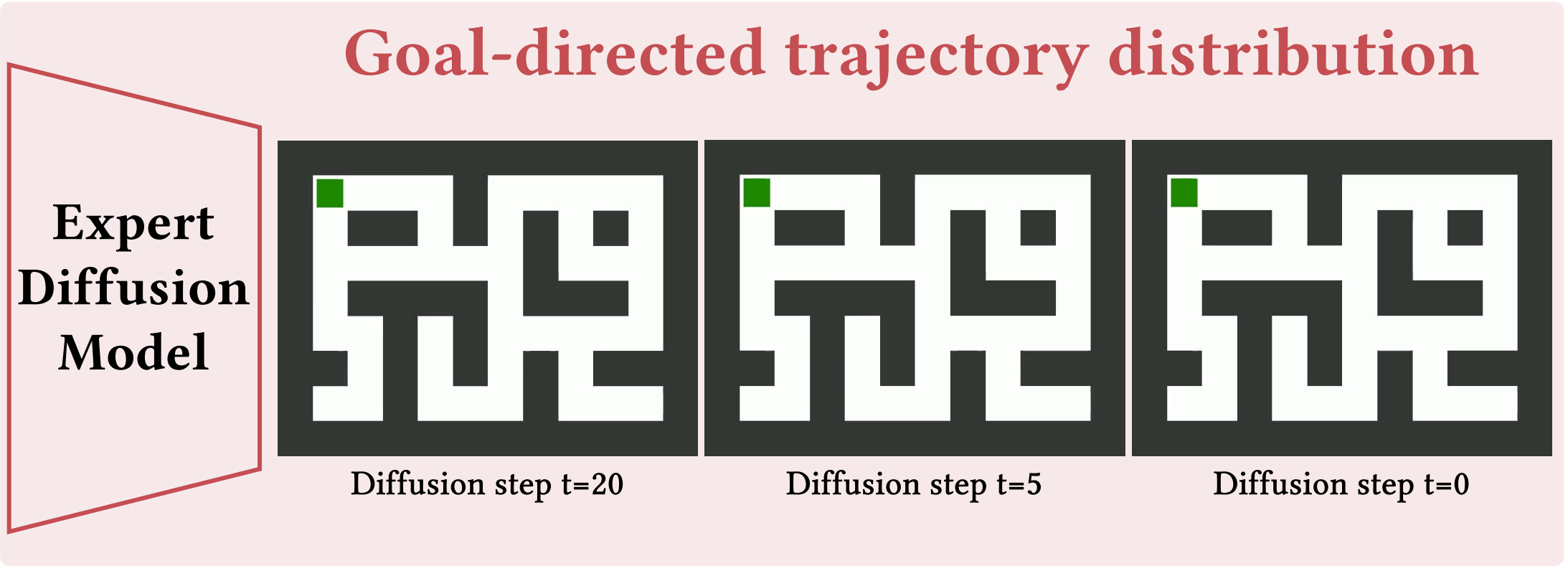


Diffusion step t=0

Janner, Michael, et al. "Planning with Diffusion for Flexible Behavior Synthesis." International Conference on Machine Learning. PMLR, 2022.

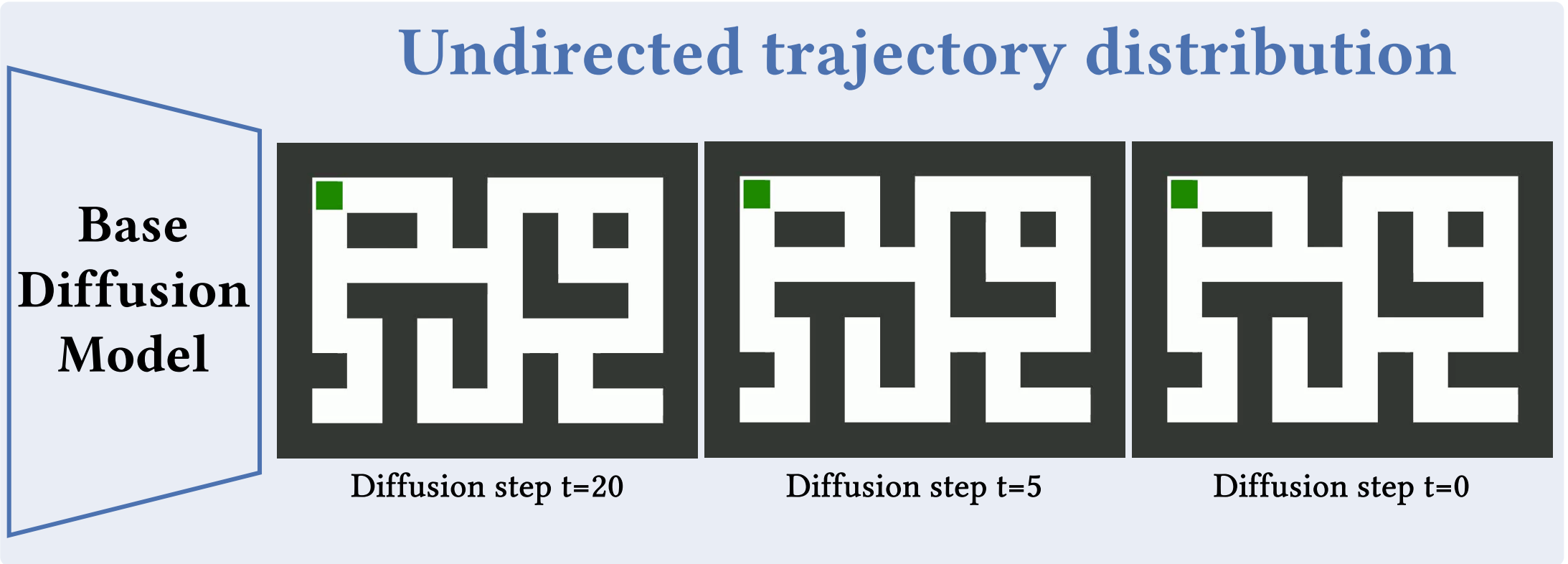
Diffusing trajectories: Inference

Method of Janner et al. 2022

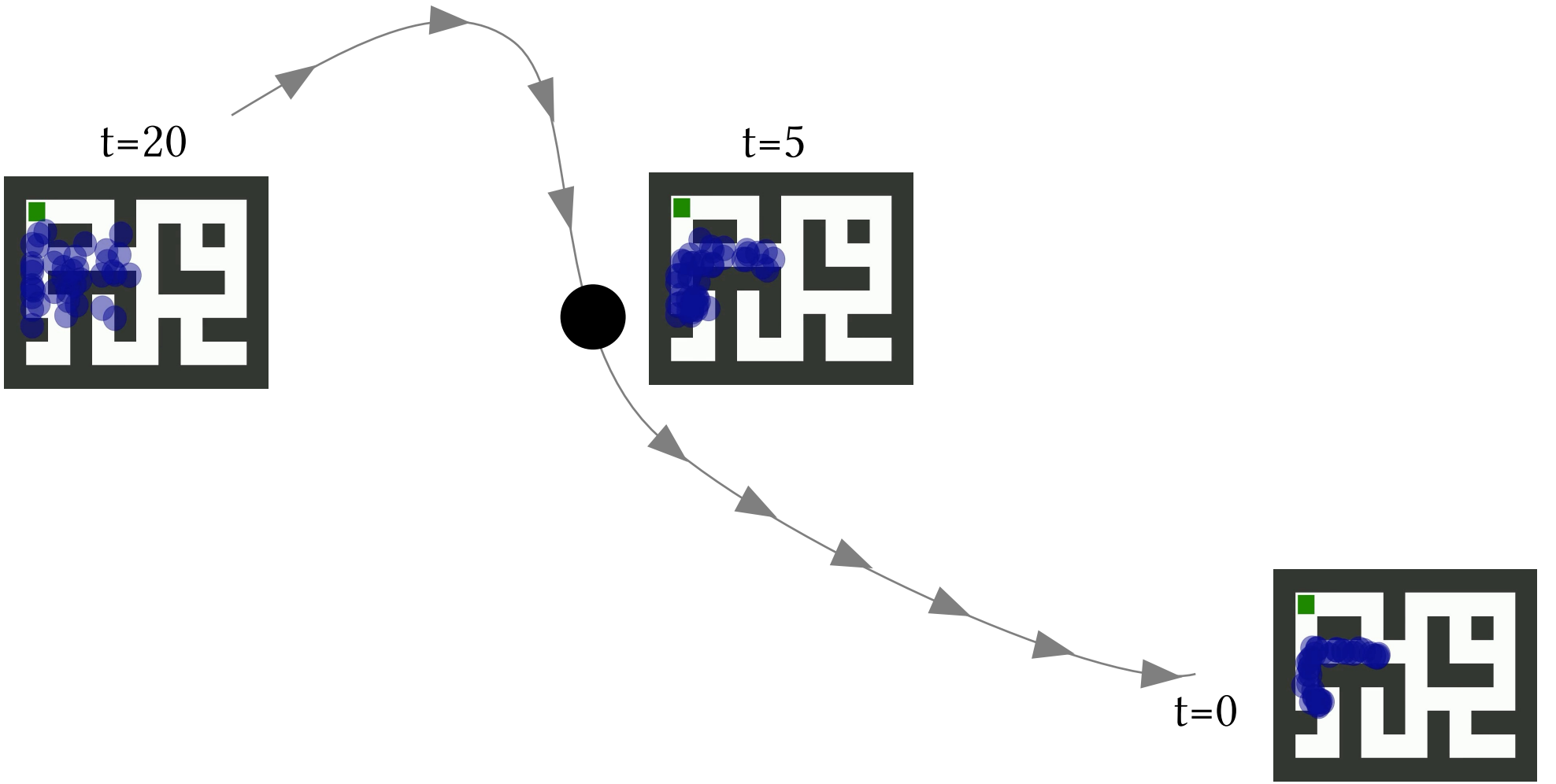


Diffusing trajectories: Inference

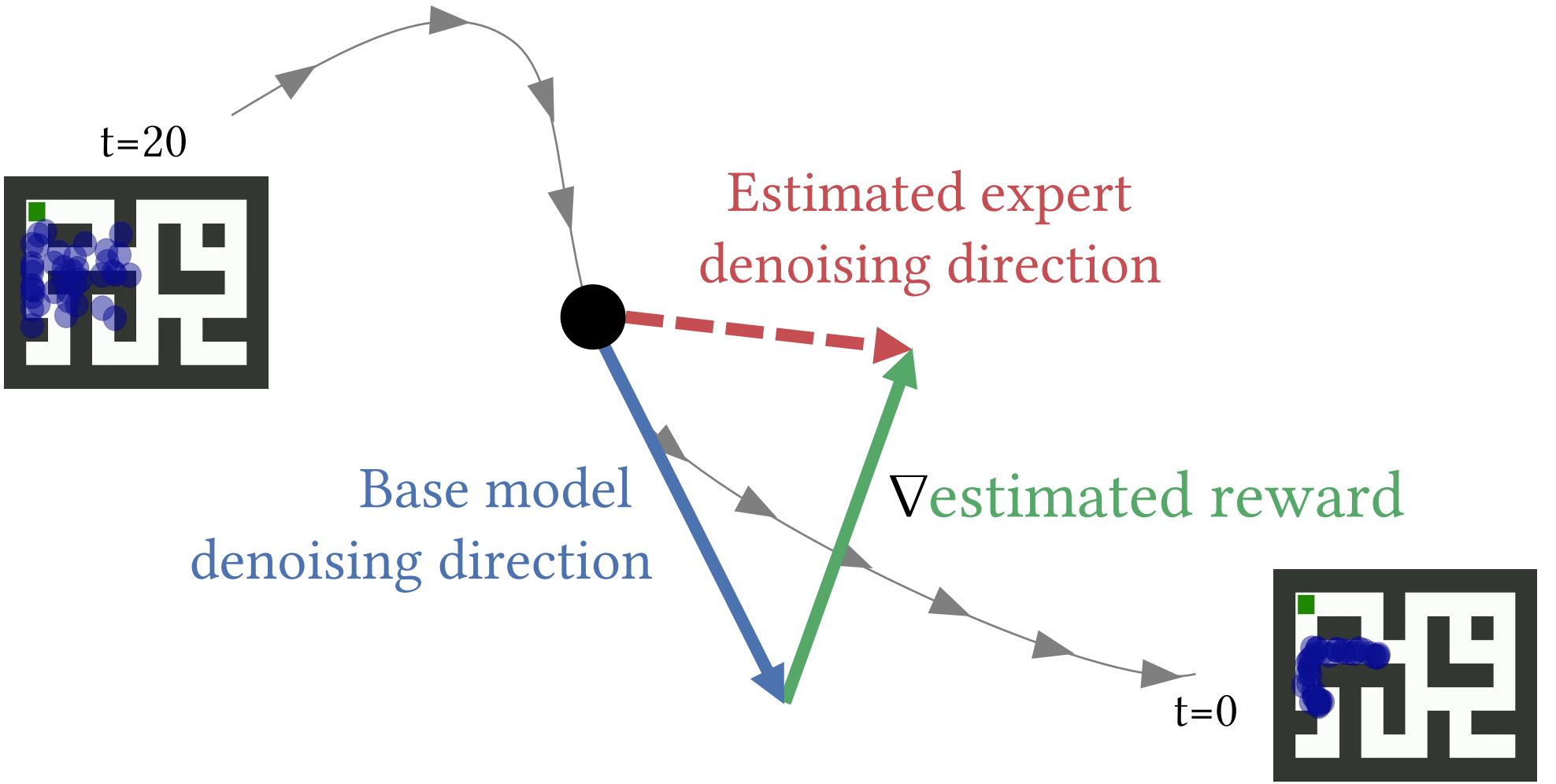
Method of Janner et al. 2022



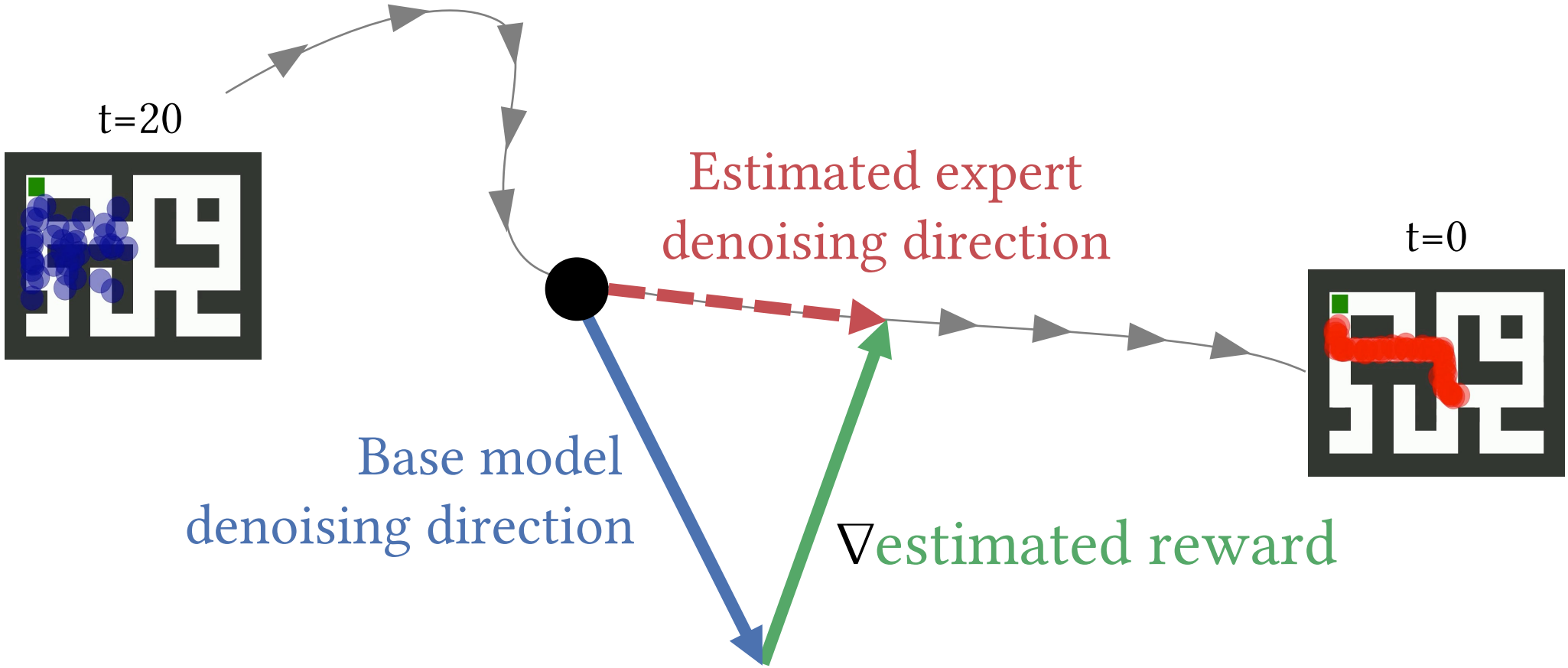
Denoising process



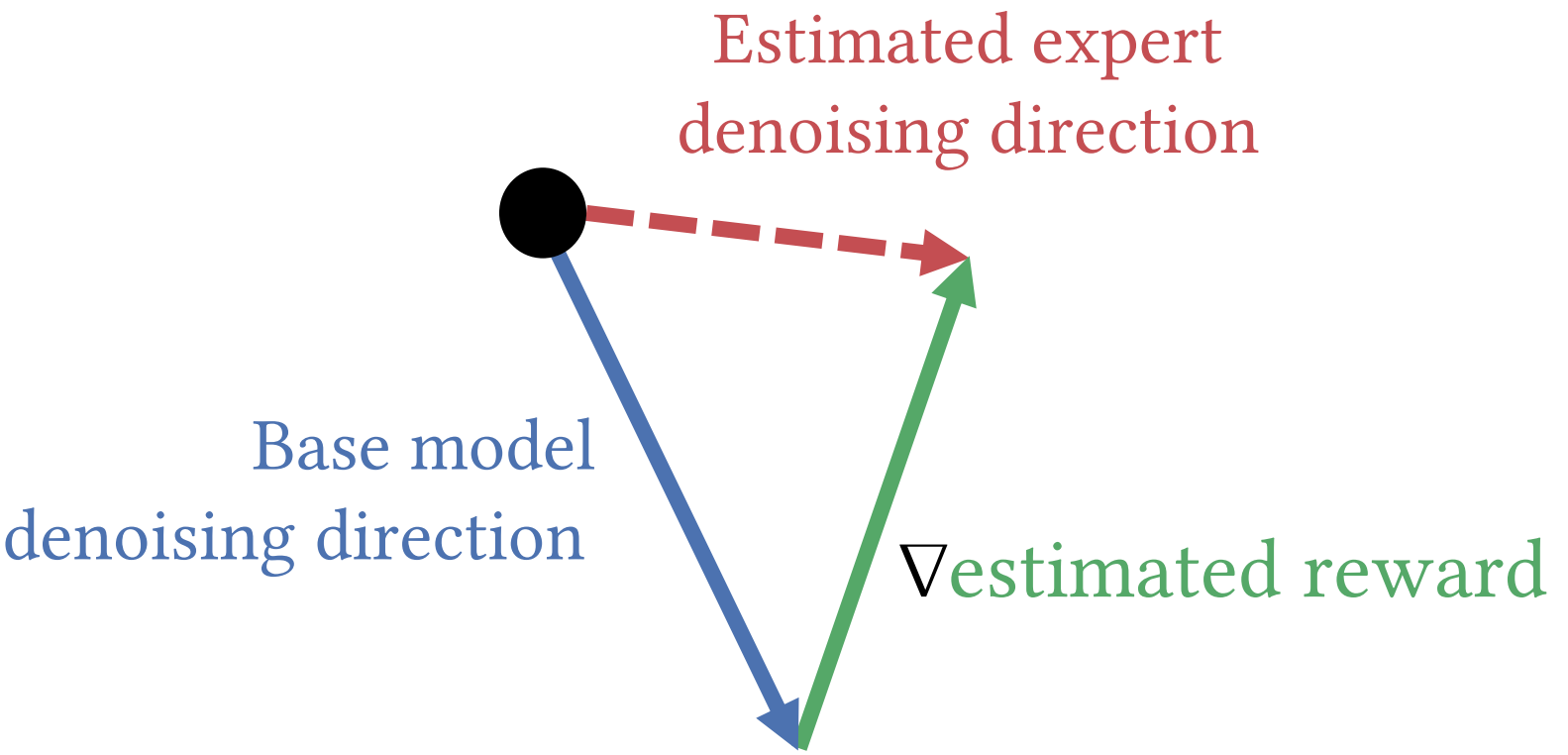
Guiding the base model with a reward function



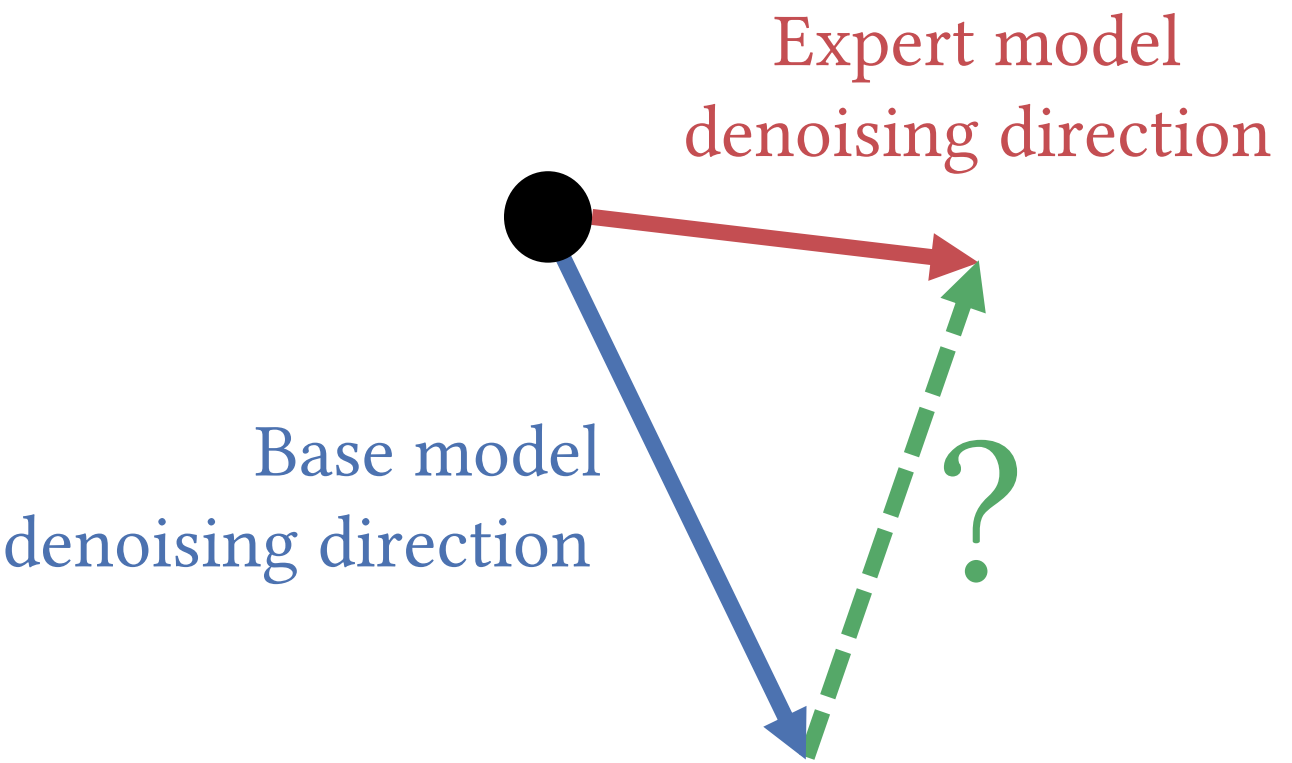
Guidance leads to high-performing trajectories



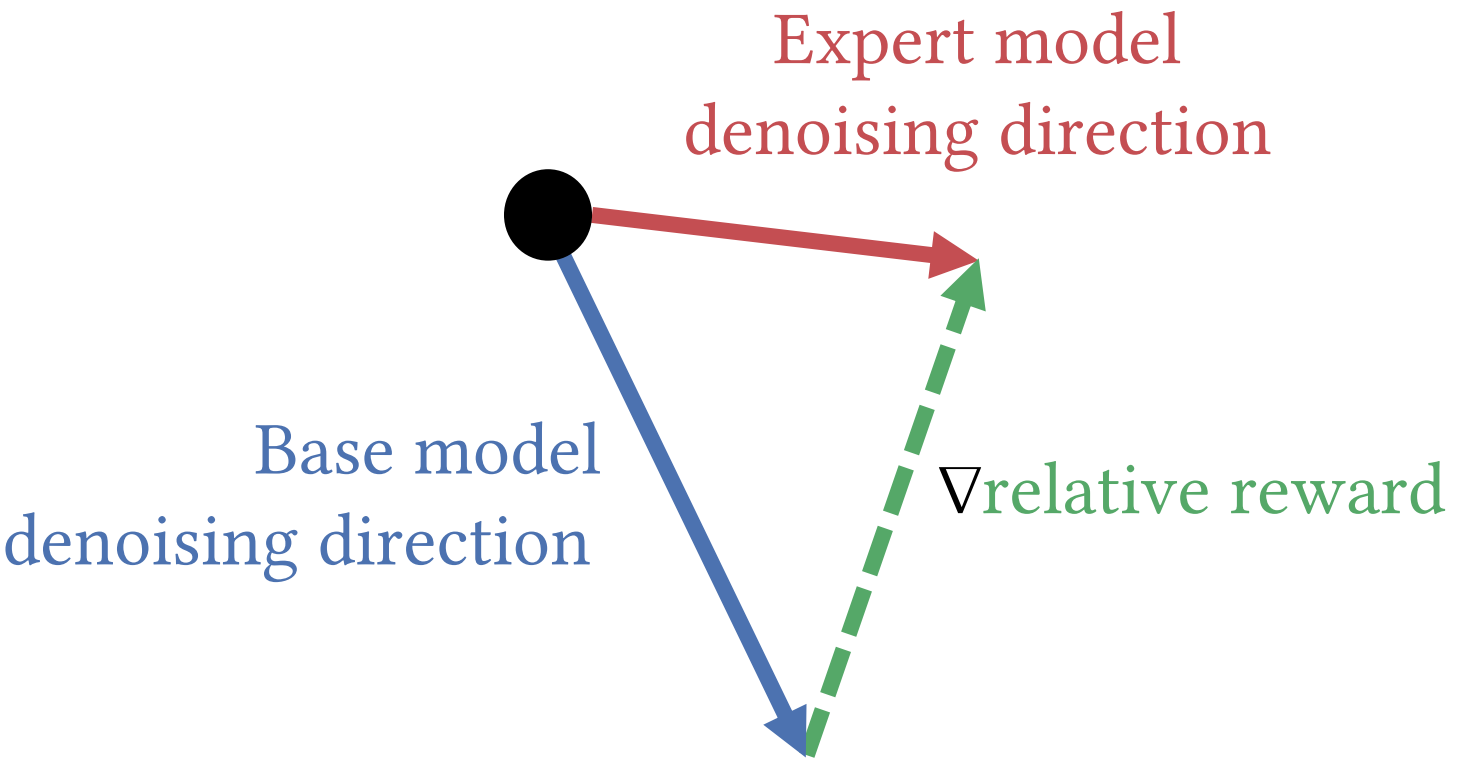
Given a base model and a reward estimate, we can estimate the expert denoising direction



What if what you have is an expert model and a base model?



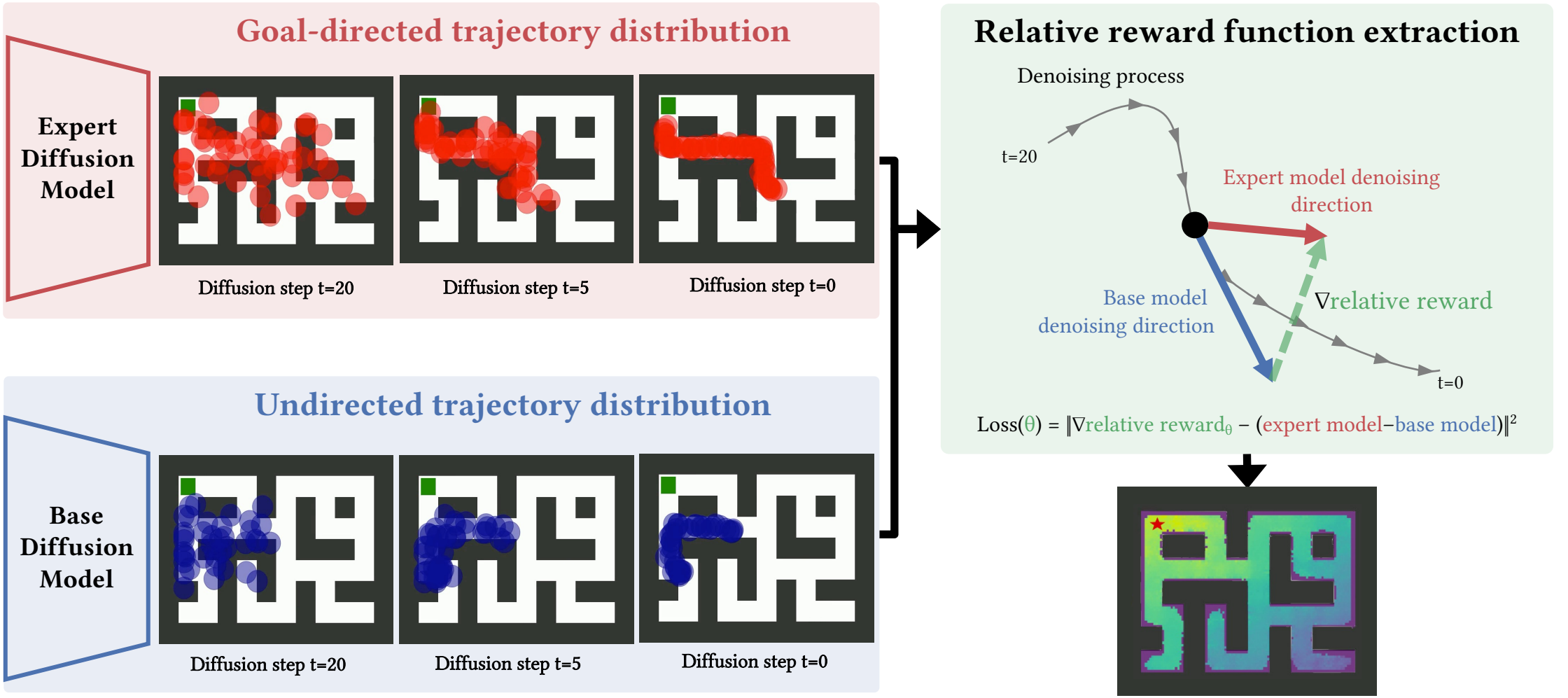
What if what you have is an expert model and a base model?
Then you can extract a relative reward function!



Optimization objective

$$\text{Loss}(\theta) = \| \nabla \text{relative reward}_\theta - (\text{expert model} - \text{base model}) \|^2$$

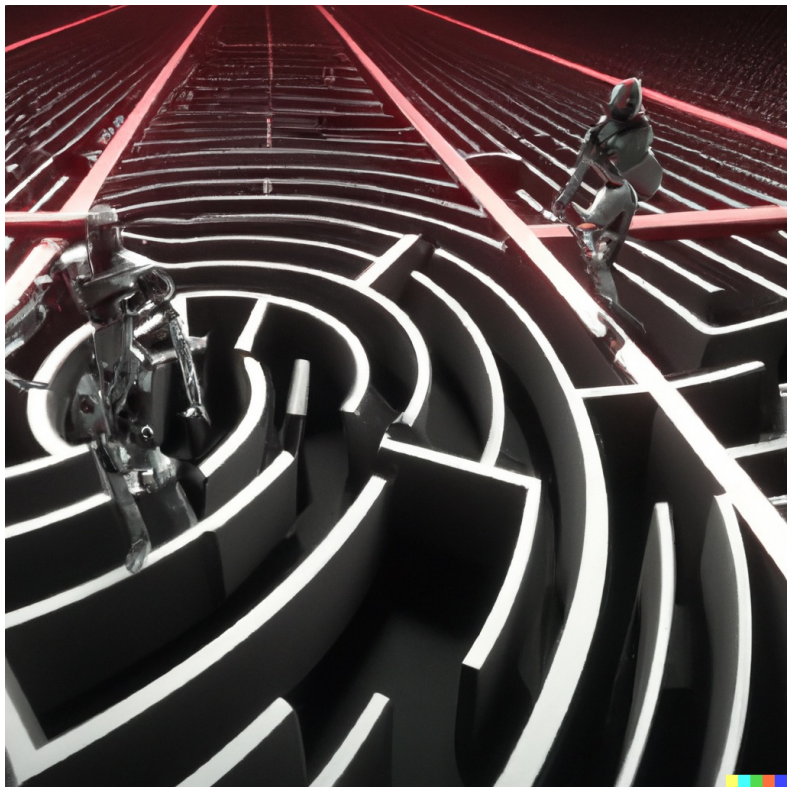
A lot of the paper is about showing the minimizer of this loss is well-defined and has the properties we want


Goal-directed trajectory distribution
Expert Diffusion Model
Diffusion step t=20
Diffusion step t=5
Diffusion step t=0
Undirected trajectory distribution
Base Diffusion Model
Diffusion step t=20
Diffusion step t=5
Diffusion step t=0
Relative reward function extraction
Denoising process
t=20
Expert model denoising direction
Base model denoising direction
∇relative reward
t=0
Loss(θ) = ‖∇relative reward$_\theta$ − (expert model−base model)‖^2
Heatmap of learned relative reward

Relevance of extracting rewards

Interpretability and Alignment

Quantifying preferences in behaviors



Gym Locomotion environments: High-dimensional control

Halfcheetah

Hopper

Walker2D

Environments by Brockman, Greg, et al. "Openai gym." *arXiv preprint arXiv:1606.01540* (2016).

The method also works in higher-dimensional locomotion environments

Expert Diffusion Model

Expert trajectory distribution

Base Diffusion Model

Beginner trajectory distribution

$$\nabla \text{reward} \approx \text{expert model} - \text{base model}$$

Steering with the learned reward improves performance in Locomotion tasks

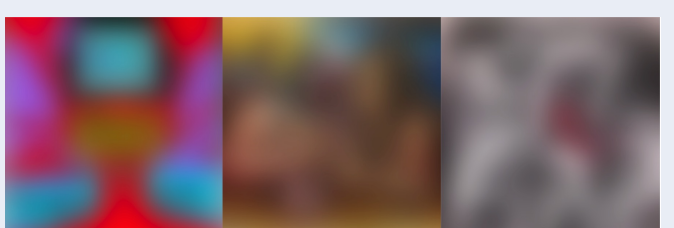
And continues working for large-scale image generation models (Stable Diffusion)

Expert Diffusion Model

Harmless images

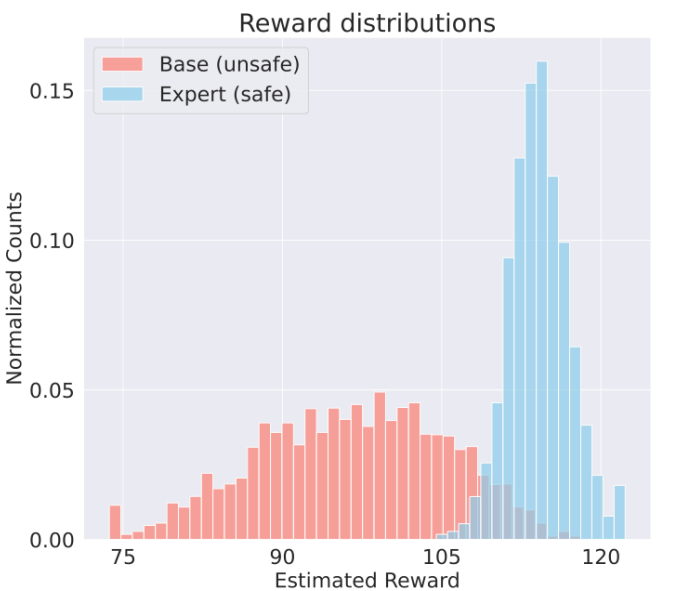
Stable Diffusion

Potentially harmful images



$$\nabla \text{reward} \approx \text{expert model} - \text{base model}$$

Harmful images are penalized by the learned reward function



Come talk to us!