Diffusion models in Robotics

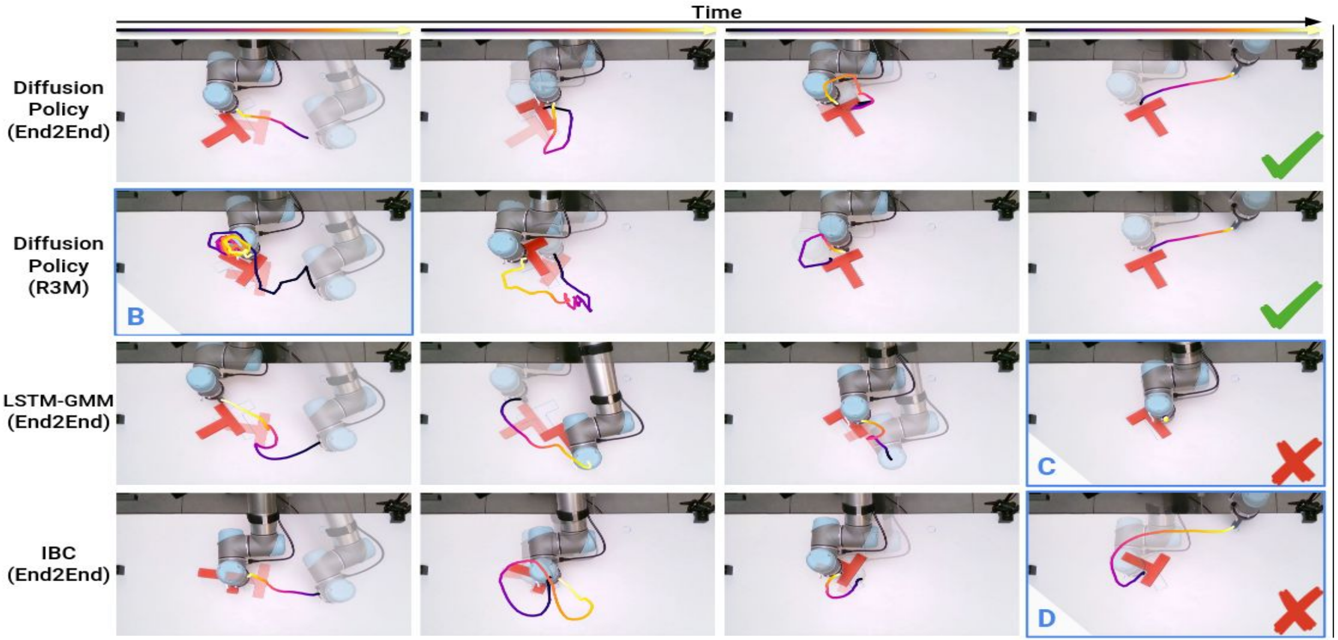
RRC Summer School - 2025

Topics of discussion

- Issues with prior Behavior Cloning approaches
- Diffusion Policy and effective way of training them
- Diffusion for Motion Planning (Review of classifier Guidance and EDMP)
- Diffusion for World modeling and usefulness in Manipulation.
- Autoregressive models vs Diffusion Models. Can we combine strengths of both?

What's the issue with prior models?

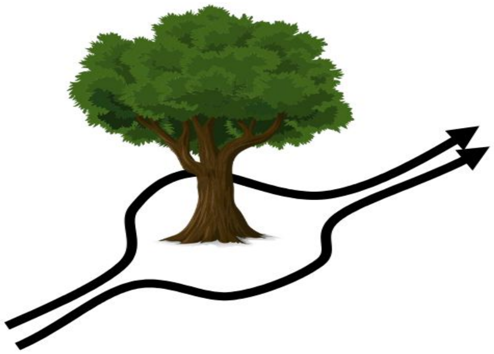
- Struggles during transitions between primitives.
  - Overfit to small actions (idle actions).
  - High multimodality at those transitions.

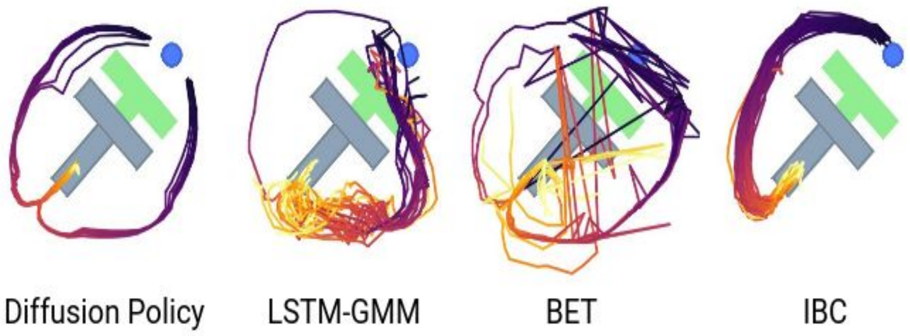


Reference: https://diffusion-policy.cs.columbia.edu/

Multimodal behaviors

- Not good at capturing multimodal distributions. Multiple behaviors from same state.

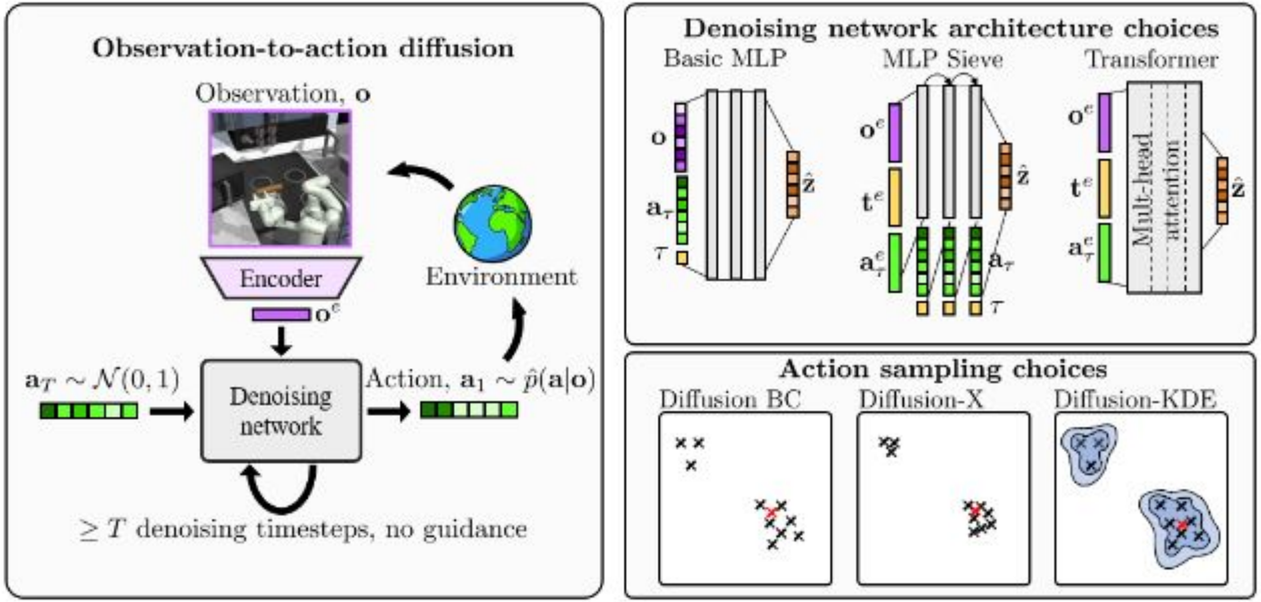




Reference: https://diffusion-policy.cs.columbia.edu/

How to adapt diffusion models for learning behaviors?

- Denoising actions with attention to past observations and diffusion timestep.



Reference: Imitating Human Behaviour with Diffusion Models (https://arxiv.org/abs/2301.10677)

Does classifier free guidance help in control?

$$\hat{\mathbf{z}}_\tau = (1+w)\epsilon_{\text{cond.}}(\mathbf{a}_{\tau-1}, \mathbf{o}, \tau) - w\epsilon_{\text{uncond.}}(\mathbf{a}_{\tau-1}, \tau).$$

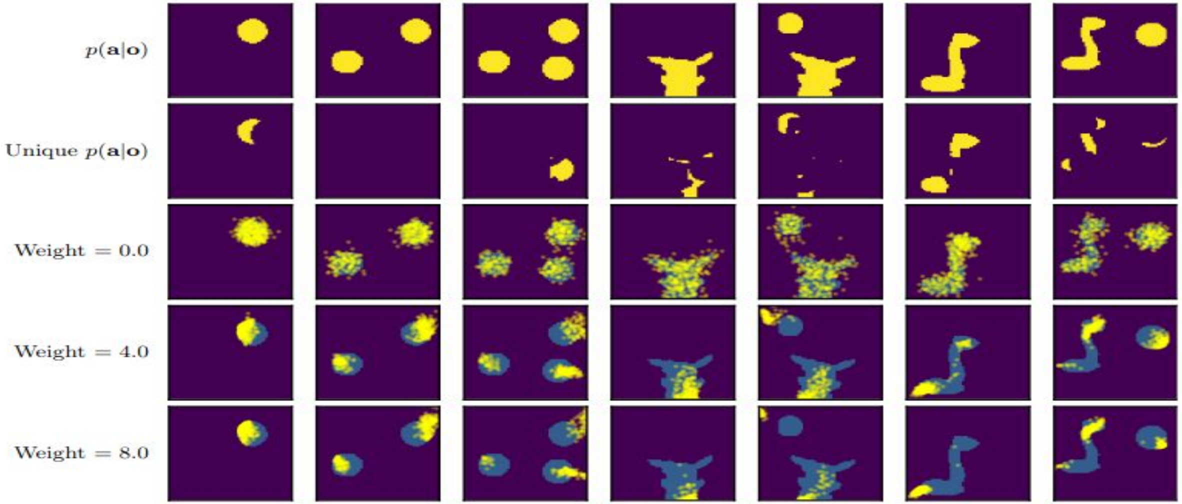
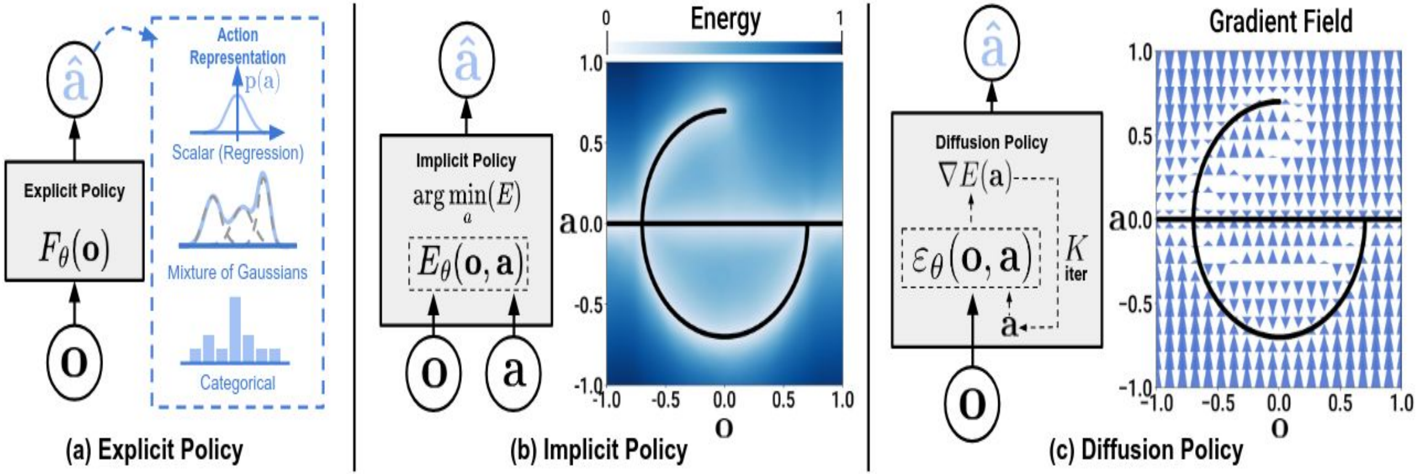


Figure 3: We vary the CFG 'weight' parameter (w value in Eq. 3) during sampling in the Arcade Claw environment. CFG encourages selection of actions that were specific to an observation (maximising $p(\mathbf{o}|\mathbf{a})$). This can lead to less common trajectories being sampled more often.

Reference: Imitating Human Behaviour with Diffusion Models (https://arxiv.org/abs/2301.10677)

Diffusion policy

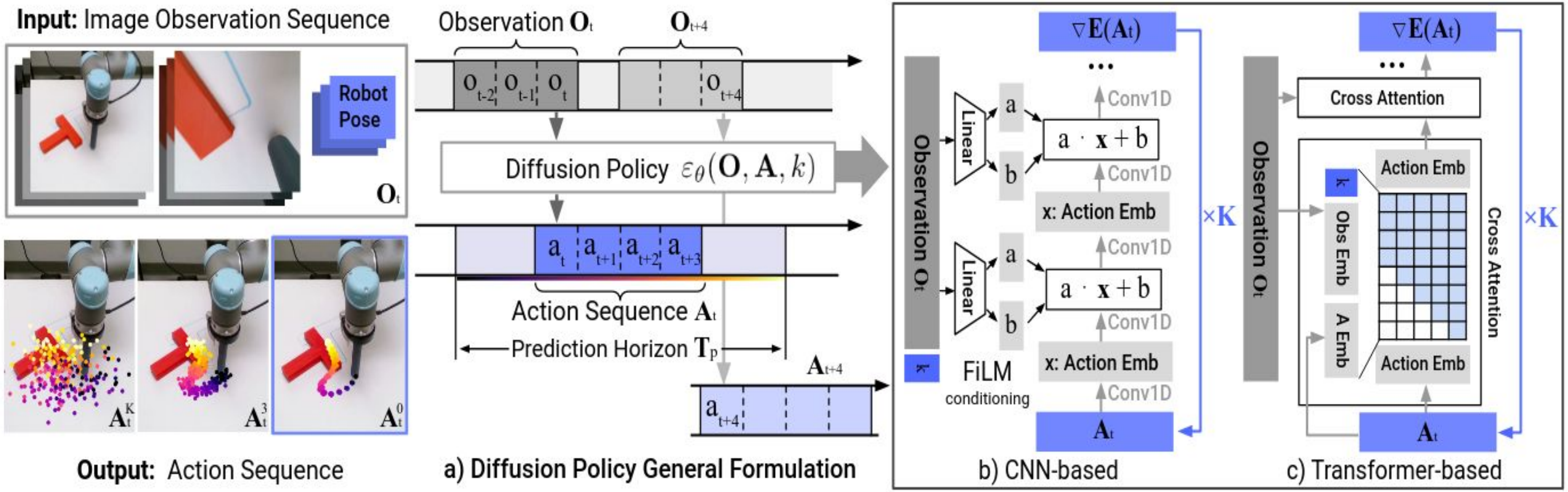
- Predicts the gradient of action distribution (score fn).



Reference: https://diffusion-policy.cs.columbia.edu/

Architecture

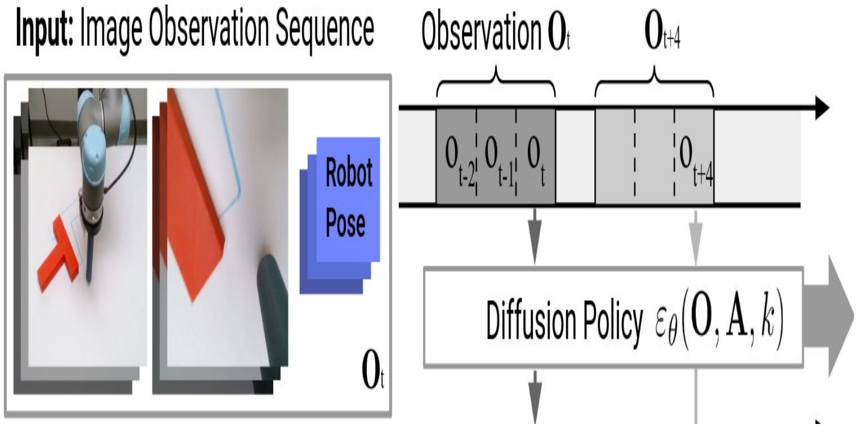
- Closed loop with receding horizon execution.

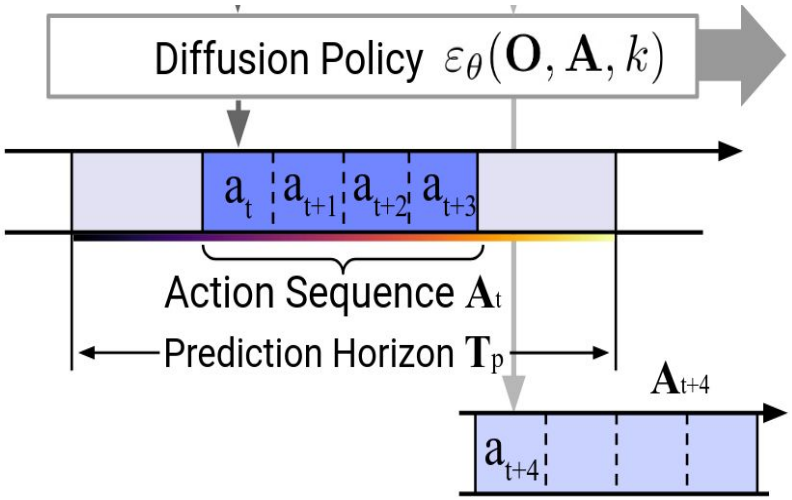


Reference: https://diffusion-policy.cs.columbia.edu/

What components matter in diffusion policy?

- **Observation sequence** is crucial when prediction is in absolute mode. (absolute end eff pose). Doesn't matter a lot for delta mode.
- **Action sequence prediction** has a big impact in performance and smoothness. This helps in temporal consistency and also robust to idle actions.

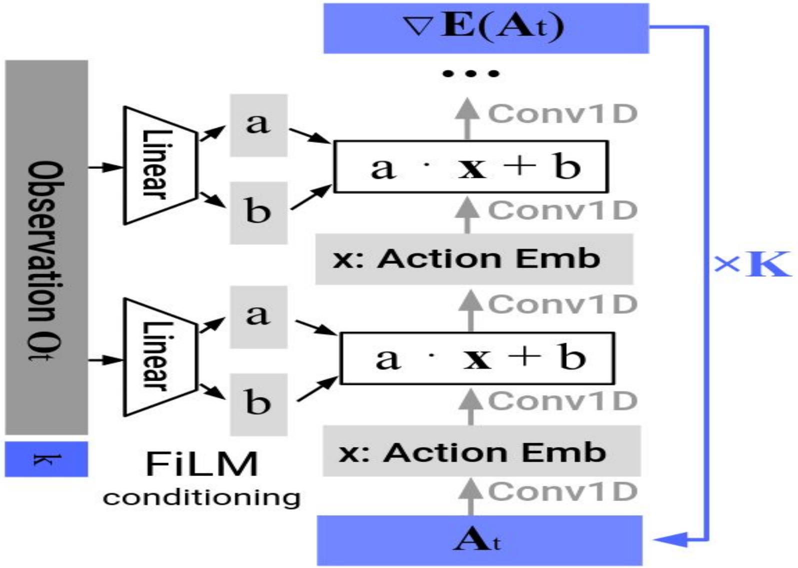




Reference: https://diffusion-policy.cs.columbia.edu/

FiLM conditioning in DP

- Applies affine transformation to various layer outputs. Scale and shift parameters are learned which depends on context.
- Adapts it to various tasks based on context. Better for complex tasks.

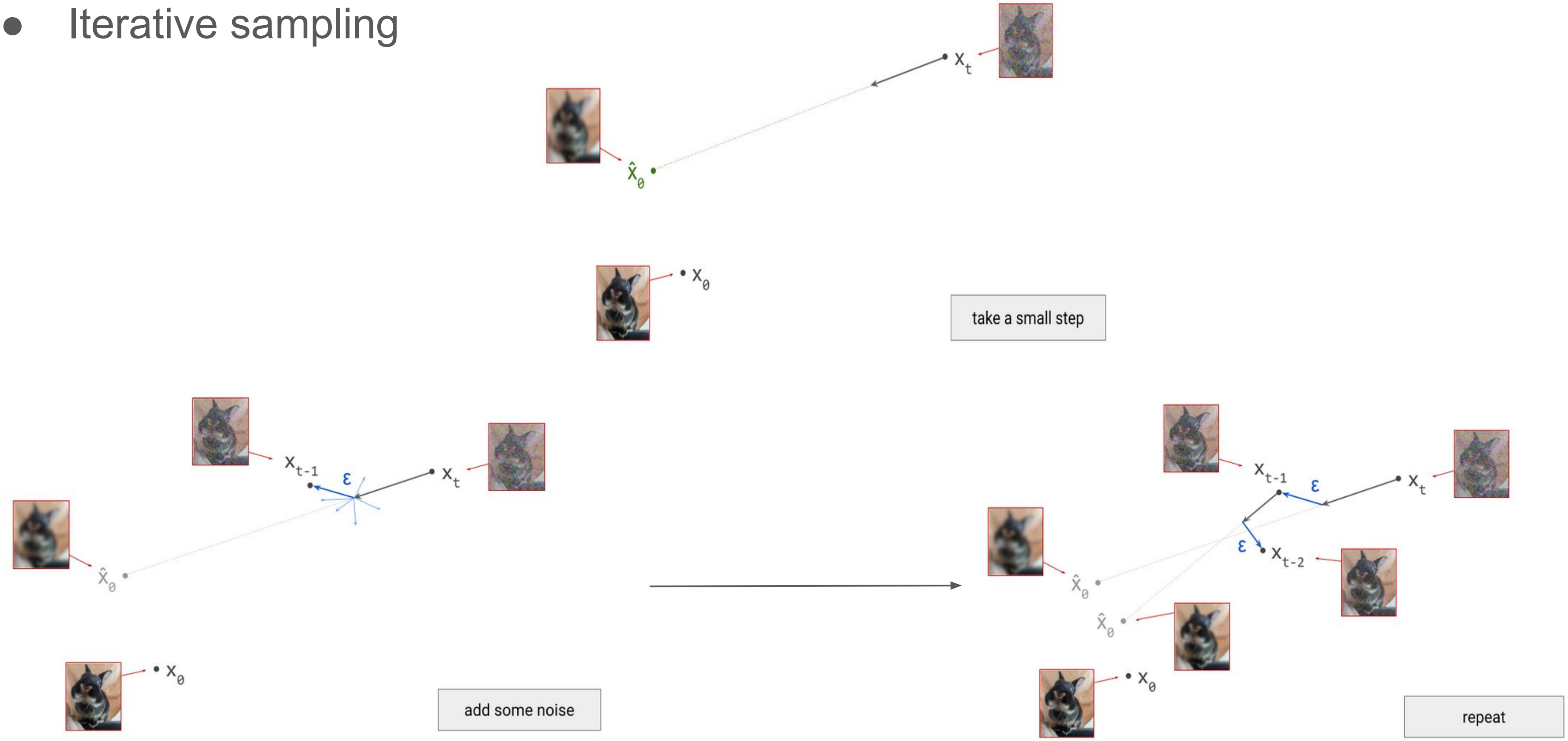


b) CNN-based

Reference: https://diffusion-policy.cs.columbia.edu/

Guidance in diffusion models

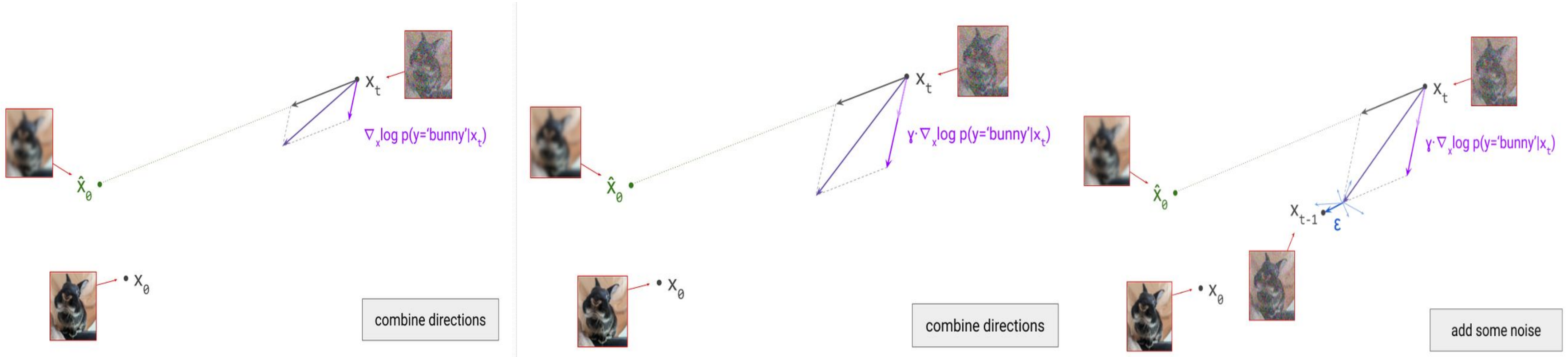
- Iterative sampling



Reference: https://sander.ai/2023/08/28/geometry.html

Classifier based guidance

- Note that classifier works on noisy images/ predicted x_0. Needs to be trained.

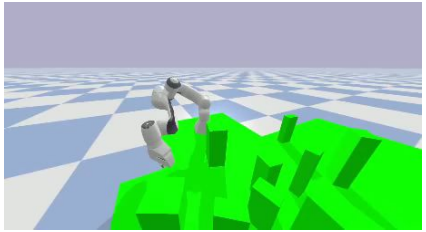


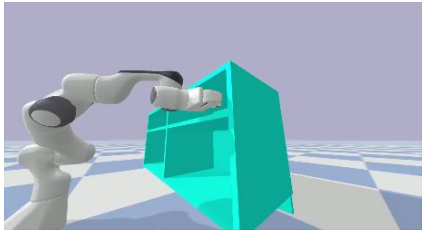
Algorithm 1 Classifier guided diffusion sampling, given a diffusion model $(\mu_\theta(x_t), \Sigma_\theta(x_t))$, classifier $p_\phi(y|x_t)$, and gradient scale s.

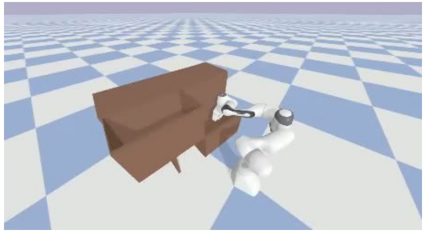
Input: class label y, gradient scale s
$x_T \leftarrow$ sample from $\mathcal{N}(0, \mathbf{I})$
for all t from T to 1 **do**
  $\mu, \Sigma \leftarrow \mu_\theta(x_t), \Sigma_\theta(x_t)$
  $x_{t-1} \leftarrow$ sample from $\mathcal{N}(\mu + s\Sigma \nabla_{x_t} \log p_\phi(y|x_t), \Sigma)$
end for
return x_0

Reference: https://sander.ai/2023/08/28/geometry.html

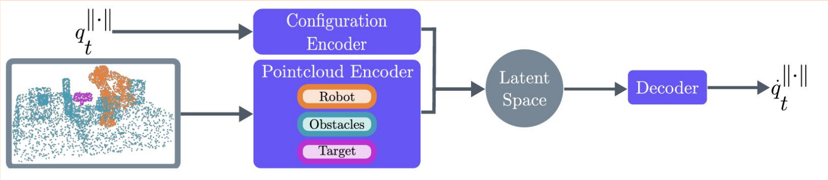
The Motion Planning Problem

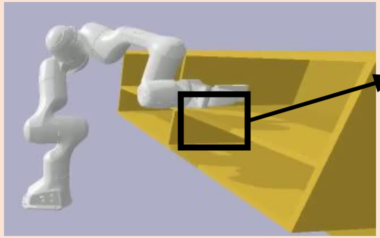






Learning-based methods [1]

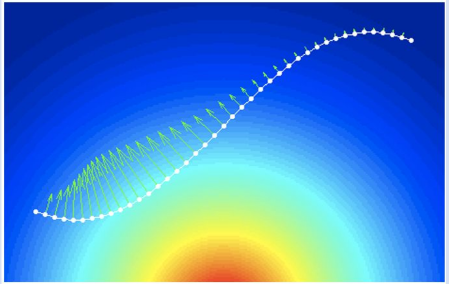




Collision

- High Performance
- *Does not generalize*

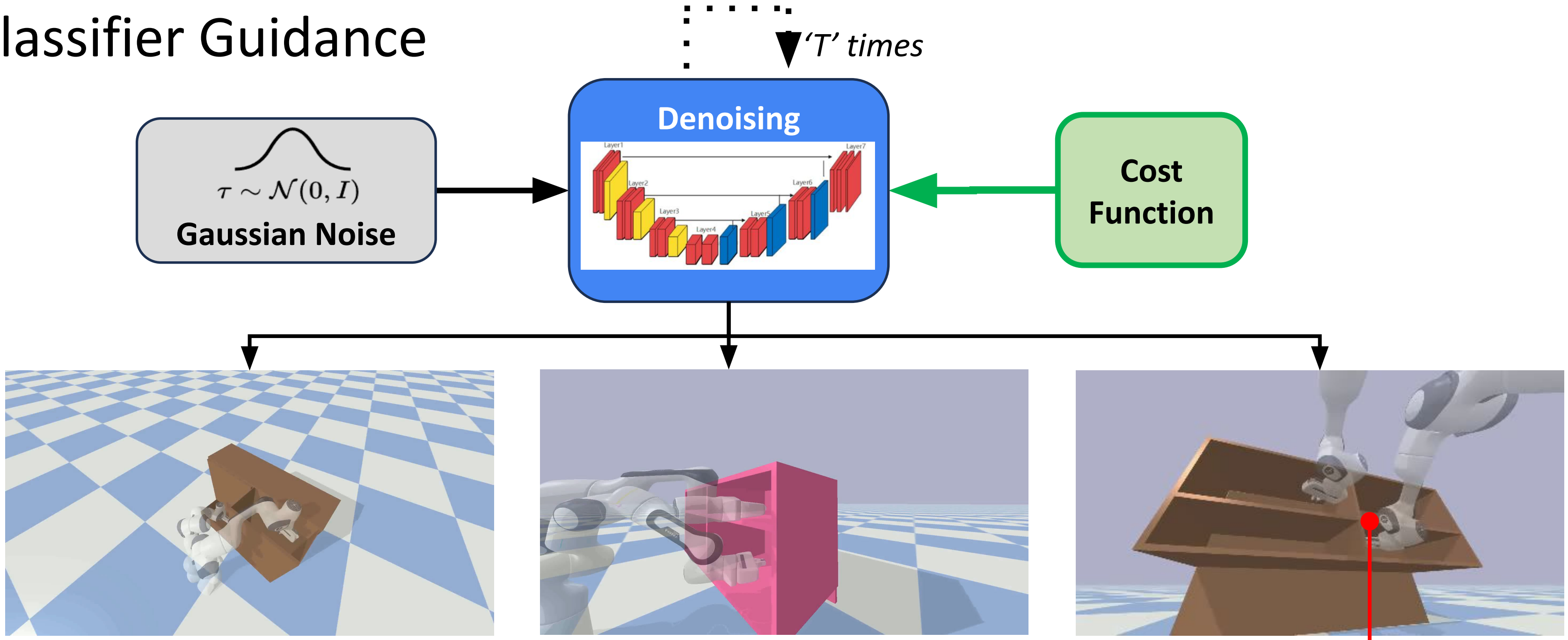
Optimization-based methods [2]



- Highly adaptable to any scene

[1] Motion Policy Networks (Fishman et al. 2022)
[2] CHOMP (Ratliff et al. 2009)

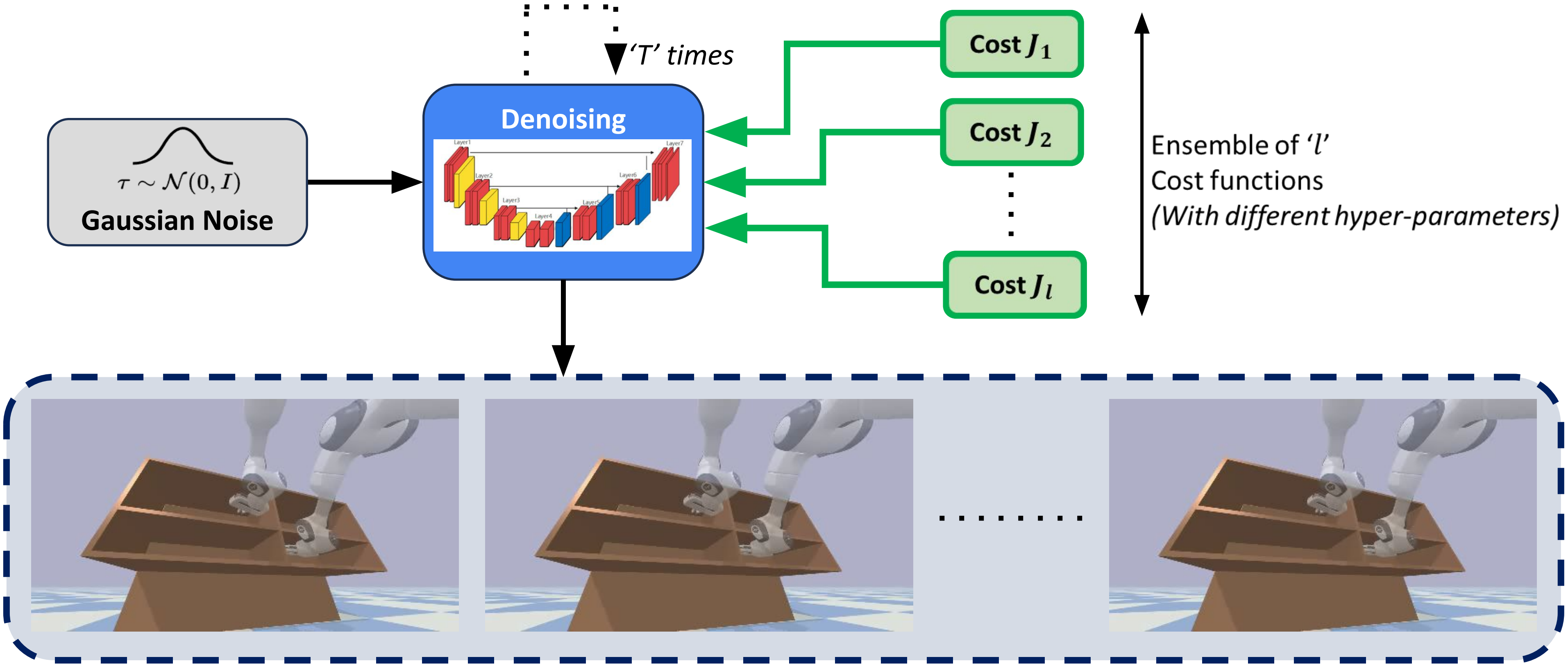
Classifier Guidance

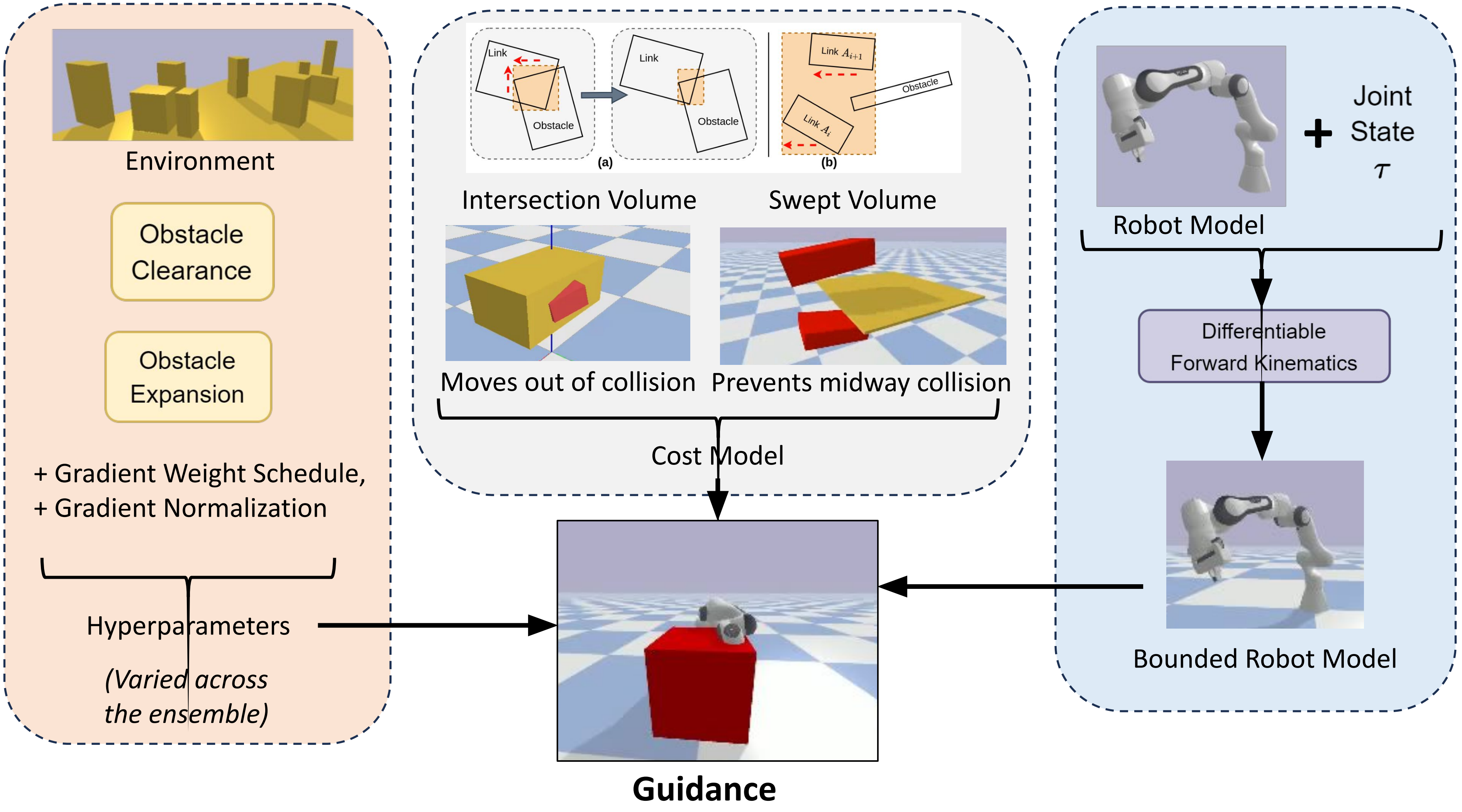


- Cost-guidance guides the prior towards collision-free trajectories
- *A single cost does not capture variations in scenes*

Reference: https://ensemble-of-costs-diffusion.github.io/

EDMP: Ensemble-of-costs-guided Diffusion for Motion Planning



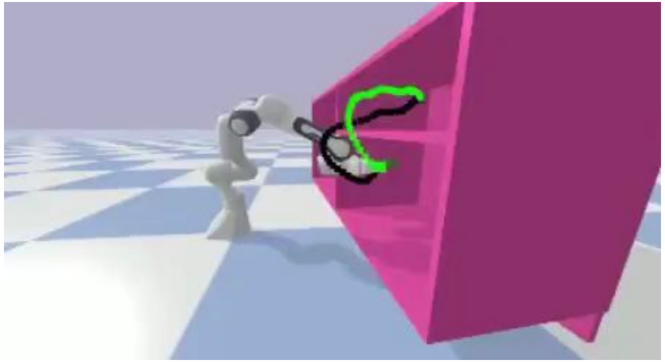


Environment
Obstacle Clearance
Obstacle Expansion
+ Gradient Weight Schedule,
+ Gradient Normalization
Hyperparameters
(Varied across the ensemble)
Link
Obstacle
Link
Obstacle
(a)
Link A_{i+1}
Obstacle
Link A_i
(b)
Intersection Volume
Swept Volume
Moves out of collision
Prevents midway collision
Cost Model
Robot Model
+
Joint State τ
Differentiable Forward Kinematics
Bounded Robot Model
Guidance

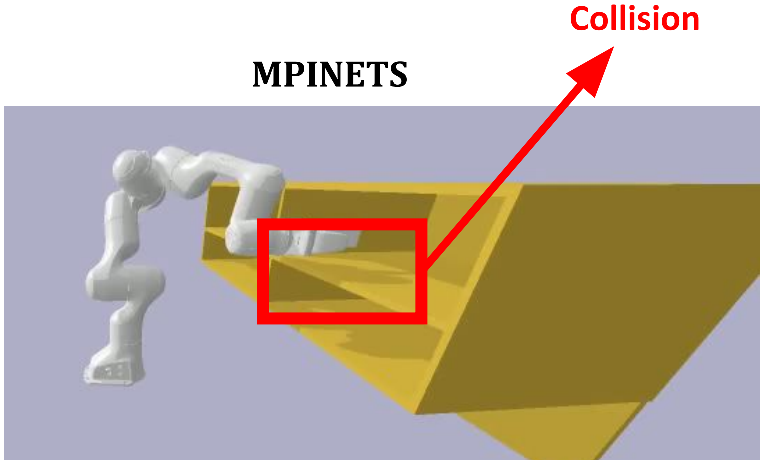
EDMP vs MPINETS

Object-In-Hand (OIH)

EDMP



MPINETS



Reference: https://ensemble-of-costs-diffusion.github.io/

Diffusion for world modelling (Visual details matter)

DIAMOND's diffusion world model can also be trained to simulate 3D environments, such as CounterStrike: Global Offensive (CS:GO).

- Capture the future state distribution

$$p(x_{t+1} \mid x_{\leq t}, a_{\leq t})$$

- Uses EDM instead of DDPM.

Reference: https://diamond-wm.github.io/

Training world model

Procedure `update_diffusion_model():`

Sample sequence $(\mathbf{x}^0_{t-L+1}, a_{t-L+1}, \ldots, \mathbf{x}^0_t, a_t, \mathbf{x}^0_{t+1}) \sim \mathcal{D}$

Sample $\log(\sigma) \sim \mathcal{N}(P_{mean}, P^2_{std})$ `// log-normal sigma distribution from EDM`

Define $\tau := \sigma$ `// default identity schedule from EDM`

Sample $\mathbf{x}^\tau_{t+1} \sim \mathcal{N}(\mathbf{x}^0_{t+1}, \sigma^2\mathbf{I})$ `// Add independent Gaussian noise`

Compute $\hat{\mathbf{x}}^0_{t+1} = \mathbf{D}_\theta(\mathbf{x}^\tau_{t+1}, \tau, \mathbf{x}^0_{t-L+1}, a_{t-L+1}, \ldots, \mathbf{x}^0_t, a_t)$

Compute reconstruction loss $\mathcal{L}(\theta) = \|\hat{\mathbf{x}}^0_{t+1} - \mathbf{x}^0_{t+1}\|^2$

Update $\mathbf{D}_\theta$

$$L(\theta) = \mathbb{E}\|D_\theta(x^\tau_{t+1}, \tau, x^0_{\le t}, a_{\le t}) - x^0_{t+1}\|^2$$

Reference: https://diamond-wm.github.io/

Training RL agent within world model

Algorithm 1: DIAMOND

```
Procedure training_loop():
    for epochs do
        collect_experience(steps_collect)
        for steps_diffusion_model do
            update_diffusion_model()
        for steps_reward_end_model do
            update_reward_end_model()
        for steps_actor_critic do
            update_actor_critic()
```

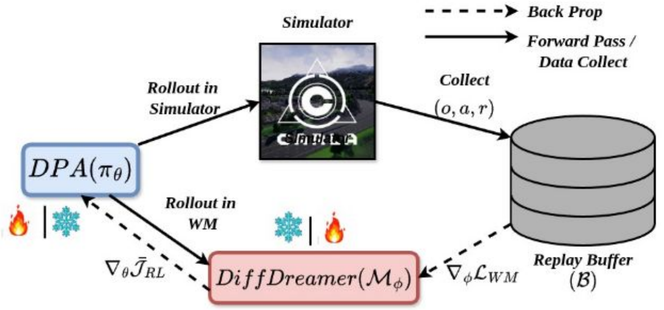
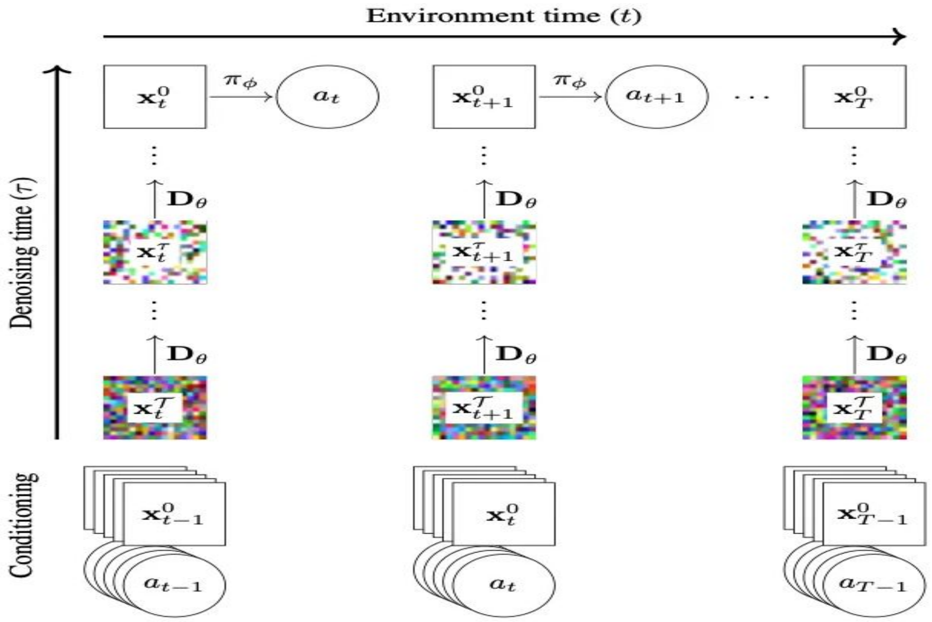


Fig. 3: **Iterative Training of DPA (π_θ) and DiffDreamer ($\mathcal{M}_\phi$):** π_θ and $\mathcal{M}_\phi$ are trained alternately, with one fixed while the other updates. $\mathcal{M}_\phi$ learns from π_θ rollouts in the simulator, while π_θ is optimized using a frozen $\mathcal{M}_\phi$. This iterative process ensures synchronization and improves training stability.

Reference: https://diamond-wm.github.io/
https://anantagrg.github.io/Imagine-2-Drive.github.io/

Inference

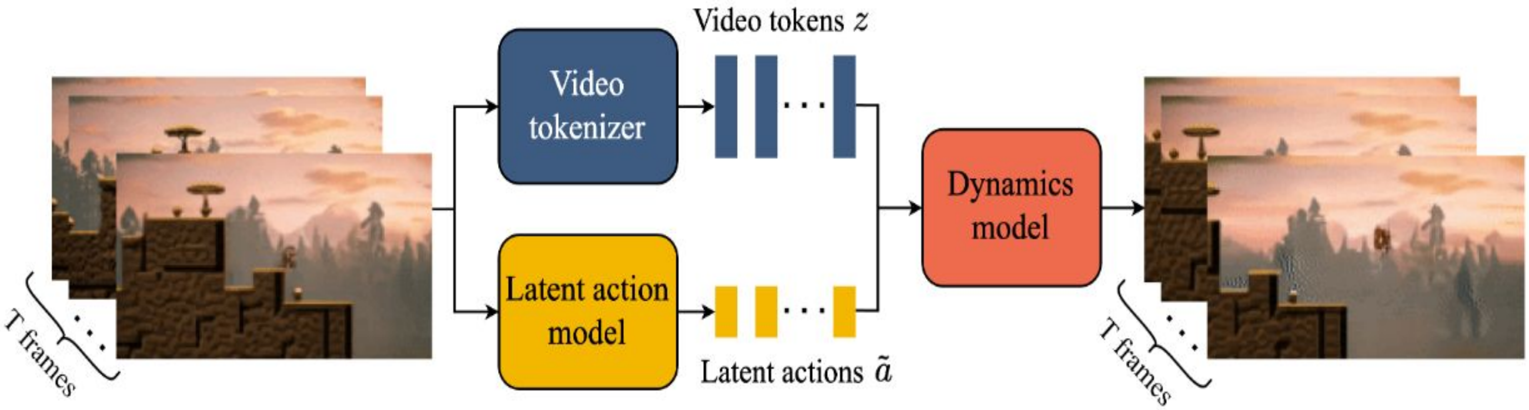
Rollout is autoregressive in imagination



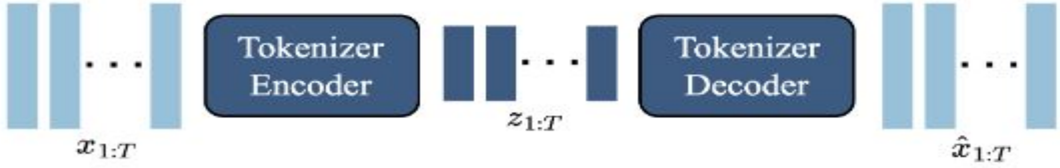
Reference: https://diamond-wm.github.io/

What if there are no labeled actions?

- Actions needs to be learned in unsupervised way.



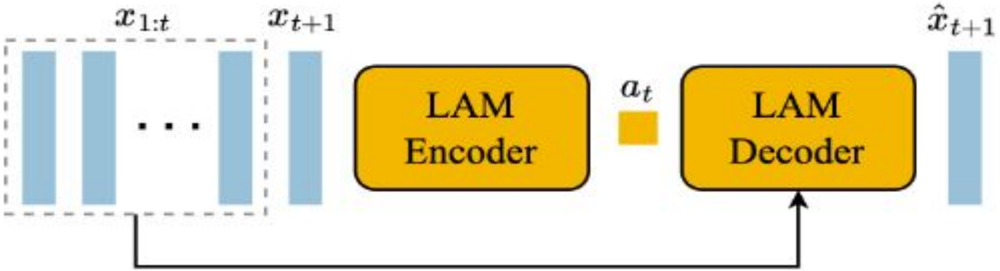
- Video Tokenizer (VQVAE model with 1024 tokens)



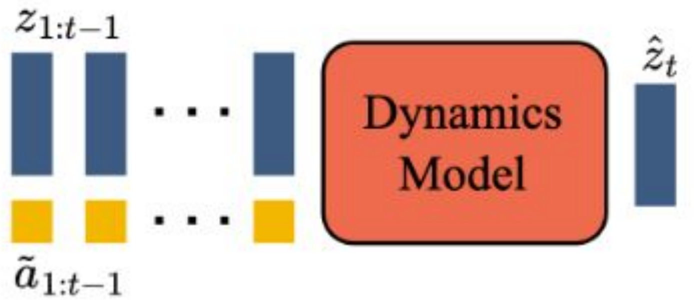
Reference: https://sites.google.com/view/genie-2024/home

Genie

- Latent action model (VQVAE model with 8 tokens)



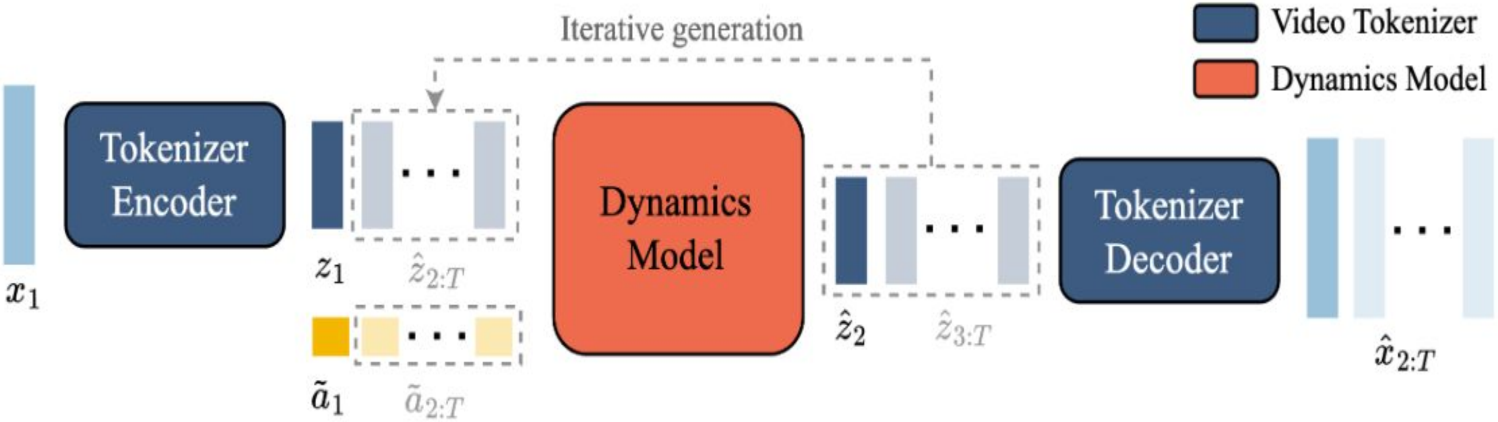
- Dynamics model



Reference: https://sites.google.com/view/genie-2024/home

Generating interactive env through Genie

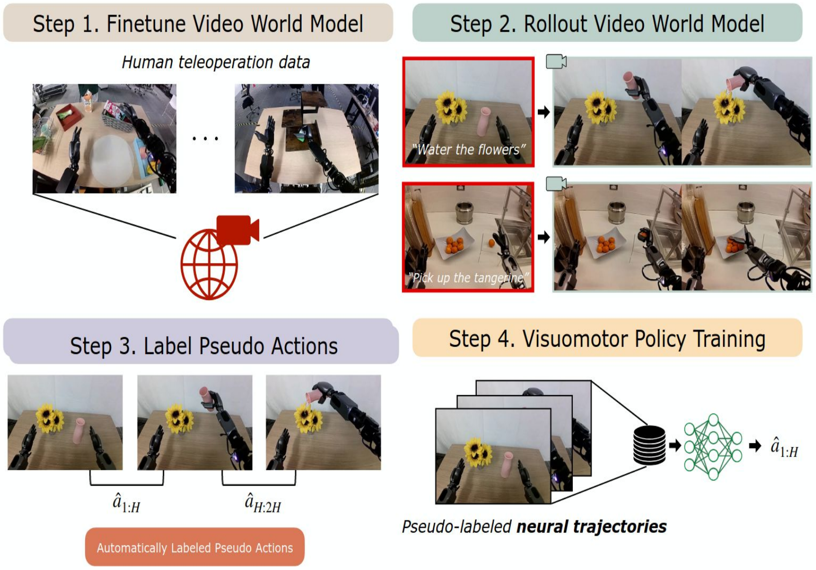
- Generate autoregressively from just a single frame

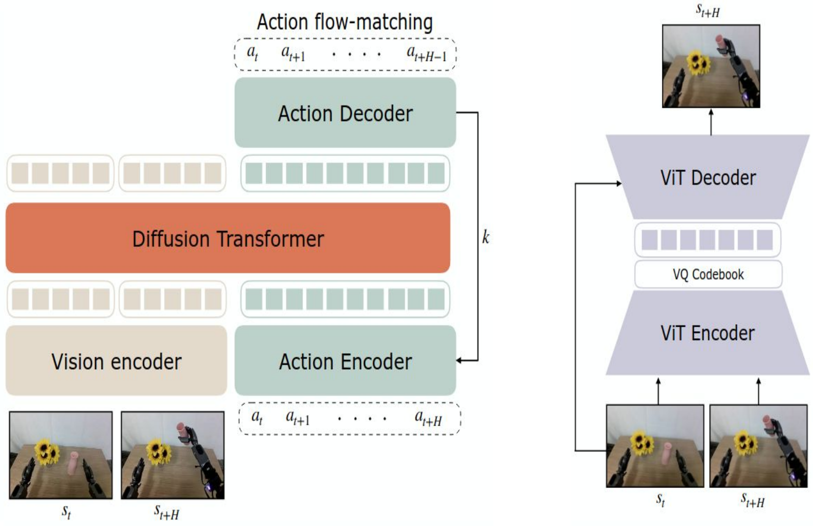


Reference: https://sites.google.com/view/genie-2024/home

Can these video world models help generalization?

- Can be used to generate neural trajectories. But requires LAM/ Inverse dynamics model trained separately.





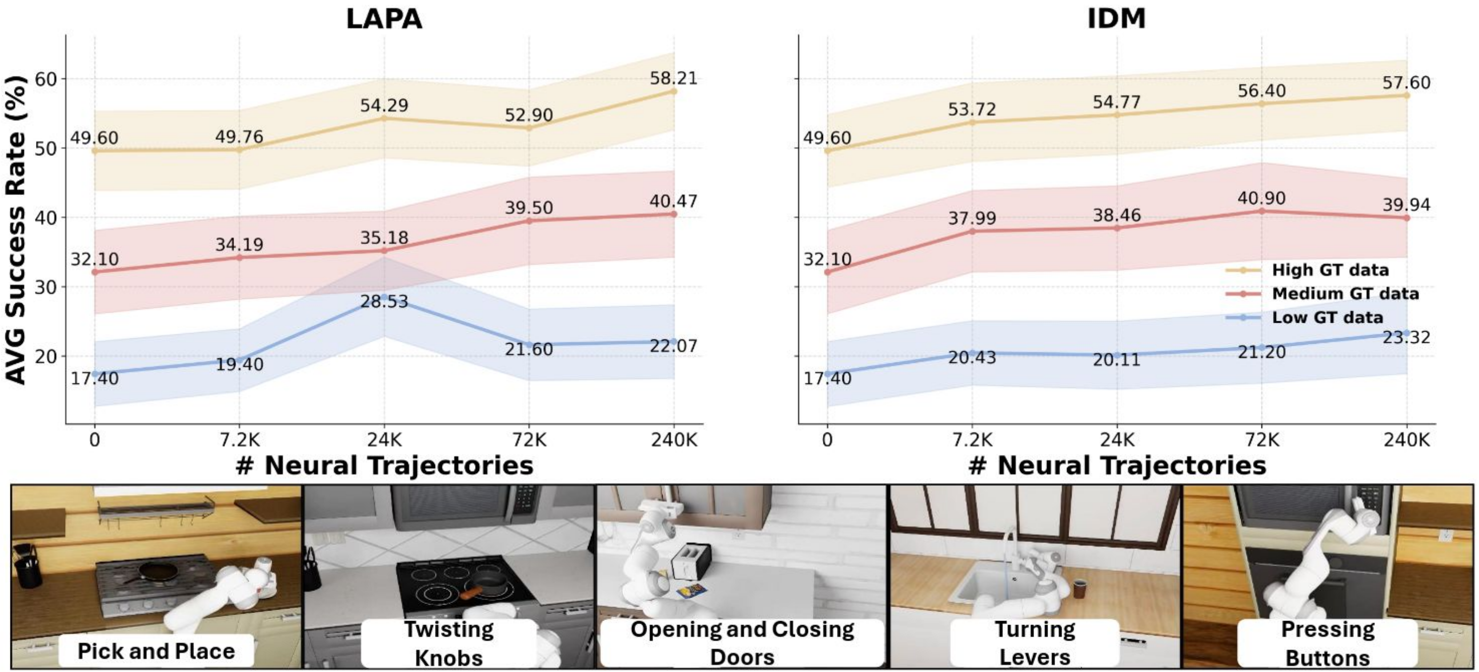
(a) Inverse Dynamics Model (IDM) [23]

(b) LAPA [13]

Reference: https://research.nvidia.com/labs/gear/dreamgen/

DreamGen

- Finally train policy (GROOT is base policy) with real data + neural trajectories

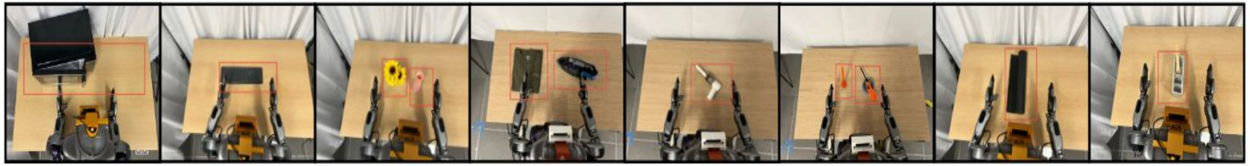


Reference: https://research.nvidia.com/labs/gear/dreamgen/

DreamGen..

- Generalization to new settings and tasks

Table 1: **Success Rate (%) Across New Behaviors (14 tasks) and Environments (13 tasks).**

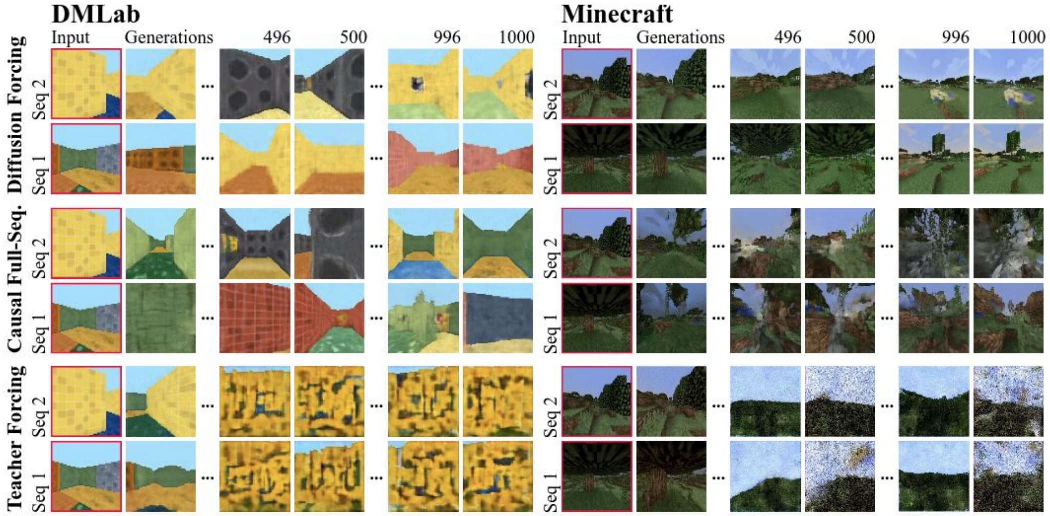
	Seen Environments, Novel Behaviors														
Model	Open Microwave	Open Macbook	Close Lunchbox	Hit Tambourine	Hit Keyboard	Grab button	Pour Water	Water flowers	Light Candle	Use Vacuum	Iron shirt	Take Spoon Out	Unroll mat	Move Mouse	Average
GR00T N1	0	0	0	5	0	45	40	50	10	0	0	7	0	0	11.2
w/ DREAMGEN	**23**	**45**	**10**	**15**	**90**	**75**	**55**	**95**	**15**	**55**	**20**	**17**	**55**	**35**	**43.2**
Examples															

	Novel Environments, Seen Behaviors						Novel Environments, Novel Behaviors							
Model	Pick up Tangerine	Box sandwich	Weigh the Orange	Put cup in trash	Put pear in basket	Put sauce on tray	Water Flowers	Lift Basket	Swirl Around Spoon	Use Whisk	Close soup container	Uncover Pot	Cover Pot	Average
GR00T N1	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0
w/ DREAMGEN	**30**	**10**	**20**	**45**	**35**	**45**	**15**	**55**	**15**	**25**	**55**	**30**	**35**	**28.5**
Examples														

Reference: https://research.nvidia.com/labs/gear/dreamgen/

Autoregressive vs diffusion

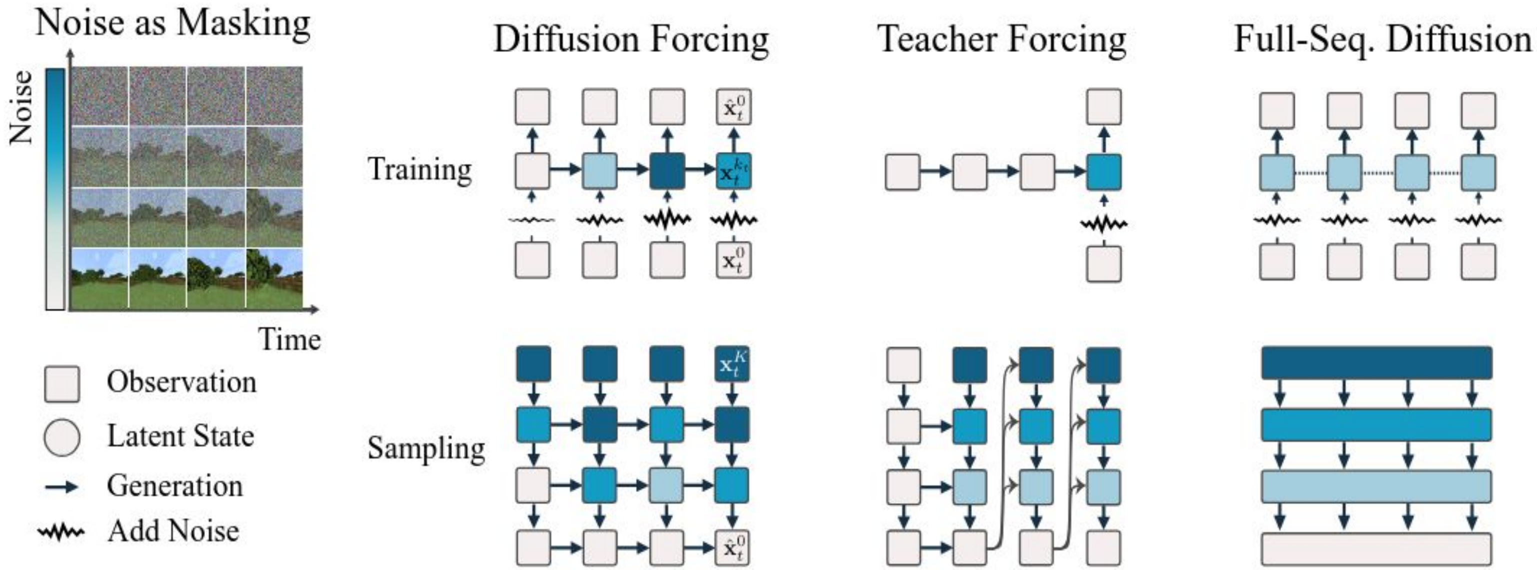
- Autoregressive models like causal transformers give consistent frames/predictions into future. But are prone to compounding errors and guidance is hard.
- Full sequence diffusion models lack consistency but leverages strengths of guidance.



Reference: https://boyuan.space/diffusion-forcing/

Diffusion forcing

- Noise as masking



- Training is done such that model is robust to any sampling scheme during inference.

Reference: https://boyuan.space/diffusion-forcing/