Offline Reinforcement Learning with Diffusion models

Team 6. 20243347 JeongWoo Park

2025.04.21.

Overview

1. Offline Reinforcement Learning review
2. Motivation and Main Problem
3. Introduction to diffusion model
    a. DDPM : Denoising Diffusion Probabilistic Modeling
4. Diffusion models for robotics
    a. Preliminary : notations & taxonomy
    b. Task planning : DALL-E-BOT
    c. Trajectory generation : Diffuser, Decision Diffuser
5. Future directions
6. Summary

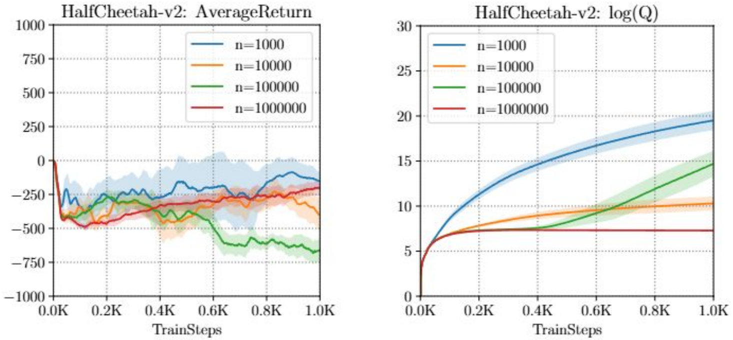
Offline Reinforcement Learning review

Offline RL algorithms :

❖ Designed to learn from static datasets without requiring online interaction. (Sample efficient)

❖ Fundamental issue: how to decide (action) values for out-of-distribution(OOD) actions?

❖ Distribution shift problem [1]: Bootstrapping error is a key source of instability



(Left) Actual return, (Right) log Q-values

Q values are much larger than actual return

❖ Previous Approaches : Pessimism (CQL), Policy constraints (BRAC), Avoid OOD actions (AWAC, IQL)

[1] Kumar, Aviral, et al. "Stabilizing off-policy q-learning via bootstrapping error reduction." NeurIPS (2019).

Motivation and Main Problem

Offline RL algorithms had advanced but challenge still remains [2]:

- ❖ Gaussian policies may fail to fit the datasets with complex distributions. (restricted expressiveness)
- ❖ Data scarcity problem in environments with high-dimensional state spaces, complex interaction.
- ❖ Per-step autoregressive planning approaches suffer from the compounding error problem.

Why don’t we utilize superior representation power of diffusion models?

Why should we use diffusion models? How about other generative models?

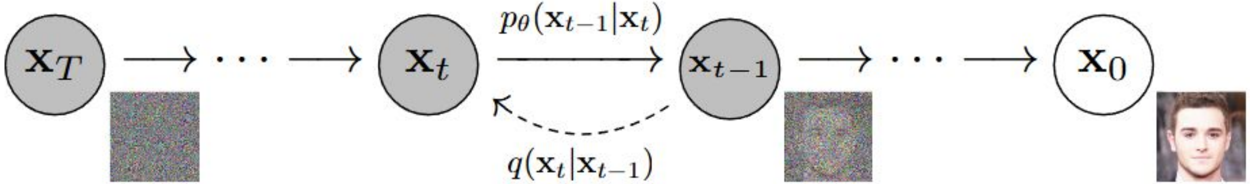
- ❖ Diffusion models achieved substantial success in diverse domains (computer vision, NLP, etc...)
- ❖ Conditional diffusion models perform better than traditional methods using GAN or Transformer.
- ❖ Achieves diverse and fine-grained motion generation with various conditioning contexts.

[2] Zhu, Zhengbang, et al. "Diffusion models for reinforcement learning: A survey." arXiv preprint arXiv:2311.01223 (2023).

Introduction to diffusion model (Probabilistic Generative models)

Generative modeling : Creating data from noise

- ❖ Sequentially corrupting training data with slowly increasing noise, and then learning to reverse this corruption in order to form a generative model of the data. [3]
- ❖ DDPM(Sohl-Dickstein et al. 2015, Ho et al. 2020) : Denoising diffusion probabilistic modeling



$\mathbf{x}_T$
$\mathbf{x}_t$
$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$
$q(\mathbf{x}_t|\mathbf{x}_{t-1})$
$\mathbf{x}_{t-1}$
$\mathbf{x}_0$

- ❖ X_0 : Original data, X_T: Gaussian noise
- ❖ Forward process (q) : Markov chain that adds gaussian noise to the original data $X_0 \sim q(X_0)$
- ❖ Reverse process (p) : Markov chain that removes noise from the gaussian noise $X_T \sim N(0, I)$
- ❖ Our training objective is to model reverse process to fit forward process posteriors.

[3] Song, Yang, et al. "Score-based generative modeling through stochastic differential equations." arXiv preprint arXiv:2011.13456 (2020).

Introduction to diffusion model (DDPM)

We can have simplified, tractable training objective [4]

- ❖ Reverse process are in same functional form with the forward process, when forward process variance β_t are small. (Sohl-Dickstein et al. 2015)
- ❖ Forward process variance are set to small constants $\beta_1 = 10^{-4}$, $\beta_{T'} = 0.02$ (data scale [-1, 1])
- ❖ Forward process admits sampling at an arbitrary timestep t in closed form. (Because it is Gaussian)
- ❖ Forward process posteriors are tractable when conditioned on X_0

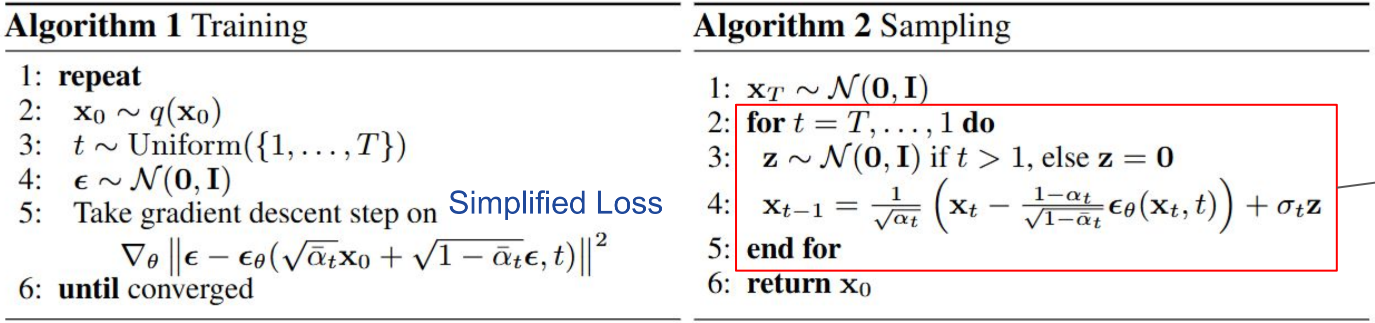
$$a_t := 1 - \beta_t,\ \bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s,\ \epsilon \sim N(0,\ I)$$

t : uniform [1,T], ϵ_θ : trained model

$$L_{\text{simple}}(\theta) := \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\right\|^2\right]$$

[4] Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." NeurIPS (2020)

Introduction to diffusion model (Network Architecture)

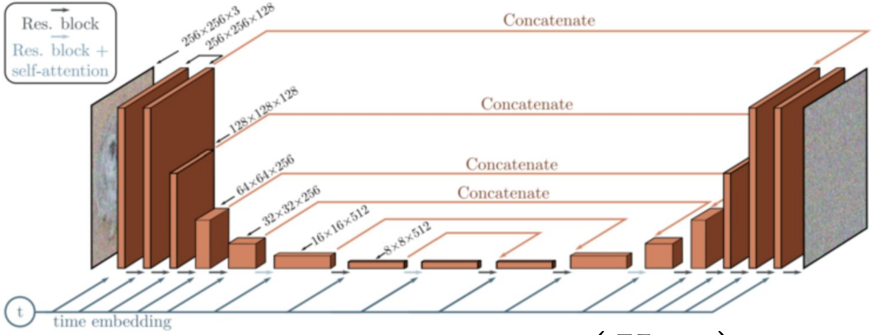


```
Algorithm 1 Training
repeat
    $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
    $t \sim \mathrm{Uniform}(\{1, \ldots, T\})$
    $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    Take gradient descent step on
        $\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$
until converged
```

Simplified Loss

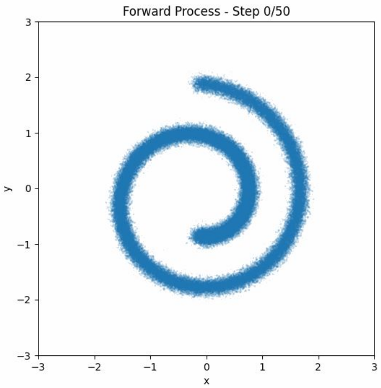
```
Algorithm 2 Sampling
$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \ldots, 1$ do
    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
    $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}$
end for
return $\mathbf{x}_0$
```

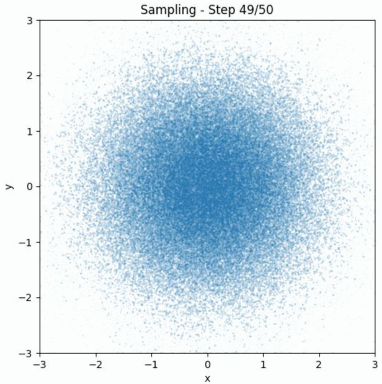
Sequential Sampling
(ex. T=1000)
⇒ Limitation : Slow



Model depends on time $\epsilon_\theta\left(X_t,\, t\right)$

Time embedding added to each layer





Diffusion Models for Robotics (Preliminary)

Taxonomy of RL algorithms (Recap)

- Model based RL : utilizes an learned dynamic model usually for planning

  (Pro) Sample efficient (Con) Model accuracy is crucial for its performance.

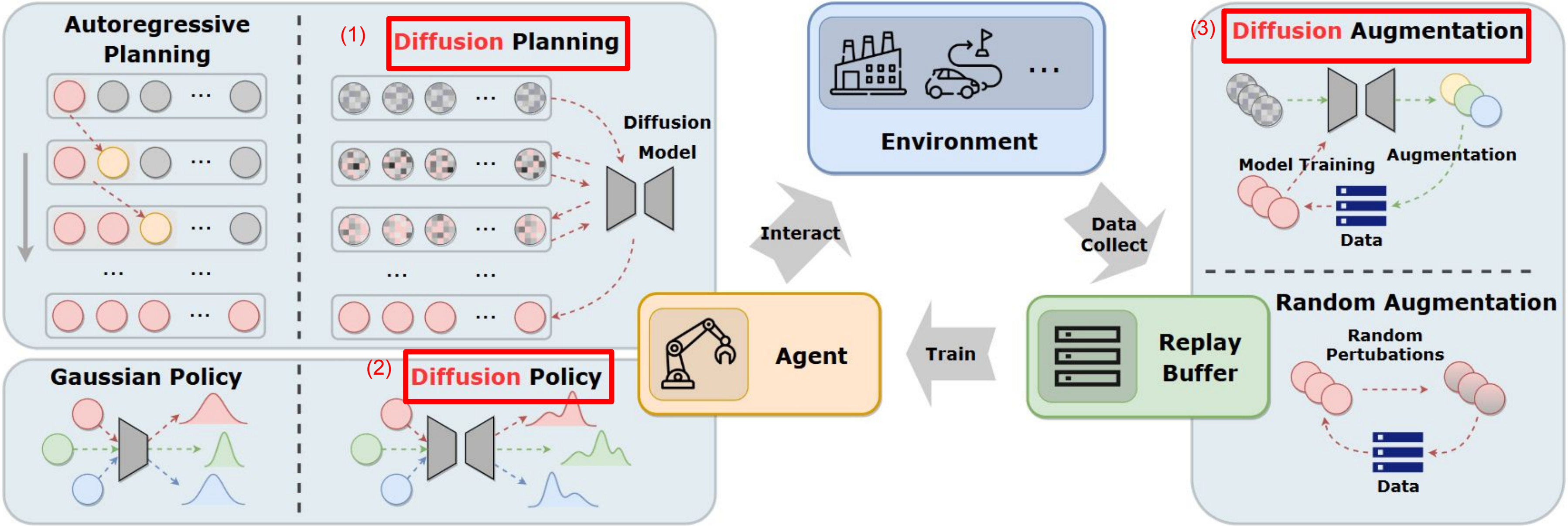
- Model-free RL : agent learns optimal policies directly from interactions with the environment

  (Pro) Simple and flexible to environment changes (Con) Poor sample efficiency

Application of diffusion models can be mainly fall into three categories

- Model based RL ⇒ Planning part of trajectories (state, action, reward)
- Model-free RL ⇒ Use diffusion models as the Policy, improving existing model-free RL solutions
- Data synthesizer (Augmentation)

Diffusion Models for Robotics (Preliminary)



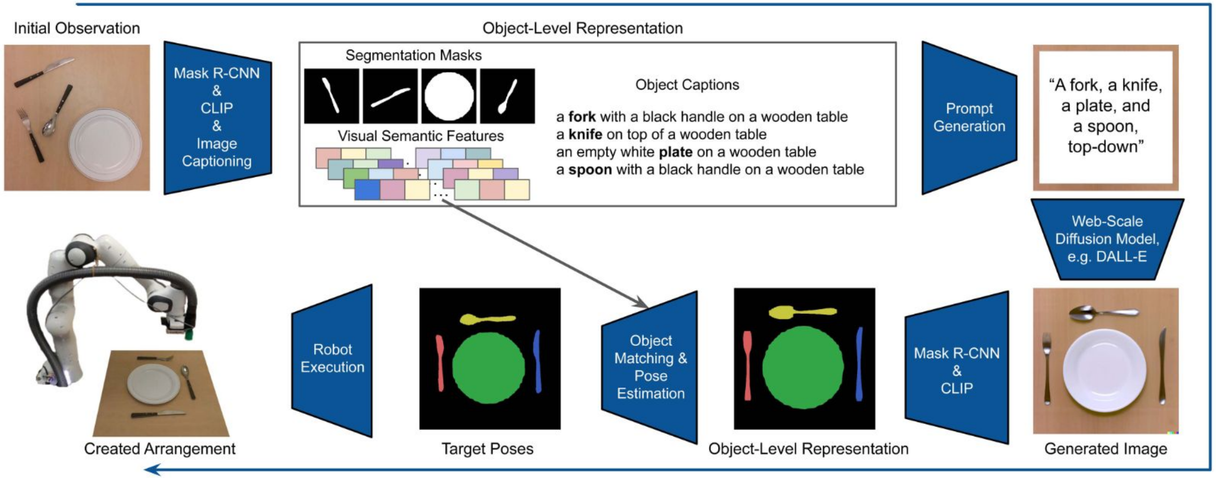
Other methods : Diffusion value functions, Diffusion + Motion Planning [5], Diffusion as a dynamic model

Applications : Offline RL, Online RL, Imitation Learning, Multiagent RL, etc …

[5] Carvalho, Joao, et al. "Conditioned score-based models for learning collision-free trajectory generation." NeurIPS 2022 Workshop

Diffusion Models for Robotics (Task Planning)

DALL-E-Bot (K. Ivan et al. 2023) [6]



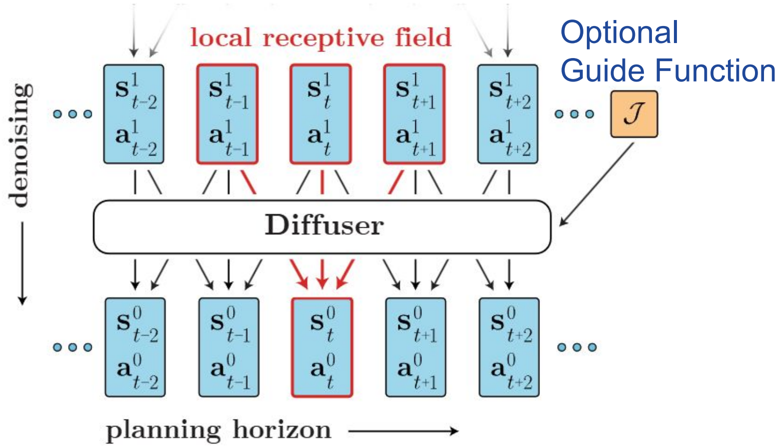
Initial Observation
Mask R-CNN & CLIP & Image Captioning
Object-Level Representation
Segmentation Masks
Visual Semantic Features
Object Captions
a **fork** with a black handle on a wooden table
a **knife** on top of a wooden table
an empty white **plate** on a wooden table
a **spoon** with a black handle on a wooden table
Prompt Generation
"A fork, a knife, a plate, and a spoon, top-down"
Web-Scale Diffusion Model, e.g. DALL-E
Generated Image
Mask R-CNN & CLIP
Object-Level Representation
Object Matching & Pose Estimation
Target Poses
Robot Execution
Created Arrangement

- ❖ Zero-shot : Only uses pre-trained DALL-E models with no demonstrations or training
- ❖ Open-set : Not restricted to a specific set of objects or scene (pre-trained on web-scale data)
- ❖ Autonomous : Does not require any user provided goal state specification, supervision

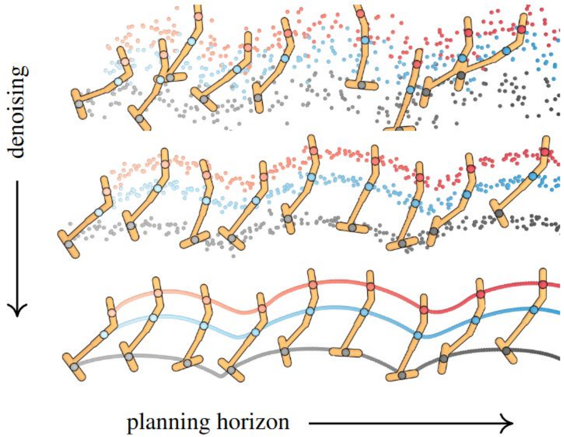
[6] Kapelyukh, Ivan, Vitalis Vosylius, and Edward Johns. "Dall-e-bot: Introducing web-scale diffusion models to robotics." IEEE RA-L(2023

Diffusion Models for Robotics (Trajectory Generation)

Diffuser (Janner et al. 2022) [7]


local receptive field
denoising
$\mathbf{s}^1_{t-2}$ $\mathbf{a}^1_{t-2}$
$\mathbf{s}^1_{t-1}$ $\mathbf{a}^1_{t-1}$
$\mathbf{s}^1_{t}$ $\mathbf{a}^1_{t}$
$\mathbf{s}^1_{t+1}$ $\mathbf{a}^1_{t+1}$
$\mathbf{s}^1_{t+2}$ $\mathbf{a}^1_{t+2}$
$\mathcal{J}$
Diffuser
$\mathbf{s}^0_{t-2}$ $\mathbf{a}^0_{t-2}$
$\mathbf{s}^0_{t-1}$ $\mathbf{a}^0_{t-1}$
$\mathbf{s}^0_{t}$ $\mathbf{a}^0_{t}$
$\mathbf{s}^0_{t+1}$ $\mathbf{a}^0_{t+1}$
$\mathbf{s}^0_{t+2}$ $\mathbf{a}^0_{t+2}$
planning horizon

Optional Guide Function



- ❖ Train a diffusion model to iteratively denoise entire trajectory concurrently. (not autoregressive)
- ❖ Sampling occurs by iteratively refining randomly initialized trajectories (ex. max reward, constraints)
- ❖ Flexible behavior synthesis through composing distribution with different learned guidance function
- ❖ Goal planning through inpainting done by specifying guidance function over explicit goal

[7] Janner, Michael, et al. "Planning with diffusion for flexible behavior synthesis." ICML (2022).

Diffusion Models for Robotics (Trajectory Generation)

Diffuser (Janner et. al 2022) [7] is a seminal work of diffusion model in Offline RL

❖ Diffusion planner as a combination of policy(trajectory optimization) and dynamic model

❖ Numerous follow-up studies : AdaptDiffuser, MetaDiffuser, SkillDiffuser, LatentDiffuser, …

❖ Utilizes classifier guidance : Auxiliary perturbation functions as a guidance function (given or learned)
Classifier is trained on the demonstration data as the diffusion model

Guided Sampling Methods (sampling from conditional distribution for desired attribute)

❖ Classifier guidance (CG) : During sampling, modify the denoising step using classifier's gradient.
(Pro) Have strong control over conditioned generation (Con) Needs additional classifier model

❖ Classifier free guidance (CFG) : Single diffusion model is trained to work with or w/o conditioning
(Pro) This often results stable results. (Con) More sampling cost, two forward passs per step

Diffusion Models for Robotics (Trajectory Generation)

Decision Diffuser (Ajay, et al. 2022) [8]

- ❖ Handles distribution shift problem by generating state sequences with conditional diffusion models followed by inverse dynamic functions to derive executable actions.
- ❖ Why? Decision Diffuser does not requires Q-function, does not face the risk of distribution shift. Generative models are trained with maximum likelihood estimation
- ❖ Utilizes Classifier-free guidance : to find the best trajectory that maximizes the return

  ⇒ From the Ablation study, CFG performed better than Q-function guided CG.

  Decision Diffuser does not requires Q-function, CFG doesn't suffer due to errors in learned Q-functions
- ❖ When agent uses torque control, inverse dynamics is a better alternative to diffusing over actions

  ⇒ Sequence over actions (joint torques) were in high-frequency and less smooth, hard to model

[8] Ajay, Anurag, et al. "Is conditional generative modeling all you need for decision-making?." ICLR (2023).

Future Directions

❖ Integrating explicit, unseen constraints into diffusion-based trajectory generation
⇒ Have to retrain for classifier-free guidance method

❖ Generated trajectories to satisfy safety constraints
⇒ Combining classic control based approaches (ex. Control Barrier Function)

❖ Application to the online RL. DIPO (Yang et. al 2023)
⇒ Expressive multimodal policy can be used for model-free online RL

[9] Yang, Long, et al. "Policy representation via diffusion probability model for reinforcement learning." preprint (2023).

Summary

So far, we briefly went through Offline RL and applications of diffusion models

- Diffusion models have superior expressiveness to model arbitrary distribution.
- Challenges of Offline RL can be mitigated by utilizing diffusion models
- DDPM : Sequentially corrupting training data with slowly increasing noise (forward process), and then learning to reverse this corruption in order (reverse process)

  DDPM proposed a simplified training objective that can be used in various domains.
- Diffusion Models for Robotics : Planning, Policy, Data Augmentation
- Researches in planning :
  - Task level planning : DALL-E-BOT
  - Trajectory level planning : Diffuser (predicts state, action / classifier guidance) ,

    Decision Diffuser (predicts state / classifier-free guidance)

Quiz

1. What is not included in the three main categories of Diffusion model for Robotics?
   a. Planning b. Dynamic Model c. Policy d. Data Augmentation

2. What was the name of the framework introduced as a seminal work for diffusion model for Offline RL?