

# Diffusion Policy

## O. Abstract

1. Diffusion Policy: Visuomotor policy ≈ Conditional denoising diffusion process  $x \in \mathbb{R}^d \rightarrow$  조건 동작  $a \in \mathbb{R}^d$

시각(difusion) 입력을 받아서

조건의 motor action을 만족하는

Policy 찾기

④ DPM: 흐름 확률 모델 (마코프 체인)

(2015)



DDPM: MSE  
(2020)  
↓

DDIM: ACF로 학습  
(2021)  
↓

Diffusion Policy: DDPM 학습 + DDIM 학습 + Policy

2. (정동분포의 초기 흐름 기울기 학습(DDPM 학습))  $\xrightarrow{\text{정동분포}} (\mathcal{E} \sim N(0, I) / \bar{\alpha}_k = \text{hyperparameter})$

정동분포 Largevin 동역학을 통해 기울기에 따른  $x$  변화 추적(DDIM 학습) (기울기는 학습 시 얻은 것 참고/denoising step)

$$x_{k-1} = \underbrace{x_k}_{\text{정동시작점}} + \underbrace{\nabla_{x_k} \log p(x_k | c)}_{\text{정동 방향}} + \underbrace{\sqrt{\tau_k} \xi_k}_{\text{(인간 부여 K의)}}$$

→ k: 흐름(denoising) Step 인덱스 ( $\rightarrow 0 \text{으로 역선했어}/\text{step} = \text{denoising 번째 흐름}$ )

$\alpha_k$ : Step 크기 ( $k$ 마다 다음) = gradient 적용 정도 ] 두다 학습 시 사용한  $\bar{\alpha}_k$ 를 이용해서 공식이 의해 값 정해짐

$\tau_k$ : 학습 noise의 표준편차

$\xi_k$ : Step마다 뽑는 표준 Gaussian noise ( $N(0, I)$ )

$$\hookrightarrow x_k = \frac{\bar{\alpha}_{k-1} - \bar{\alpha}_k}{\sqrt{\bar{\alpha}_{k-1} \bar{\alpha}_k}}, \quad \tau_k = \sqrt{\frac{1 - \bar{\alpha}_{k-1}}{1 - \bar{\alpha}_k} \times \frac{\bar{\alpha}_{k-1} - \bar{\alpha}_k}{\bar{\alpha}_{k-1}}}$$

임의 k

1...K<sub>train</sub>

$\langle x_{DPM}(\text{정동}) x_0 + \sqrt{\tau} C + \text{임의의 노이즈 단계}(k) \rangle$  일 때의

$\langle x_{(k, k, c)} \rangle$  를 통한  $\mathcal{L} = \|\mathcal{E} - \mathcal{E}_\theta(x_{(k, k, c)})\|^2$  (MSE)

정동학습 학습, 즉  $\mathcal{E}$ 가 정동학습 학습

= 기울기 학습

$$\hookrightarrow \nabla_{x_k} \log p(x_{(k, k, c)}) = -\frac{1}{\tau_k} \mathcal{E}_\theta(x_{(k, k, c)}) \quad (\theta = \text{parameter})$$

⇒ denoising 학습 X,  $(x_{(k, k, c)})$  일 때 학습  
수행하지 않음

$\mathcal{E}$ : 흐름 때 쓰는 forward noise

$\xi$ : 흐름 때 쓰는 backward noise

임의)  $x_{k, k, c}$

$x_k$  (이미 noise 섞어진 상태),  $k, c$



$$x_k = \sqrt{\bar{\alpha}_k} x_0 + \sqrt{1 - \bar{\alpha}_k} \mathcal{E}$$

↓ MSE

noise  $\mathcal{E}$  빼주도록

$$\hat{\mathcal{E}} = \mathcal{E}_\theta(x_{k, k, c}) \text{ 예측}$$

# I. Introduction

1.

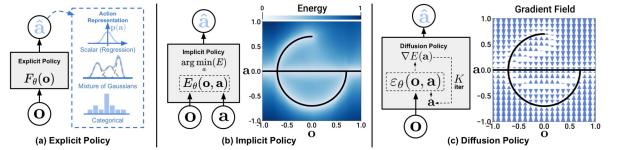


Figure 1. Policy Representations. a) Explicit policy with different types of action representations. b) Implicit policy learns an energy function conditioned on both action and observation and optimizes for actions that minimize the energy landscape c) Diffusion policy refines noise into actions via a learned gradient field. This formulation provides stable training, allows the learned policy to accurately model multimodal action distributions, and accommodates high-dimensional action sequences.

## (a) Explicit Policy

→ 관찰  $o$  (신경망) 가 두 가지로 둘 중  $\hat{a}$  를 찾는  $F_\theta(o)$

## (b) Implicit Policy

→ 미지수 찾기  $E_\theta(o, a)$  찾기

↓

관찰  $o$  기반 최소  $\hat{a}$  찾기

## (c) Diffusion Policy

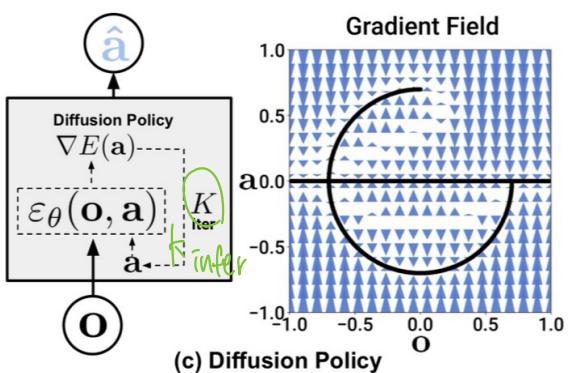
→  $\hat{a}$  를 바로 찾지 X

( $K_{\text{infer}}$ )

→ 관찰  $o$  기반  $\nabla E(a)$  를 파악 ( $K_{\text{train}}$  를 통해 학습)

→ noise 를 시작하는  $\hat{a}$  를  $\nabla E(a)$  를 따라  $K_{\text{train}}$  학습으로 진짜값으로 최종 얻는  $\hat{a}$

→  $E$  함수 자체 학습 X 기울기를 학습 ( $\nabla E(a) = \mathcal{E} o | \theta_\theta$ )



## 2. 다음으로 학습 분포의 확장 O

→ 기울기 학습 + Langevin 학습  $\Rightarrow$  실제 path O

→ 고차원 경로 추적 (high sequence)

→ diffusion은 고차원 경로 추적 한 번에 여러 step seq 학습 O (시간 일관성 O)

온갖 학습

→  $E$  기반 학습 학습 (그림 b) : 확률 학습 확장이 학습 불안정  $\Rightarrow$   $E$  기반 학습

### 3. closed-loop action sequences

→ 한 단계의 행동을 예측하고, 그 중 일부 몇步만 실행 후  $\text{MSE} \Omega_t(\hat{x}) = 0.012$  를 계산  $\Rightarrow$  Receding Horizon  
 예측  $\Omega_t(\hat{x})$  (T<sub>p</sub> 깊이 예측)  
 sliding window (만족하는 경우)

시점  $t+T_a$  까지 예측  $\Omega_{t+T_a}$   
 사용한  $T_a$  이후에 예측은 T<sub>p</sub> 깊이 만큼하고 또 앞 있음을 사용

### Visual conditioning

→  $\Omega_t(\hat{x}) = 0.012$ 는 C(조건)로 하여 충분히 포함 X → 개선  $\downarrow$  (방법은 denoising)  
 → 이미지는 noise X, clip embedding 같은 조건으로 사용

### Time-series diffusion transformer

→ VS CNN (시계열을 over-smoothing : 과도한 블러 필터링 (영화화))  
 → token을 보고 transformer

## 2. Diffusion Policy Formulation

### 2.1 Denoising Diffusion Probabilistic Models

$$x^{k-1} = \alpha(x^k - \nabla_\theta(x^k, k) + \mathcal{N}(0, \sigma^2 I)) : \text{역방향 (Denoising)} \quad \Rightarrow K_{\text{infer}} \text{ 만족 반복함}$$

수학적 예측 혹은 시각적 노이즈  
(보다 낮아짐, 일정성)

$$\begin{aligned} \nabla_{x_k} \log p(x_k | c) &= -\frac{1}{\tau_k} \varepsilon_\theta(x_k, k, c) \\ &= x_k - \frac{\alpha_k}{\tau_k} \varepsilon_\theta(x_k, k, c) + \nabla_{x_k} \varepsilon_\theta(x_k, k, c) \end{aligned}$$

$$\begin{aligned} &= x_k - \frac{\alpha_k}{\tau_k} \varepsilon_\theta(x_k, k, c) + \nabla_{x_k} \varepsilon_\theta(x_k, k, c) \\ &\sim N(0, I) \end{aligned}$$

$\sim N(0, \sigma^2 I) :$  혹은 시각적 노이즈  $\leftarrow$  "아직 작은 예측한 noise"  $\rightarrow$  MSE

↳ "Stochastic Langevin Dynamics"

: 짧은 시간 간격 (Sampling)

가우시안 짧은 샘플링  $x_k \xrightarrow{K_{\text{inter}} \text{ 만족}} x_0$  (단계)

↳ denoising 결과가 평균으로 애인하지 않은 하나의 허를 얻기 위해 증가 (평균 추적의 한 단계)

↳ step 기동될 때, 간소화하고 상관 X + 평균 = 0인 Gaussian noise  
(: 수학적 예측)

이므로 방향 평균 X

↓ 대체로 같은 식

$$(1) \quad \mathbf{x}' = \mathbf{x} - \gamma \nabla E(\mathbf{x})$$

## 2. hyper parameter

→ 학습률:  $\bar{\alpha}_k$  (noise scheduler),  $K_{\text{train}}$   
 학습률:  $\text{Kinfer}$  (, 수동학습  $\alpha$ )

## 2.2 DDPM Training

1. 손실함수:  $\mathcal{L} = \text{MSE}(\epsilon_g^k, \epsilon_g(x^0 + \epsilon^k, k))$

↓  
Image noise

→ denoising 과정을 통해  $q(x_t)$  및 원본 데이터 분포  $p(x_0)$  사이의 KL 베이스의 연결성을 찾는다

## 2.3 Diffusion for visumotor Policy Learning

1. 기본 DDPM의  $\pi = \text{DPM}(\pi_t | \theta)$

↓

2. 기본의 Visumotor:  $\pi = \text{Visumotor}$   
 $\pi_t(a_t | o_t)$ 은  $C(o_t)$ 를 하여 denoising

2.

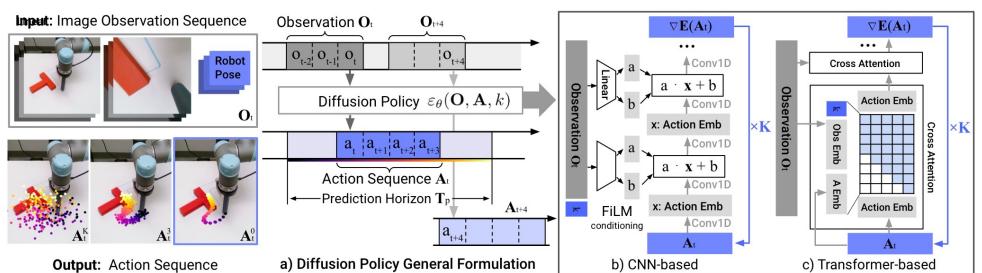


Figure 2. Diffusion Policy Overview a) General formulation. At time step  $t$ , the policy takes the latest  $T_o$  steps of observation data  $O_t$  as input and outputs  $T_a$  steps of actions  $A_t$ . b) In the CNN-based Diffusion Policy, FiLM (Feature-wise Linear Modulation) Perez et al. (2018) conditioning of the observation feature  $O_t$  is applied to every convolution layer, channel-wise. Starting from  $A_t^K$  drawn from Gaussian noise, the output of noise-prediction network  $\epsilon_\theta$  is subtracted, repeating  $K$  times to get  $A_t^K$ , the denoised action sequence. c) In the Transformer-based Vaswani et al. (2017) Diffusion Policy, the embedding of observation  $O_t$  is passed into a multi-head cross-attention layer of each transformer decoder block. Each action embedding is constrained to only attend to itself and previous action embeddings (causal attention) using the attention mask illustrated.

<Diffusion Policy>

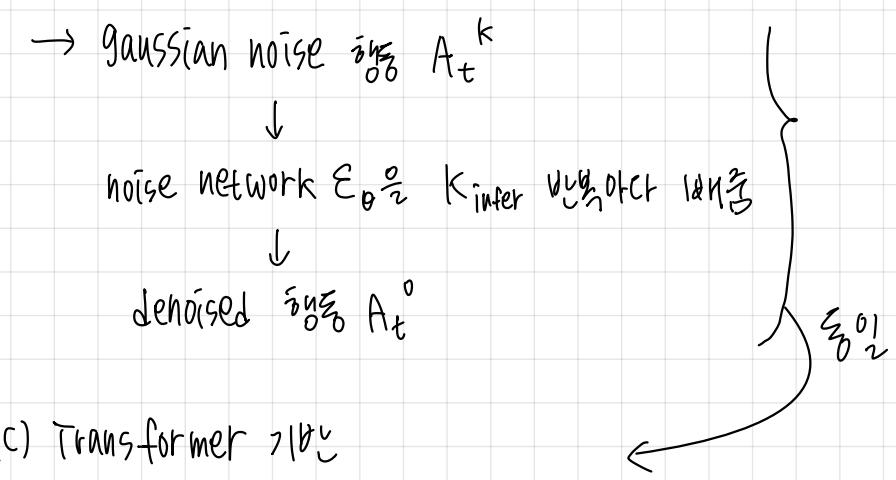
(a) 일반 정식화

→ 시점 + 시간  $T_o$  깊이의 관찰 데이터 (초기화된 이미지 + 노이즈)  $O_t$  를

관찰으로 받아서  $T_a$  깊이의 관찰  $A_t$  예측

(b) CNN 기반

→  $O_t$  특징의 FiLM (Feature-wise Linear Modulation) 을 조정하거나  $\text{Conv}$  층의 channelize



### (c) Transformer 기법

→  $O_t$ 의 임베딩을 CNN(over-smoothing) 대신

Transformer decoder 블록의 multi-head  
Cross-attention으로 替換

↓

attention mask  $\rightarrow$  causal attention

: 자신과 이전imestep 임베이팅만  
attention

### 3. "Closed-loop action-sequence prediction"

→ 시점  $t$ 에서  $T_o$  깊이의 관찰을 대입(E1(초기화된 이미지 임베이팅))  $O_t$ 를

임의로 받아서  $T_p$  깊이의 관찰  $A_t$  예측

↳ 사용은 앞의  $T_a$  깊이 행동만 사용

( $T_o$ : 관찰 지정,  $T_p$ : 행동 예측 지정,  $T_a$ : 행동 실행 지정)

가장 최근 obs  $\Phi T_o$  가  $\Phi T_p \Phi T_a$  깊이는 실제 실행

→ 다음 주제 설정이 warming-start(온도 시작)

### 4. "Visual observation conditioning"

→ 결합 분포  $P(A_t | O_t)$  대신 조건부 분포  $P(A_t | O_t)$ 을 근사

→ 관찰 속도 비례 X, 조건부  $\rightarrow$  속도 ↑, 시간  $T$ , Vision encoder의 end-to-end 학습 가능

→  $\mathcal{L}$

$$\begin{cases} \text{식(1)} \rightarrow A_t^{k-1} = \alpha(A_t^k - \gamma \epsilon_\theta(O_t, A_t^k, k) + \mathcal{N}(0, \sigma^2 I)) \\ \text{식(2)(Loss)} \rightarrow \mathcal{L} = MSE(\epsilon^k, \epsilon_\theta(O_t, A_t^0 + \epsilon^k, k)) \end{cases}$$

식(1) ←  $x^{k-1} = \alpha(x^k - \gamma \epsilon_\theta(x^k, k) + \mathcal{N}(0, \sigma^2 I))$   
식(2) ←  $\mathcal{L} = MSE(\epsilon^k, \epsilon_\theta(x^0 + \epsilon^k, k))$

## 3. Key Design Decisions

### 3.1 Network Architecture Options

#### 1. noise network $\epsilon_\theta$ 의 CNN vs Transformer

↳ CNN-based diffusion policy

→ 1D 시리즈 CNN

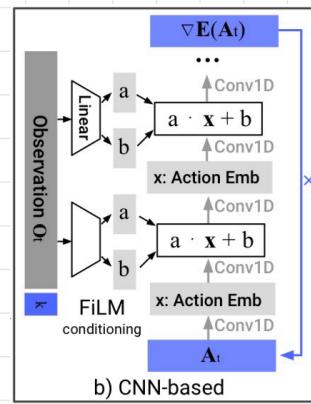
→  $P(A_t | O_t)$ 은 주제별 분포 ( $\leftrightarrow$  확장 분포:  $P(A_t, O_t)$ )

→ 단점: 계층간 (전체  $x$ ; 조건화)

= over-smoothing

⇒ 같은 seq 길이까지  $\epsilon_\theta$ : 가능한 차트 (over-smoothing)

( $\because$  시리즈 CNN의 계층별 영향이 서로 다른 seq)



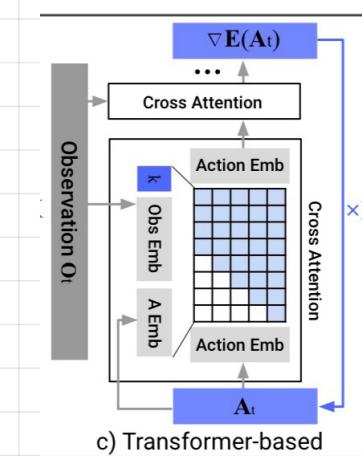
↳ Time-series diffusion transformer

→ CNN의 over-smoothing ⇒ "Transformer"

→  $O_t$ : feature emb + image sequence  
(Visual Encoder) (shared MLP)

$K_{\text{infer}}$  번째 토큰의 Action Emb +  $A_t^k$  은 "token" → transformer decoder  $\Rightarrow \epsilon_\theta(O_t, A_t^k, K)$  ( $K=V$ )  
각 토큰의 토큰으로

→ CNN의 hyperparameter에 더 민감



### 3.2 Visual Encoder

1. 각 시청 영상 encoding 을 모두 concatenate → 카운트  $O_t$

2. Encoder: ResNet 18 (시청 영상 X)

→ Average pooling 및 spatial softmax pooling (같은 영역 유지)

→ Batch Norm → Group Norm 으로 바꿈

DDPM에서 사용되는  
EMA / 평균화 증가  
단계 ↑

### 3.3 Noise schedule

1. MIP 고사인 스케줄

## 3.4 Accelerating Inference for Real-time Control

1. DDIM으로 예측도 실시간 제어를 위한 빠른 추론 속도

$K_{\text{train}} \uparrow$ : 품질화 배우기  
 $K_{\text{infer}} \downarrow$ : 빠른 추론 속도

## 4. Intriguing Properties of Diffusion Policy

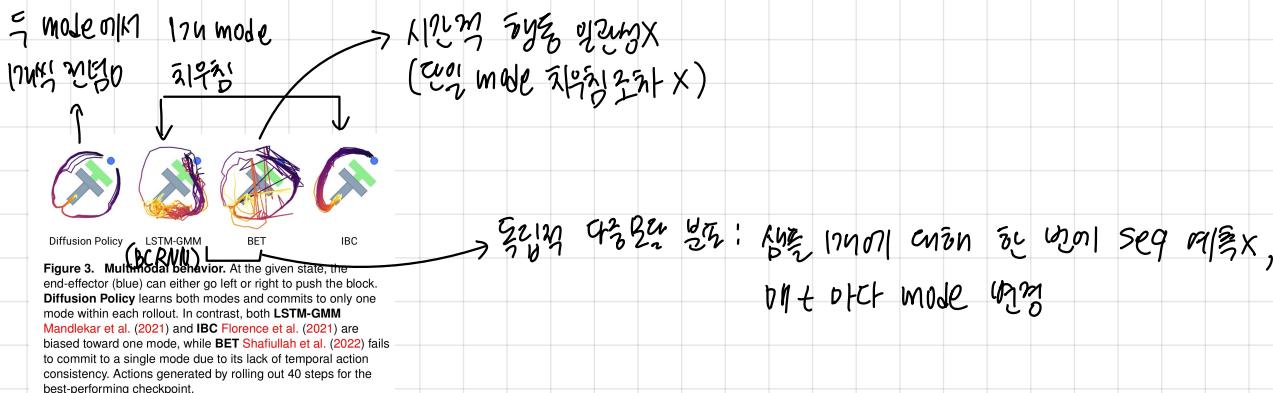
### 4.1 Model Multi-Modal Action Distributions

1. 다중 모델링 방식: Stochastic Sampling + Stochastic Horizon

$$= \text{학습된 각 모드별 확률} = A_t^k \text{의 noise } \sim N(0, I)$$

$$\nabla_k P_k \text{은 } k\text{-th 모드 확률 } \text{의 변화량입니다}$$

2. Push-T 정책의 다중 모드 행동



→ 학습된 확률 0

### 4.2 Synergy with Position control

1. 행동 공간을 Velocity control x

Position control o ( $\theta_0 \uparrow$ )

→ 다중 모델링 더  $\uparrow$

→ Compounding error (누적 오류)가 더  $\downarrow$

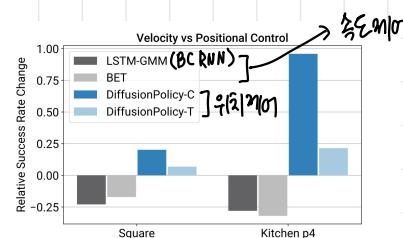


Figure 4. Velocity v.s. Position Control. The performance difference when switching from velocity to position control. While both BCRNN and BET performance decrease, Diffusion Policy is able to leverage the advantage of position and improve its performance.

## 4.3 Benefits of Action-sequence Prediction

1. 다른 policy 학습 방식: 고차원 출력에서 시그모이드 x  
→ IBC, BCRNN, BET

↑

2. DDPM: 행동 seq 예측에 사용 (자연 흐름)
  - 시간적 행동 일관성 (4.1, 2 참고)
  - Idle actions에 대한 강조: 가능성이 x  
→ 일시정지 일대의 동일한 위치 행동의 seq or 속도의 seq의 data가 가능성이 x

## 4.4 Training Stability

1. IBC (Implicit Policy): 이론적으로 diffusion policy와 유사

→ but, "학습 불안정성"

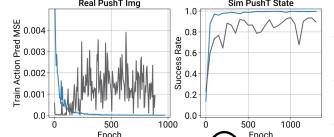


Figure 6. Training Stability. Left: IBC fails to infer training actions with increasing accuracy despite smoothly decreasing training loss for energy function. Right: IBC's evaluation success rate oscillates, making checkpoint selection difficult (evaluated using policy rollouts in simulation).

→ EBM (Energy-based Model)은 행동 분포 P에 영향

$$p_{\theta}(\mathbf{a}|\mathbf{o}) = \frac{e^{-E_{\theta}(\mathbf{o}, \mathbf{a})}}{Z(\mathbf{o}, \theta)}$$

: 미분 가능한  
: 확장 가능성

$\frac{\partial}{\partial \mathbf{a}} \log p(\mathbf{a}|\mathbf{o})$

→ 손실함수

$$\mathcal{L}_{InfoNCE} = -\log \left( \frac{e^{-E_{\theta}(\mathbf{o}, \mathbf{a})}}{e^{-E_{\theta}(\mathbf{o}, \mathbf{a})} + \sum_{j=1}^{N_{neg}} e^{-E_{\theta}(\mathbf{o}, \tilde{\mathbf{a}}_j)}} \right)$$

: 미분 가능한 확장 가능성

: negative sample set  $\{\tilde{\mathbf{a}}_j\}_{j=1}^{N_{neg}}$  사용

→ 학습 불안정성

: Diffusion Policy DDPM

$\nabla_{\mathbf{a}} \log p(\mathbf{a}|\mathbf{o}) = -\nabla_{\mathbf{a}} E_{\theta}(\mathbf{a}, \mathbf{o}) - \underbrace{\nabla_{\mathbf{a}} \log Z(\mathbf{o}, \theta)}_{=0} \approx -\nabla_{\mathbf{a}} E_{\theta}(\mathbf{a}, \mathbf{o})$

$\nabla_{\mathbf{a}} E_{\theta}(\mathbf{a}, \mathbf{o}) = \nabla_{\mathbf{a}} (-\frac{1}{2} \mathbf{a}^T \mathbf{a} - \frac{1}{2} \log |\mathbf{a}|^2)$

$\nabla_{\mathbf{a}} E_{\theta}(\mathbf{a}, \mathbf{o}) = \nabla_{\mathbf{a}} (-\frac{1}{2} \mathbf{a}^T \mathbf{a} - \frac{1}{2} \log |\mathbf{a}|^2) = \nabla_{\mathbf{a}} (\frac{1}{2} \mathbf{a}^T \mathbf{a}) = \mathbf{a}$

## 4.5 Connections to Control Theory

1. Linear system:  $s_{t+1} = As_t + Ba_t + w_t$ ,  $w_t \sim \mathcal{N}(0, \Sigma_w)$ .

$$(T_p = \text{예측 단계}) \begin{cases} T_p = 1 : \mathcal{L} = MSE(\varepsilon^k, \varepsilon_{\theta}(s_t, -Ks_t + \varepsilon^k, k)) / \varepsilon_{\theta}(s, a, k) = \frac{1}{\Theta} [a + Ks] / \text{DDIM} : a = -Ks \text{ 및 } Ks \text{ 예측} \\ T_p > 1 : a_{t-p} = -K(A - BK)^T s_t \end{cases}$$

2. Major impact policy ↗ 훈련 non-linear obj. planing 예상 X

## 5. Evaluation

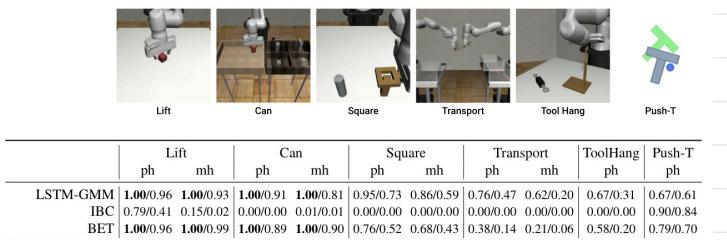
### 5.1 Simulation Environments and dataset

#### 1. Robomimic benchmark 결과

로봇 수 / 각체 수 / 물체 수 / 속도 / 조작 대상 / 단위 / 연관 / 인증									
Task	# Rob	# Obj	ActD	#PH	#MH	Steps	Img?	HiPrec	
<b>Simulation Benchmark</b>									
Lift	1	1	7	200	300	400	Yes	No	
Can	1	1	7	200	300	400	Yes	No	
Square	1	1	7	200	300	400	Yes	Yes	
Transport	2	3	14	200	300	700	Yes	No	
ToolHang	1	2	7	200	0	700	Yes	Yes	
Push-T	1	1	2	200	0	300	Yes	Yes	
BlockPush	1	2	2	0	0	350	No	No	
Kitchen	1	7	9	656	0	280	No	No	
<b>Realworld Benchmark</b>									
Push-T	1	1	2	136	0	600	Yes	Yes	
6DoF Pour	1	liquid	6	90	0	600	Yes	No	
Peri Spread	1	liquid	6	90	0	600	Yes	No	
Mug Flip	1	1	7	250	0	600	Yes	No	

**Table 3. Tasks Summary.** # Rob: number of robots, #Obj: number of objects, ActD: action dimension, PH: proficient-human demonstration, MH: multi-human demonstration, Steps: max number of rollout steps, HiPrec: whether the task has a high precision requirement. BlockPush uses 1000 episodes of scripted demonstrations.

### 5.2 Evaluation Methodology



**Table 1. Behavior Cloning Benchmark (State Policy)** We present success rates with different checkpoint selection methods in the format of (max performance) / (average of last 10 checkpoints), with each averaged across 3 training seeds and 50 different environment initial conditions (150 in total). LSTM-GMM corresponds to BC-RNN in RoboMimic Mandlekar et al. (2021), which we reproduced and obtained slightly better results than the original paper. Our results show that Diffusion Policy significantly improves state-of-the-art performance across the board.

LSTM-GMM 1.00/0.93 1.00/0.91 1.00/0.91 1.00/0.93 1.00/0.97 1.00/0.96 1.00/0.96 1.00/0.96 1.00/0.96 1.00/0.96 1.00/0.96 1.00/0.96

IBC 0.79/0.41 0.15/0.02 0.00/0.00 0.01/0.01 0.00/0.00 0.00/0.00 0.00/0.00 0.00/0.00 0.00/0.00 0.00/0.00 0.00/0.00 0.00/0.00

BET 1.00/0.91 1.00/0.99 1.00/0.89 1.00/0.90 0.76/0.52 0.68/0.43 0.38/0.14 0.21/0.06 0.58/0.20 0.79/0.70

DiffusionPolicy-C 1.00/0.98 1.00/0.97 1.00/0.96 1.00/0.93 1.00/0.97 1.00/0.92 1.00/0.93 1.00/0.97 1.00/0.96 1.00/0.96 1.00/0.96 1.00/0.96

DiffusionPolicy-T 1.00/1.00 1.00/1.00 1.00/1.00 1.00/0.94 1.00/0.89 0.95/0.81 1.00/0.83 1.00/0.98 1.00/0.98 1.00/0.98 1.00/0.98 1.00/0.98

DiffusionPolicy-C 1.00/0.98 1.00/0.97 1.00/0.96 1.00/0.92 1.00/0.98 1.00/0.93 1.00/0.97 1.00/0.96 1.00/0.95 1.00/0.95 1.00/0.95 1.00/0.95

DiffusionPolicy-T 1.00/1.00 1.00/1.00 1.00/1.00 1.00/0.94 1.00/0.89 0.95/0.81 1.00/0.83 1.00/0.98 1.00/0.98 1.00/0.98 1.00/0.98 1.00/0.98

DiffusionPolicy-C 1.00/0.98 1.00/0.97 1.00/0.96 1.00/0.92 1.00/0.98 1.00/0.93 1.00/0.97 1.00/0.96 1.00/0.95 1.00/0.95 1.00/0.95 1.00/0.95

DiffusionPolicy-T 1.00/1.00 1.00/1.00 1.00/1.00 1.00/0.94 1.00/0.89 0.95/0.81 1.00/0.83 1.00/0.98 1.00/0.98 1.00/0.98 1.00/0.98 1.00/0.98

DiffusionPolicy-C 1.00/0.98 1.00/0.97 1.00/0.96 1.00/0.92 1.00/0.98 1.00/0.93 1.00/0.97 1.00/0.96 1.00/0.95 1.00/0.95 1.00/0.95 1.00/0.95

DiffusionPolicy-T 1.00/1.00 1.00/1.00 1.00/1.00 1.00/0.94 1.00/0.89 0.95/0.81 1.00/0.83 1.00/0.98 1.00/0.98 1.00/0.98 1.00/0.98 1.00/0.98

DiffusionPolicy-C 1.00/0.98 1.00/0.97 1.00/0.96 1.00/0.92 1.00/0.98 1.00/0.93 1.00/0.97 1.00/0.96 1.00/0.95 1.00/0.95 1.00/0.95 1.00/0.95

DiffusionPolicy-T 1.00/1.00 1.00/1.00 1.00/1.00 1.00/0.94 1.00/0.89 0.95/0.81 1.00/0.83 1.00/0.98 1.00/0.98 1.00/0.98 1.00/0.98 1.00/0.98

DiffusionPolicy-C 1.00/0.98 1.00/0.97 1.00/0.96 1.00/0.92 1.00/0.98 1.00/0.93 1.00/0.97 1.00/0.96 1.00/0.95 1.00/0.95 1.00/0.95 1.00/0.95

DiffusionPolicy-T 1.00/1.00 1.00/1.00 1.00/1.00 1.00/0.94 1.00/0.89 0.95/0.81 1.00/0.83 1.00/0.98 1.00/0.98 1.00/0.98 1.00/0.98 1.00/0.98

DiffusionPolicy-C 1.00/0.98 1.00/0.97 1.00/0.96 1.00/0.92 1.00/0.98 1.00/0.93 1.00/0.97 1.00/0.96 1.00/0.95 1.00/0.95 1.00/0.95 1.00/0.95

DiffusionPolicy-T 1.00/1.00 1.00/1.00 1.00/1.00 1.00/0.94 1.00/0.89 0.95/0.81 1.00/0.83 1.00/0.98 1.00/0.98 1.00/0.98 1.00/0.98 1.00/0.98

DiffusionPolicy-C 1.00/0.98 1.00/0.97 1.00/0.96 1.00/0.92 1.00/0.98 1.00/0.93 1.00/0.97 1.00/0.96 1.00/0.95 1.00/0.95 1.00/0.95 1.00/0.95

DiffusionPolicy-T 1.00/1.00 1.00/1.00 1.00/1.00 1.00/0.94 1.00/0.89 0.95/0.81 1.00/0.83 1.00/0.98 1.00/0.98 1.00/0.98 1.00/0.98 1.00/0.98

DiffusionPolicy-C 1.00/0.98 1.00/0.97 1.00/0.96 1.00/0.92 1.00/0.98 1.00/0.93 1.00/0.97 1.00/0.96 1.00/0.95 1.00/0.95 1.00/0.95 1.00/0.95

DiffusionPolicy-T 1.00/1.00 1.00/1.00 1.00/1.00 1.00/0.94 1.00/0.89 0.95/0.81 1.00/0.83 1.00/0.98 1.00/0.98 1.00/0.98 1.00/0.98 1.00/0.98

DiffusionPolicy-C 1.00/0.98 1.00/0.97 1.00/0.96 1.00/0.92 1.00/0.98 1.00/0.93 1.00/0.97 1.00/0.96 1.00/0.95 1.00/0.95 1.00/0.95 1.00/0.95

DiffusionPolicy-T 1.00/1.00 1.00/1.00 1.00/1.00 1.00/0.94 1.00/0.89 0.95/0.81 1.00/0.83 1.00/0.98 1.00/0.98 1.00/0.98 1.00/0.98 1.00/0.98

DiffusionPolicy-C 1.00/0.98 1.00/0.97 1.00/0.96 1.00/0.92 1.00/0.98 1.00/0.93 1.00/0.97 1.00/0.96 1.00/0.95 1.00/0.95 1.00/0.95 1.00/0.95

DiffusionPolicy-T 1.00/1.00 1.00/1.00 1.00/1.00 1.00/0.94 1.00/0.89 0.95/0.81 1.00/0.83 1.00/0.98 1.00/0.98 1.00/0.98 1.00/0.98 1.00/0.98

DiffusionPolicy-C 1.00/0.98 1.00/0.97 1.00/0.96 1.00/0.92 1.00/0.98 1.00/0.93 1.00/0.97 1.00/0.96 1.00/0.95 1.00/0.95 1.00/0.95 1.00/0.95

DiffusionPolicy-T 1.00/1.00 1.00/1.00 1.00/1.00 1.00/0.94 1.00/0.89 0.95/0.81 1.00/0.83 1.00/0.98 1.00/0.98 1.00/0.98 1.00/0.98 1.00/0.98

DiffusionPolicy-C 1.00/0.98 1.00/0.97 1.00/0.96 1.00/0.92 1.00/0.98 1.00/0.93 1.00/0.97 1.00/0.96 1.00/0.95 1.00/0.95 1.00/0.95 1.00/0.95

DiffusionPolicy-T 1.00/1.00 1.00/1.00 1.00/1.00 1.00/0.94 1.00/0.89 0.95/0.81 1.00/0.83 1.00/0.98 1.00/0.98 1.00/0.98 1.00/0.98 1.00/0.98

DiffusionPolicy-C 1.00/0.98 1.00/0.97 1.00/0.96 1.00/0.92 1.00/0.98 1.00/0.93 1.00/0.97 1.00/0.96 1.00/0.95 1.00/0.95 1.00/0.95 1.00/0.95

DiffusionPolicy-T 1.00/1.00 1.00/1.00 1.00/1.00 1.00/0.94 1.00/0.89 0.95/0.81 1.00/0.83 1.00/0.98 1.00/0.98 1.00/0.98 1.00/0.98 1.00/0.98

DiffusionPolicy-C 1.00/0.98 1.00/0.97 1.00/0.96 1.00/0.92 1.00/0.98 1.00/0.93 1.00/0.97 1.00/0.96 1.00/0.95 1.00/0.95 1.00/0.95 1.00/0.95

DiffusionPolicy-T 1.00/1.00 1.00/1.00 1.00/1.00 1.00/0.94 1.00/0.89 0.95/0.81 1.00/0.83 1.00/0.98 1.00/0.98 1.00/0.98 1.00/0.98 1.00/0.98

DiffusionPolicy-C 1.00/0.98 1.00/0.97 1.00/0.96 1.00/0.92 1.00/0.98 1.00/0.93 1.00/0.97 1.00/0.96 1.00/0.95 1.00/0.95 1.00/0.95 1.00/0.95

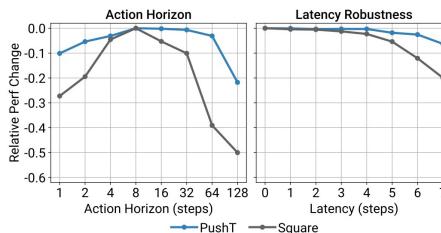
DiffusionPolicy-T 1.00/1.00 1.00/1.00 1.00/1.00 1.00/0.94 1.00/0.89 0.95/0.81 1.00/0.83 1.00/0.98 1.00/0.98 1.00/0.98 1.00/0.98 1.00/0.98

DiffusionPolicy-C 1.00/0.98 1.00/0.97 1.00/0.96 1.00/0.92 1.00/0.98 1.00/0.93 1.00/0.97 1.00/0.96 1.00/0.95 1.00/0.95 1.00/0.95 1.00/0.95

DiffusionPolicy-T 1.00/1.00 1.00/1.00 1.00/1.00 1.00/0.94 1.00/0.89 0.95/0.81 1.00/0.83 1.00/0.98 1.00/0.98 1

## 5.3 Key Findings

1. Diffusion Policy can express short-horizon multimodality (4.1.2)  
→ 201
2. Diffusion Policy can express long-horizon multimodality
3. Diffusion Policy can better leverage position control (4.2.1)
4. The tradeoff in action horizon  
→ 4.3 but, horizon (여기서 말한)  
→ ↑ : ↓ (trade off)  
→ 8 steps ↗
5. Robustness against latency  
→ ∵ Receding Horizon (1.3)
6. Diffusion Policy is stable to train (4.4.1)



**Figure 5. Diffusion Policy Ablation Study.** Change (difference) in success rate relative to the maximum for each task is shown on the Y-axis. **Left:** trade-off between temporal consistency and responsiveness when selecting the action horizon. **Right:** Diffusion Policy with position control is robust against latency. Latency is defined as the number of steps between the last frame of observations to the first action that can be executed.

## 5.4 Ablation Study

### 1. visual Encoder

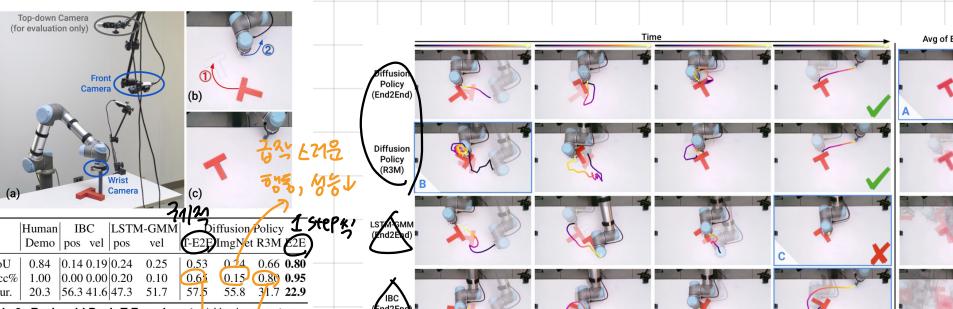
Architecture & Pretrain Dataset	From Scratch	frozen	Pretrained finetuning
Resnet18 (in21)	0.94	0.58	0.92
Resnet34 (in21)	0.92	0.40	0.94
ViT-base (clip)	0.22	0.70	0.98

**Table 5. Vision Encoder Comparison** All models are trained on the robomimic square (ph) task using CNN-based diffusion policy. Each model is trained for 500 epochs and evaluated every 50 epochs under 50 different environment initial conditions.

## 6. Realworld Evaluation

### 6.1 Realworld Push-T Task

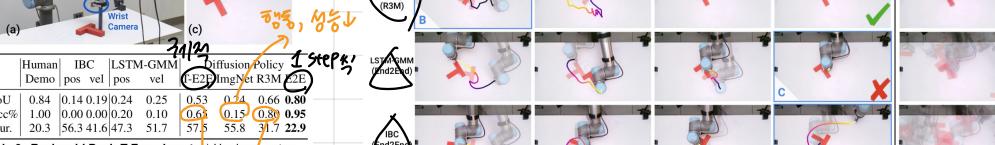
#### 1. End-to-End vs Pre-trained vision encoders



**Table 6. Realworld Push-T Experiment.** a) Hardware setup. b) Illustration of the task. The robot needs (1) precisely push the T-shaped block into the target region, and (2) move the end-effector to the end-zone. c) The ground truth end state used to calculate IoU metrics used in this table. Success is defined by the end-state IoU greater than the minimum IoU in the demonstration dataset. Average episode duration presented in seconds. E2E stands for end-to-end trained Transformer-based Diffusion Policy

→ jittery ↘

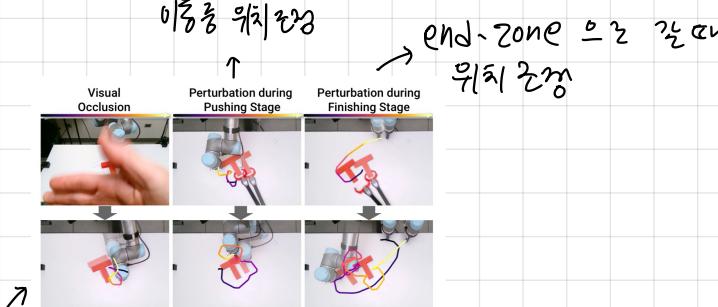
→ jittery ↘



**Figure 7. Realworld Push-T Comparisons.** Columns 1-4 show action trajectories based on key events. The last column shows averaged images of the end state. A: Diffusion policy (End2End) achieves more accurate and consistent end states. B: Diffusion Policy (R3M) gets stuck initially but later recovers and finishes the task. C: LSTM-GMM fails to reach the end zone while adjusting the T block, blocking the eval camera view. D: IBC prematurely ends the pushing stage.

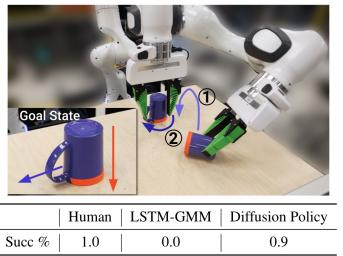
End-to-End < R3M but, R3M: jittery

## 2. Robustness against Perturbation



**Figure 8. Robustness Test for Diffusion Policy.** Left: A waving hand in front of the camera for 3 seconds causes slight jitter, but the predicted actions still function as expected. Middle: Diffusion Policy immediately corrects shifted block position to the goal state during the pushing stage. Right: Policy immediately aborts heading to the end zone, returning the block to goal state upon detecting block shift. This novel behavior was never demonstrated. Please check the videos in the supplementary material.

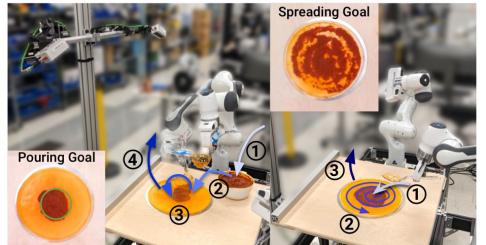
## 6.2 Mug Flipping Task



	Human	LSTM-GMM	Diffusion Policy
Succ %	1.0	0.0	0.9

**Figure 9. 6DoF Mug Flipping Task.** The robot needs to ① Pickup a randomly placed mug and place it lip down (marked orange). ② Rotate the mug such that its handle is pointing left.

## 6.3 Sauce Pouring and Spreading



	Pour IoU	Succ	Spread Coverage	Succ %
Human	0.79	1.00	0.79	1.00
LSTM-GMM	0.06	0.00	0.27	0.00
Diffusion Policy	<b>0.74</b>	<b>0.79</b>	<b>0.77</b>	<b>1.00</b>

**Figure 10. Realworld Sauce Manipulation.** [Left] 6DoF pouring Task. The robot needs to ① dip the ladle to scoop sauce from the bowl, ② approach the center of the pizza dough, ③ pour sauce, and ④ lift the ladle to finish the task. [Right] Periodic spreading Task The robot needs to ① approach the center of the sauce with a grapsed spoon, ② spread the sauce to cover pizza in a spiral pattern, and ③ lift the spoon to finish the task.

## 7. Realworld Bimanual Tasks

### 1. Bimanual : ofte

#### 7.1 observation and Action spaces

#### 7.2 Teleoperation

#### 7.3 Bimanual Egg Beater

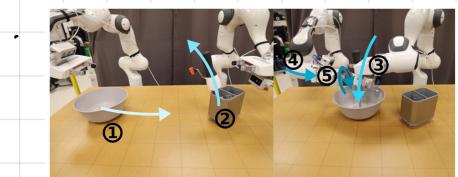


Figure 11. Bimanual Egg Beater Manipulation. The robot needs to ① push the bowl into position (only if too close to the left arm), ② approach and pick up the egg beater with the right arm, ③ place the egg beater in the bowl, ④ approach and grasp the egg beater crank handle, and ⑤ turn the crank handle 3 or more times.

#### 7.4 Bimanual Mat Unrolling

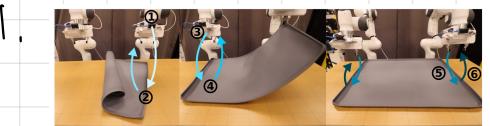


Figure 12. Bimanual Mat Unrolling. The robot needs to ① pick up one side of the mat (if needed), using the left or right arm, ② lift and unroll the mat (if needed), ③ ensure that both sides of the mat are grasped, ④ lift the mat, ⑤ place the mat oriented with the table, mostly centered, and ⑥ release the mat.

#### 7.5 Bimanual Shirt Folding

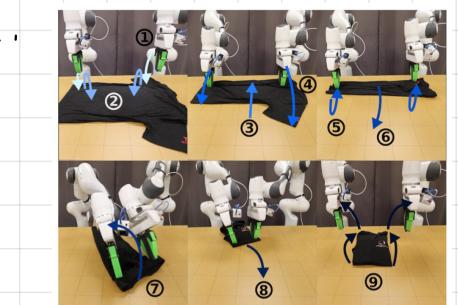


Figure 13. Bimanual Shirt Folding. The robot needs to ① approach and grasp the closest sleeve with both arms, ② fold the sleeve and release, ③ drag the shirt closer (if needed), ④ approach and grasp the other sleeve with both arms, ⑤ fold the sleeve and release, ⑥ drag the shirt to a orientation for folding, ⑦ grasp and fold the shirt in half by its collar, ⑧ drag the shirt to the center, and ⑨ smooth out the shirt and move the arms away.

## 8. Related Work

1. EXPLICIT POLICY
2. IMPLICIT POLICY
3. DIFFUSION MODELS

## 9. Limitations and Future Work

1. LSTM-GMM 보다 계산비용·학습 시간 ↑

## 10. Conclusions