World Model-Based Vision Language Action Model

2026. 08. 28

정재우

Data Mining & Quality Analytics Lab.

Data Mining
Quality Analytics

발표자 소개

- **정재우(Jae woo Jung)**
  - 고려대학교 산업경영공학과 대학원 재학
    - 석사 과정 (2025.09 ~ present)

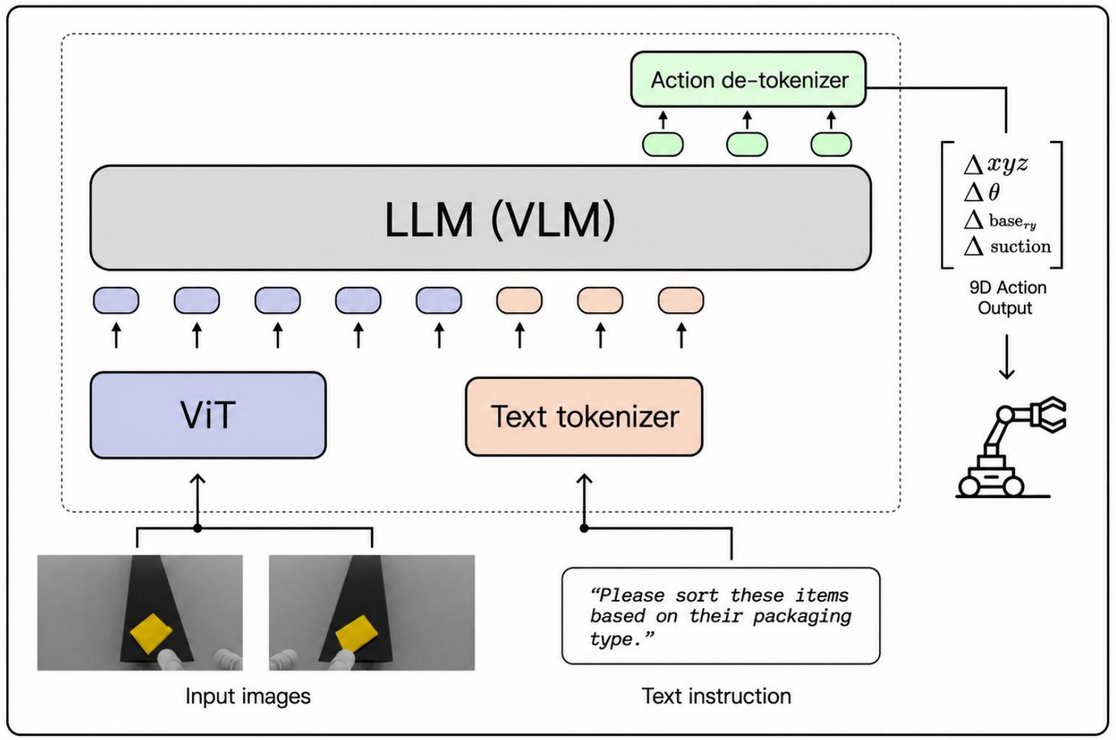
- **Research Interest**
  - Reinforcement Learning
    - Preference-based Reinforcement Learning
    - World Model
  - VLA & Physical AI

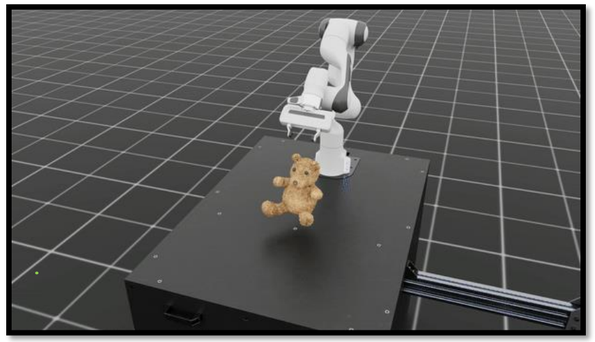
- **Contact**
  - **jjw0513@korea.ac.kr**

Preliminaries

Vision Language Model

- **Vision Language Action(VLA)**
  - 카메라 영상과 사람의 말을 함께 입력 받아, 로봇의 동작을 곧바로 만들어내는 모델
  - 대량의 데이터로 사전 학습한 VLM을 로봇 데이터로 추가 학습하여 제어 테스크에 유연하게 맞추는 방법론
  - Ex :「빨간 컵을 집어」테스크를 녹인 지시어를 입력하면 새로운 물체나 배치 등 여러 요구 동작 에도 대응 가능

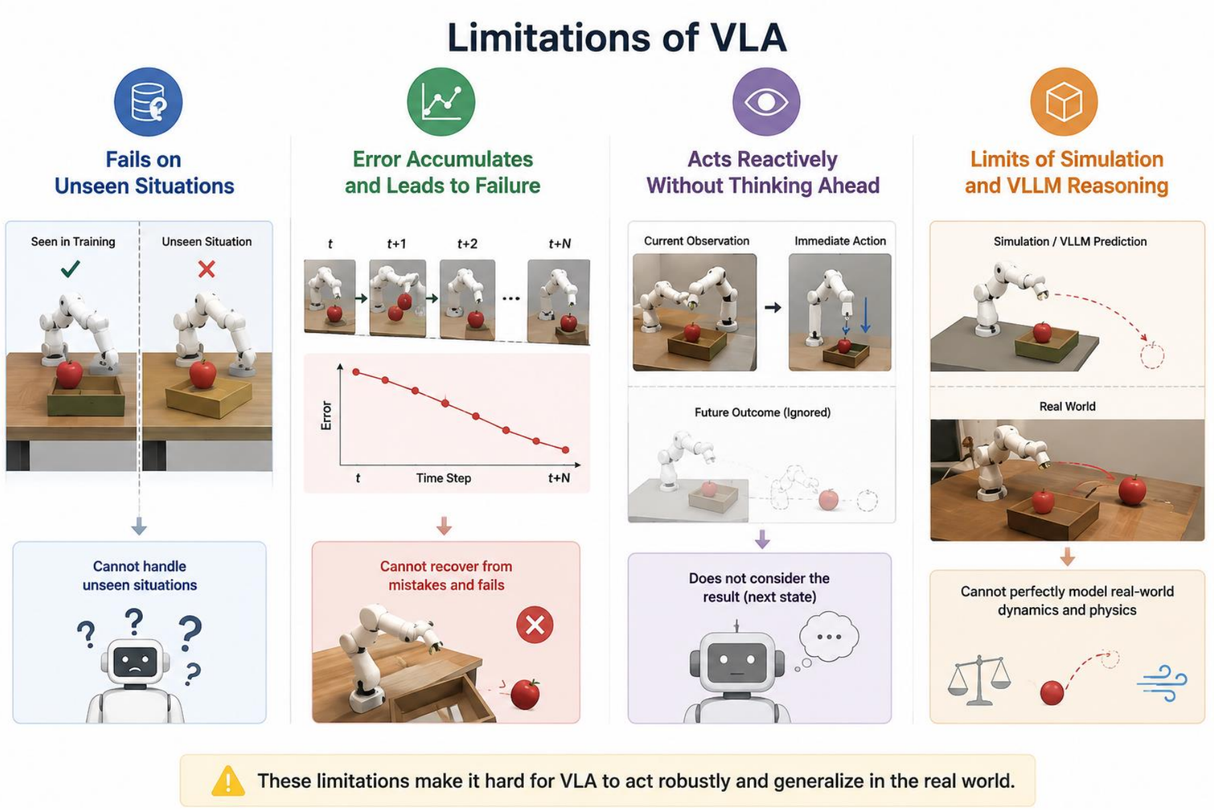




Preliminaries

Limitation of VLA

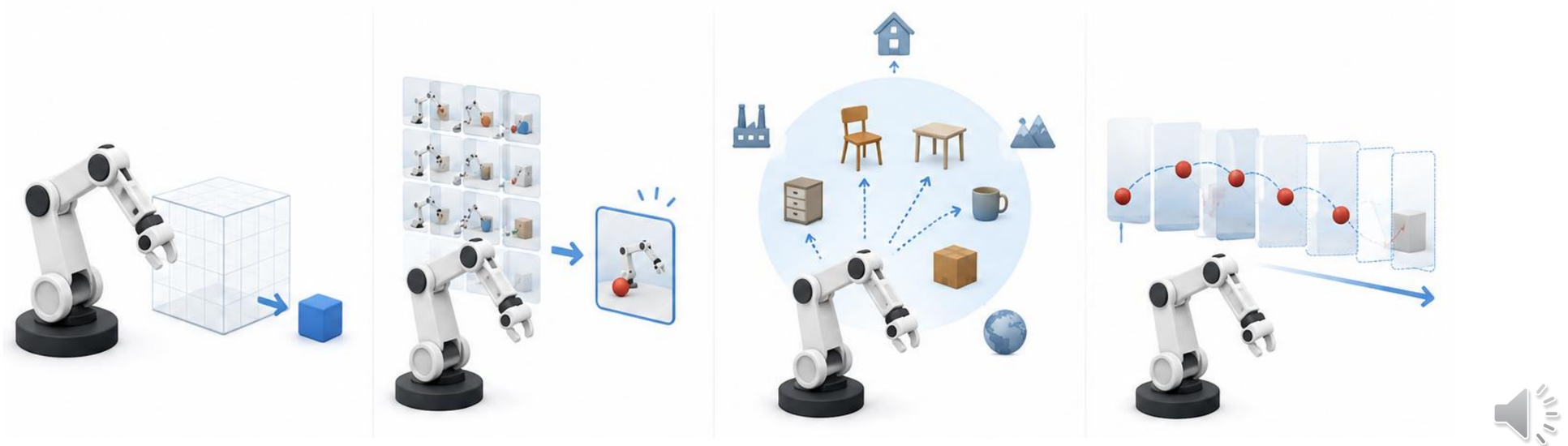
- **Limitation of VLA**
  - 학습 데이터에 없던 상황을 만나면 대처하지 못함
  - 한 번 어긋나면 오차가 쌓이고, 스스로 되돌리지 못한 채 실패로 이어짐
  - 화면을 보고 곧바로 동작을 내놓을 뿐, 동작의 결과(다음 상태)를 생각하지 않음
  - 시뮬레이션 및 VLLM의 추론 공간은 실세계의 동역학 및 물리 법칙을 재현하는데 한계가 존재



Preliminaries

Research Trends of VLA

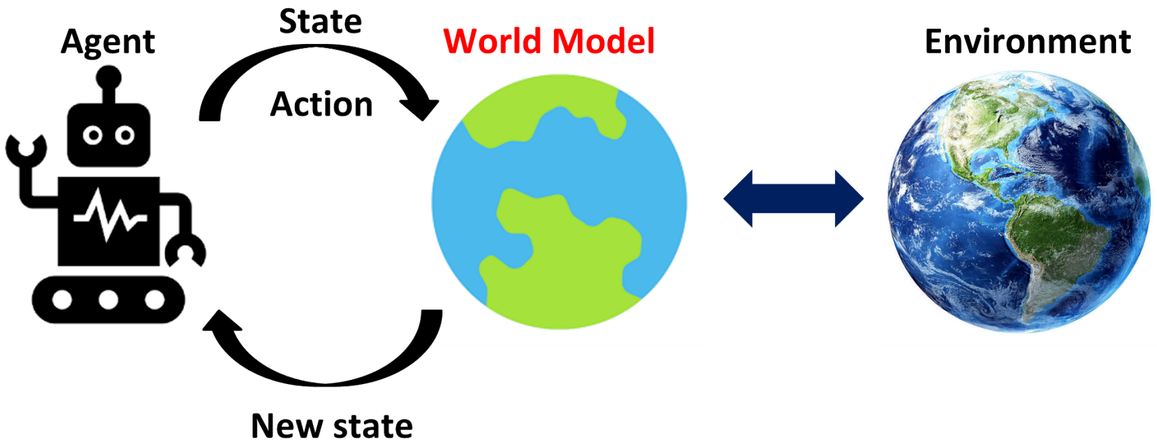
- **Research trends of VLA**
  - 경량화 : 큰 모델의 사이즈를 줄여 로봇 위에서 실시간 추론 및 성능 보장
  - 데이터 효율성 : 인간의 시연 혹은 로봇 데이터로도 강건한 학습
  - 일반화 : 처음 보는 물체 · 배치 · 환경에서도 동작
  - **동역학 이해 & 미래 예측 : 행동의 결과를 미리 그려보고 계획 -> World Model 도입**

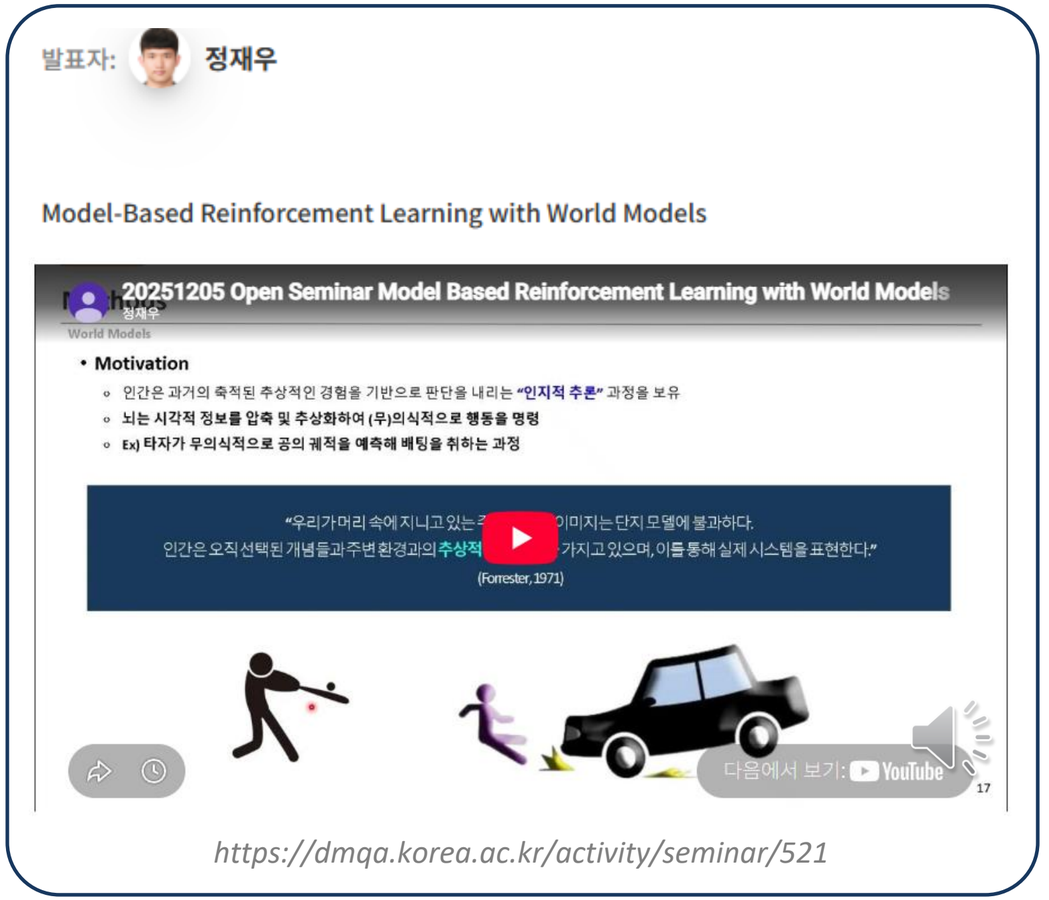


Preliminaries

World Model

- **What is world model?**
  - 행동을 하면 세상이 어떻게 바뀔지 미리 그려보는 내부 모델
  - 실제로 행동을 취하지 않고, 시뮬레이션으로 여러 상상을 시도할 수 있어, 데이터 및 환경과의 상호작용에 대한 부담이 감소
  - 잠재 공간 내 시뮬레이션 과정에서 테스크 환경의 동역학을 반영 가능



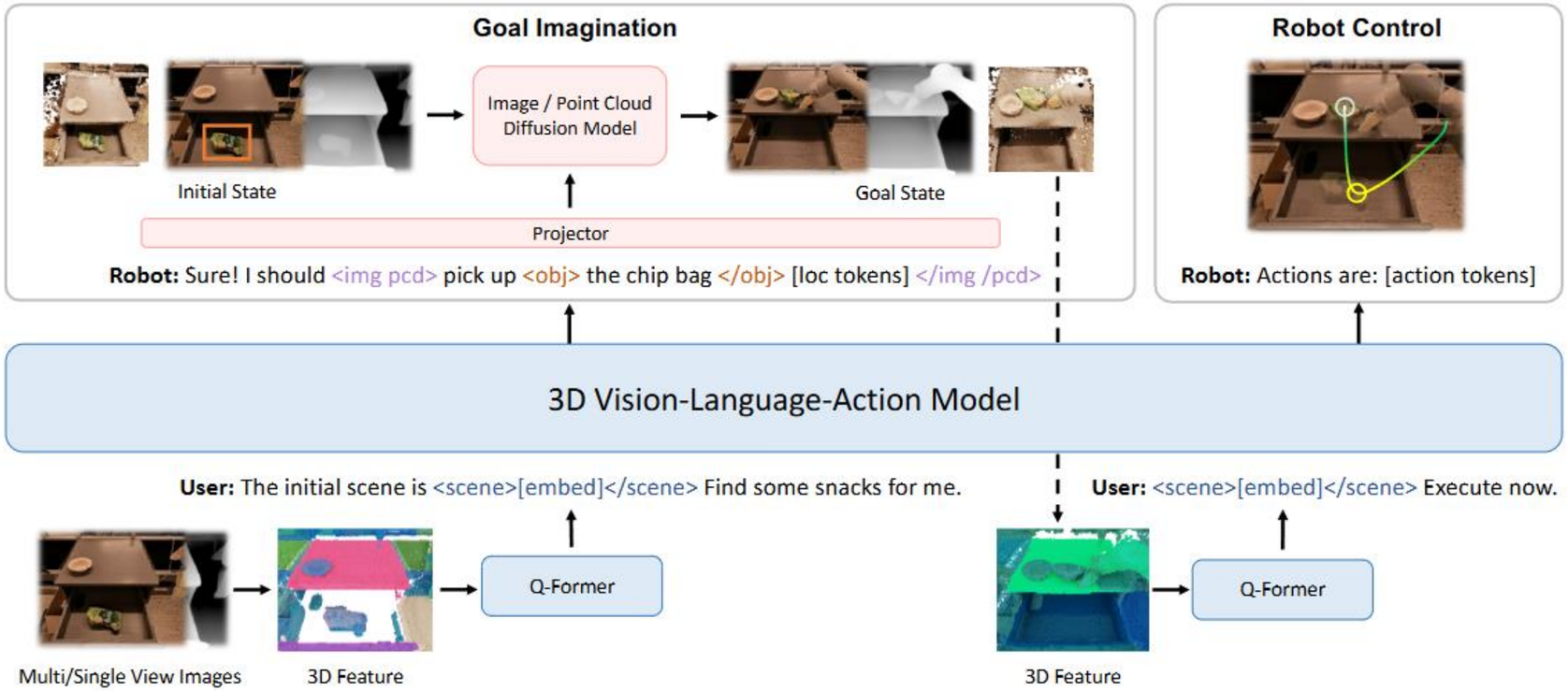


https://dmqa.korea.ac.kr/activity/seminar/521

Methods (1)

3D-VLA

- **3D-VLA : A 3D Vision-Language-Action Generative World Model (ICML, 2024)**
  - **모델이 스스로 도달하고자 하는 목표 상태를 생성하여, 이를 바탕으로 적절한 행동을 계획**
    - 목표 장면을 미리 생성하여, 모델이 특정 시점의 장면(상태) 및 행동과 목표 장면 간의 오차를 줄여 나감
  - 기존 VLA의 2D 영상(이미지) 데이터를 3D 공간 데이터로 재구성
  - 3D-LLM + Projector + Diffusion decoder를 통해 모달리티 정렬 및 물리적 환경에 대응



Zhen, H., Qiu, X., Chen, P., Yang, J., Yan, X., Du, Y., ... & Gan, C. (2024). 3d-vla: A 3d vision-language-action generative world model. *arXiv preprint arXiv:2403.09631*.

Methods (1)

3D-VLA

- **Motivation**
  - 최근 VLA 모델은 2D 입력에 의존하지만, 실제 세계는 3D로 구성되어 있으며 간격을 해소하는 연구 부재
    - EX : 「가장 멀리 있는 컵을 가운데 서랍에 넣어라」와 같은 공간 명령을 수행할 수 없음
  - 더불어 시각 데이터에 대한 지각을 통해 실제 행동을 결정하지만, 실제 세계의 역학(dynamics)과 행동 간의 관계를 간과
  - 궁극적으로 인간이 3D 물리적 세계를 이해하고 계획 및 행동하는 것처럼, 3D 데이터를 다루며 이해하는 지능이 필요

① **Reliance on 2D inputs**
2D 이미지에 국한되어 3D 물리 세계에 대처하지 못함

② **Direct perception to action mapping**
실세계의 동역학 및 행동과 역학간 관계 고려 부재

③ **Lake of 3D-annotated data**
기존 embodied 데이터셋은 2D 중심, 3D 주석 부재

기존 VLA

① **3D-based backbone + Interaction tokens**
3D 장면 특징을 인코딩하고 3D 환경과 상호작용할 어휘 정의

② **Generative world model**
목표 이미지 및 포인트 클라우드를 상상(예측)해 행동을 계획

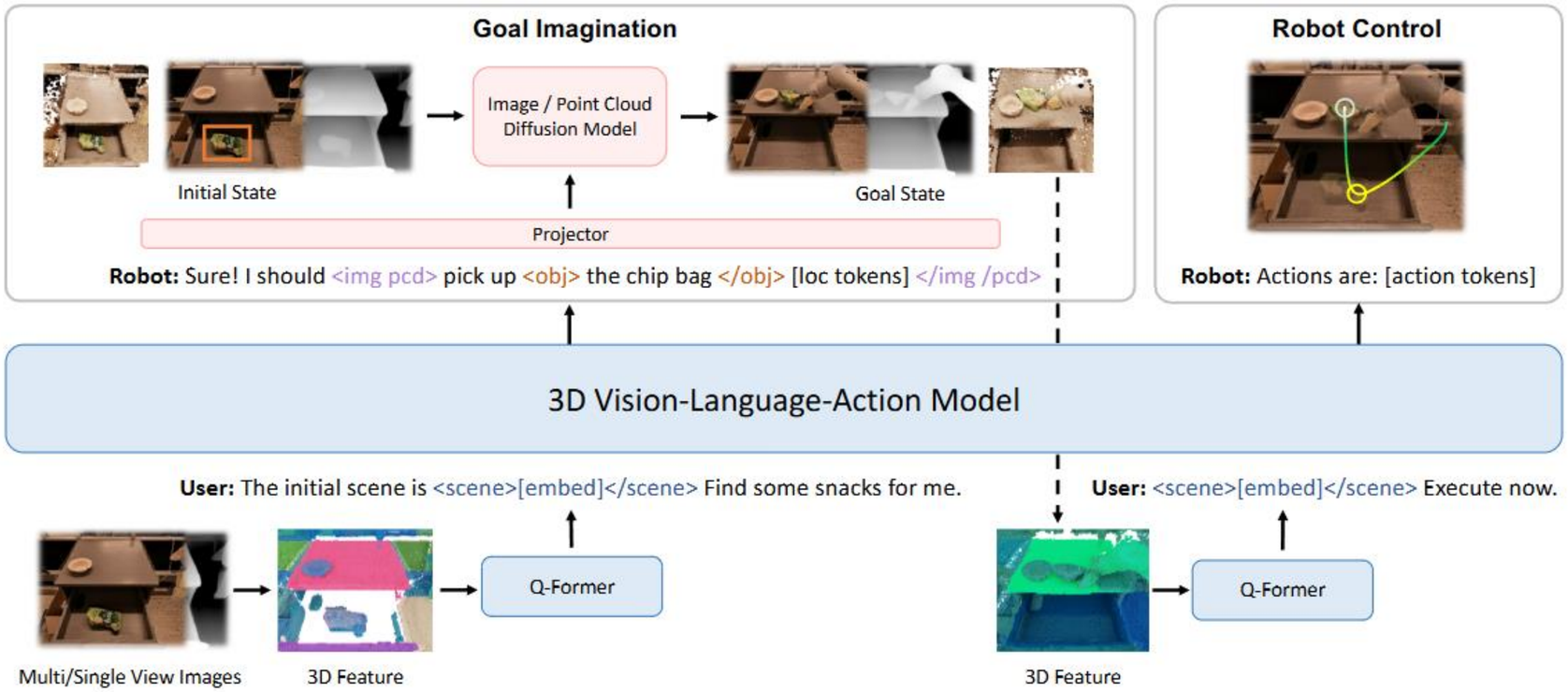
③ **3D Embodied Instruction Tuning Dataset**
깊이 추정 기반 파이프라인으로 3D 데이터 확보

3D-VLA

Methods (1)

3D-VLA

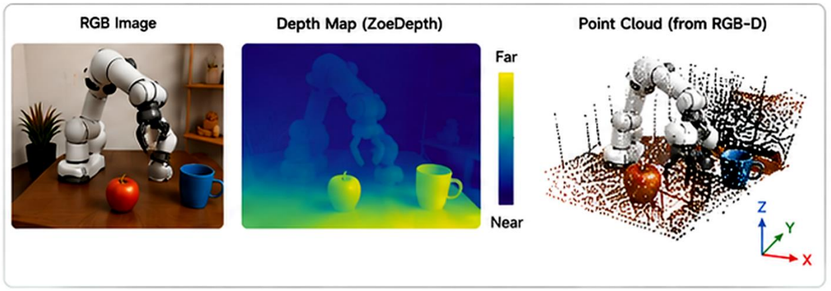
- **Overview of 3D-VLA**
  - 이미지 입력을 Q-Former로 처리하여 3D 공간 정보를 압축하고, 이를 LLM의 입력 토큰으로 주입
  - LLM은 입력 장면과 명령으로 미래의 목표 상태를 예측하고, Projector를 통해 디퓨전 모델에 정보를 전달하여 미래의 목표 상태를 생성
  - LLM은 현재의 상태와 상상한 목표 상태를 비교하여, 그 차이를 좁히기 위한 최적의 '로봇 조작 액션 토큰'을 순차적으로 출력
  - 생성된 목표 상태는 LLM으로 재입력되어 로봇이 목표에 도달할 때까지 행동을 수정 및 제어 반복
  - 「**지각 → 상상 → 행동**」3단 구성이며, Goal Imagination 단계가 이 논문의 world model에 해당

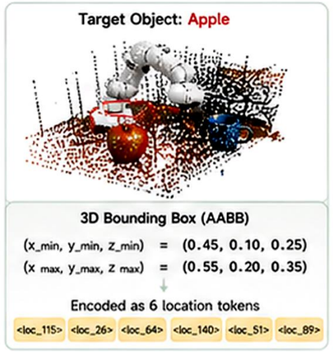


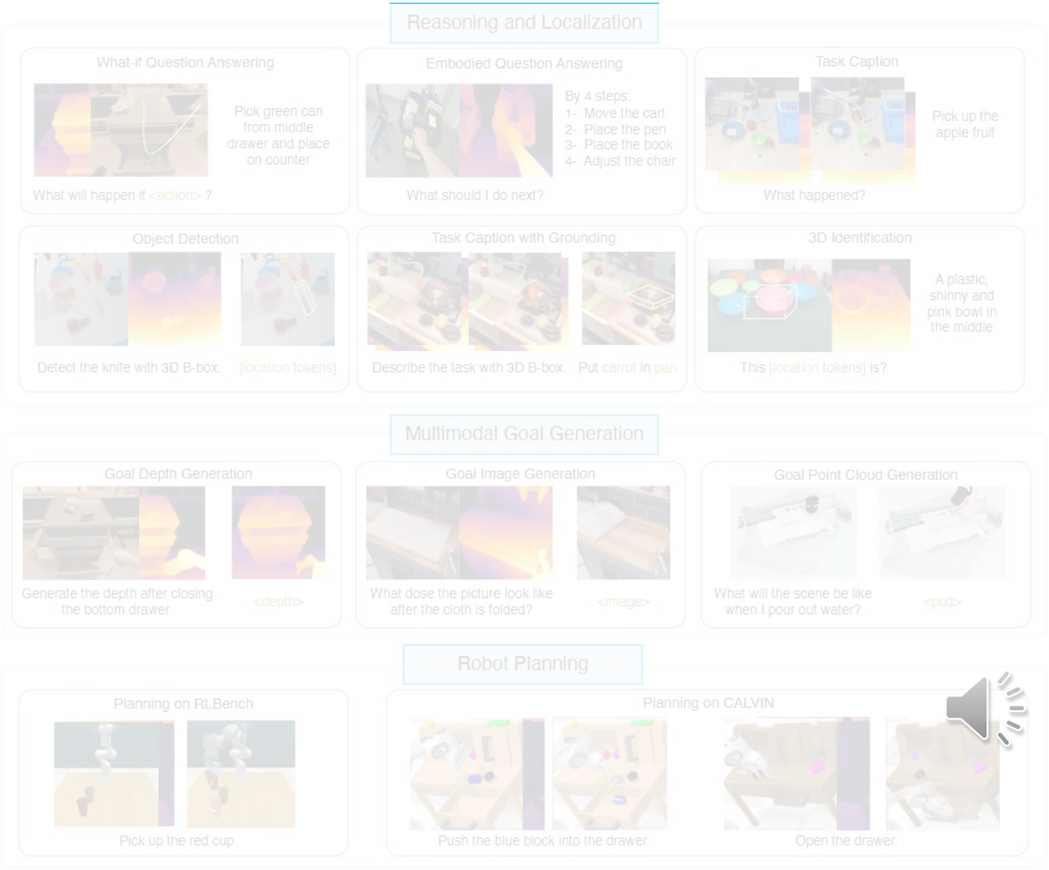
Methods (1)

3D-VLA

- **3D data for learning**
  - 기존 2D 로봇 영상 데이터의 정적인 배경과 동적인 객체를 분리하여 3D 공간 정보를 재구성
  - 확보된 3D 데이터와 ChatGPT를 활용하여 로봇의 동작을 설명하는 3D 기반 QA 및 지시어 쌍을 대규모로 자동 생성
  - 사용자와 ChatGPT가 확보한 3D 기반 QA & instruction 쌍으로 LLM 파인 튜닝



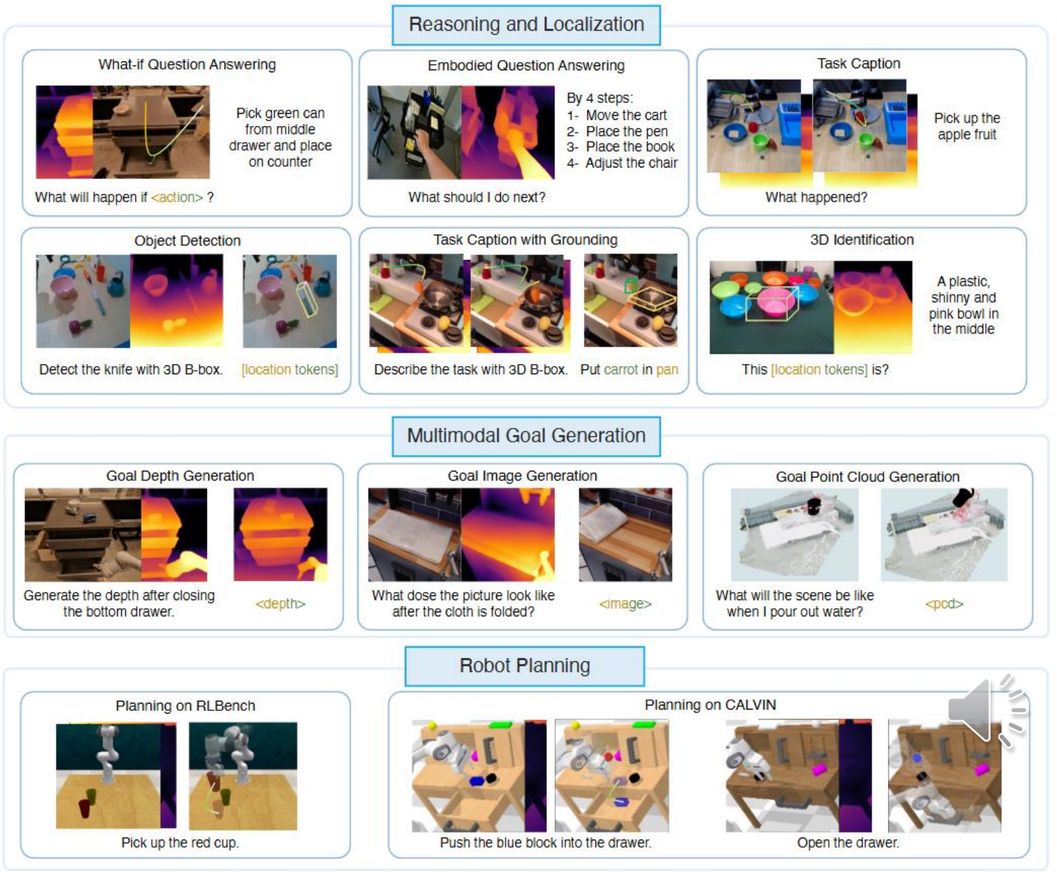




Methods (1)

3D-VLA

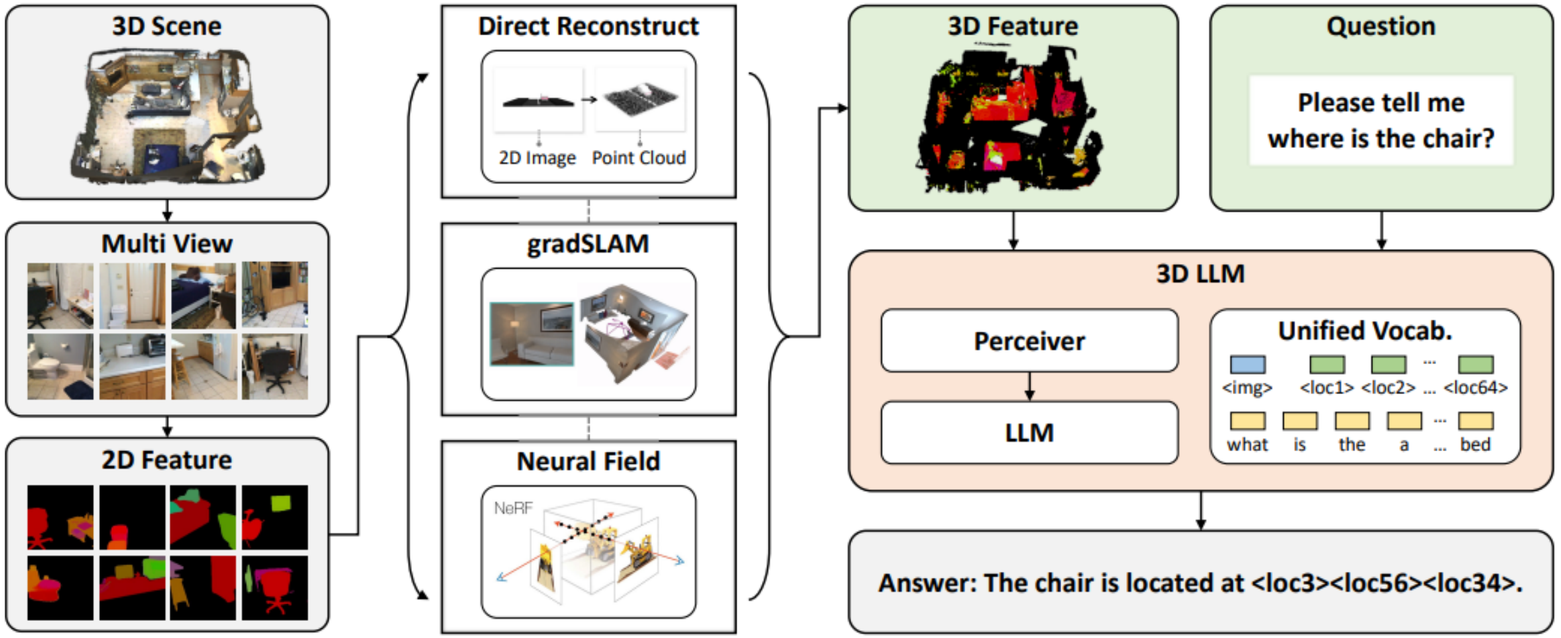
- **3D data for learning**
  - 기존 2D 로봇 영상 데이터의 정적인 배경과 동적인 객체를 분리하여 3D 공간 정보를 재구성
  - 확보된 3D 데이터와 ChatGPT를 활용하여 로봇의 동작을 설명하는 3D 기반 QA 및 지시어 쌍을 대규모로 자동 생성
  - 사용자와 ChatGPT가 확보한 3D 기반 QA & instruction 쌍으로 LLM 파인 튜닝

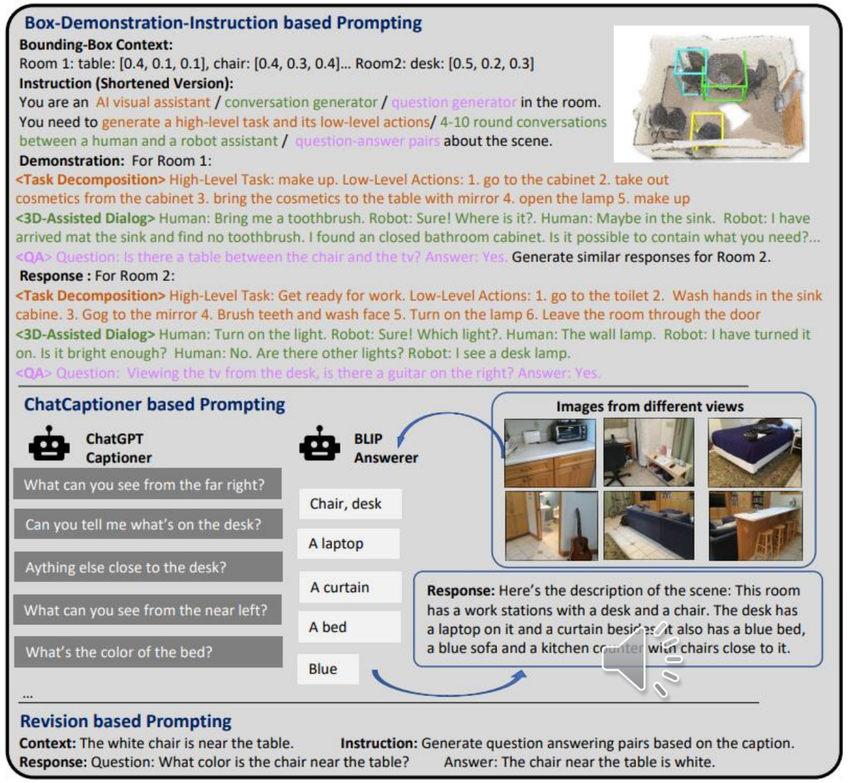


Methods (1)

3D-VLA

- **Methodology : Backbone (3D LLM)**
  - 3D 공간 이해가 필요하지만 기존 VLM 구조와 호환되지 않음 → 멀티뷰 특징을 Q-Former로 압축하는 3D-LLM 방식 채택
  - 하지만 3D LLM은 로봇이나 제어 데이터가 아닌 공간이나 내부 구조 등을 위주로 학습하였으므로, 그대로 사용하기 어려움
  - 따라서 BLIP2 FlanT5의 사전 학습 가중치를 접목



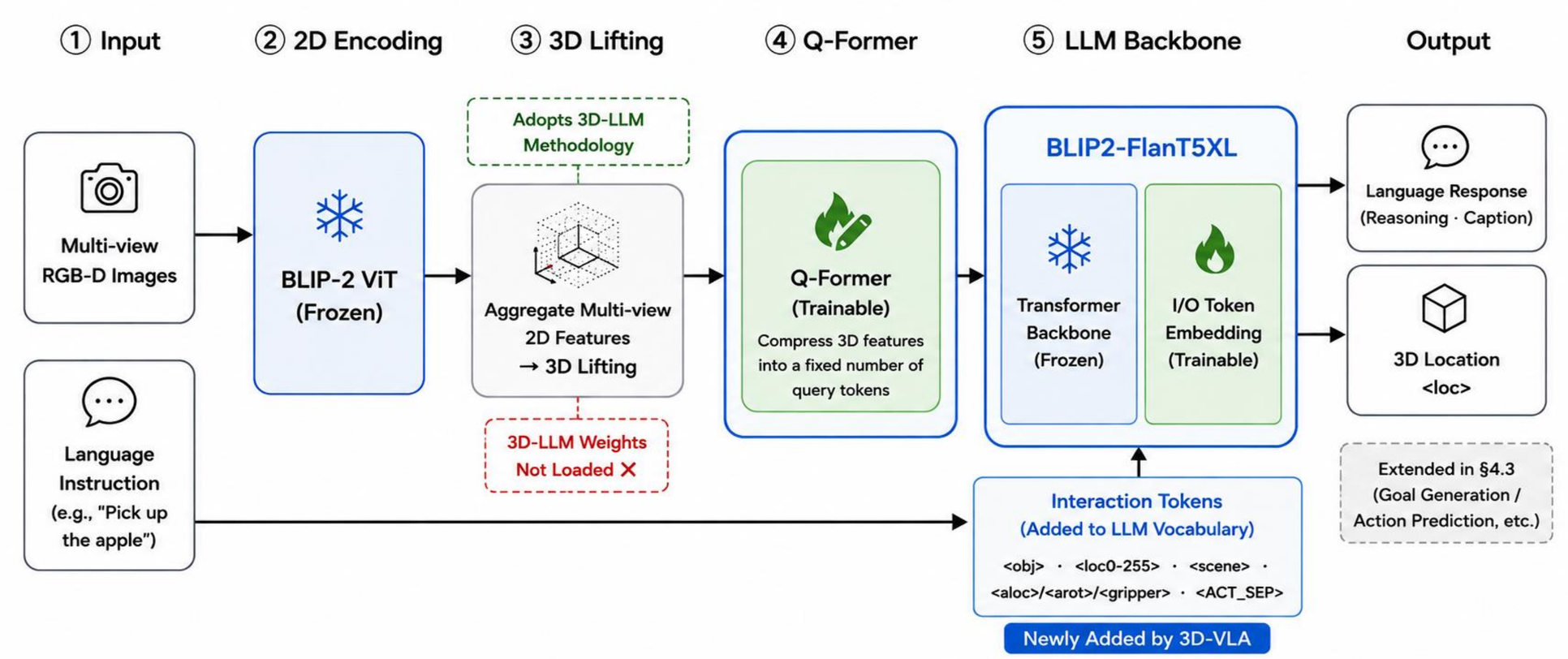


Hong, Y., Zhen, H., Chen, P., Zheng, S., Du, Y., Chen, Z., & Gan, C. (2023). 3d-llm: Injecting the 3d world into large language models. *Advances in Neural Information Processing Systems*, *36*, 20482-20494.

Methods (1)

3D-VLA

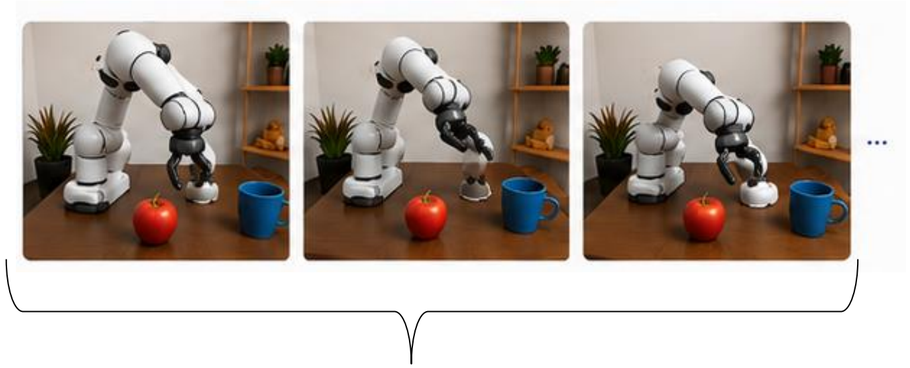
- **Methodology : Backbone (3D LLM + Blip2-FlanT5)**
  - 3D 공간 이해가 필요하지만 기존 VLM 구조와 호환되지 않음 → 멀티뷰 특징을 Q-Former로 압축하는 3D-LLM 방식 채택
  - 로봇 데이터가 대규모가 아니므로 scratch learning 불가 → 시각·언어 정렬이 끝난 BLIP2-FlanT5XL을 백본으로 재활용
  - 단, 3D-LLM은 정적 장면 위주 사전 학습 되었으므로, 가중치는 로드하지 않고 토큰 임베딩과 Q-Former만 학습



Methods (1)

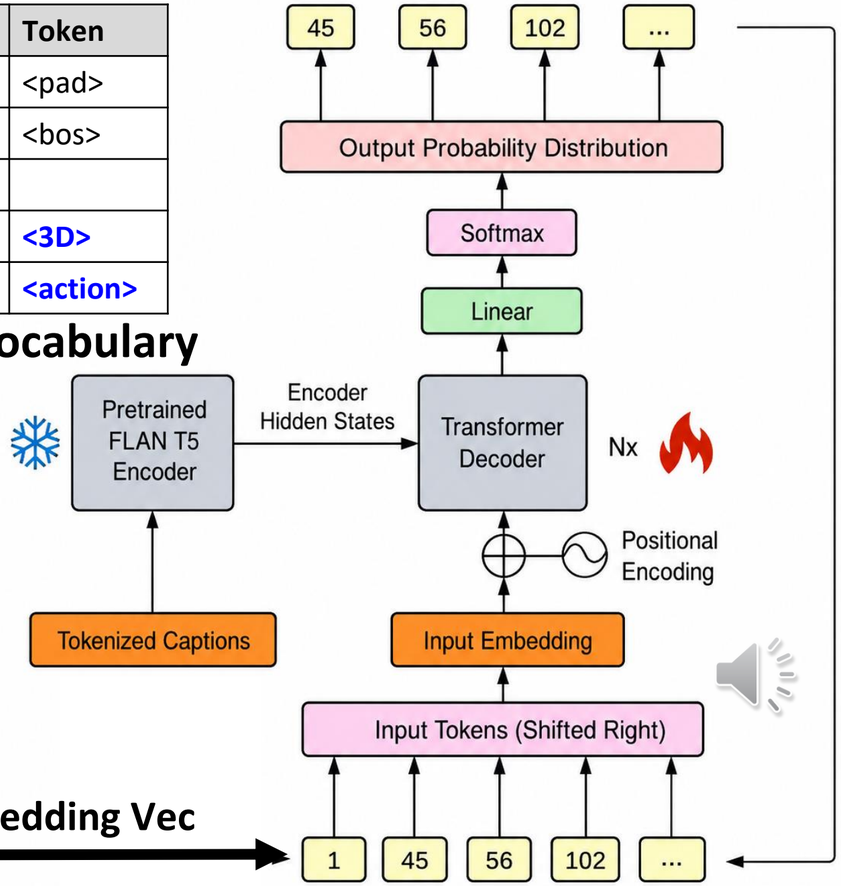
3D-VLA

- **Methodology : Backbone (3D LLM + Blip2-FlanT5)**
  - 3D 공간 이해가 필요하지만 기존 VLM 구조와 호환되지 않음 → 멀티뷰 특징을 Q-Former로 압축하는 3D-LLM 방식 채택
  - 로봇 데이터가 대규모가 아니므로 scratch learning 불가 → 시각·언어 정렬이 끝난 BLIP2-FlanT5XL을 백본으로 재활용
  - 단, 3D-LLM은 정적 장면 위주 사전 학습 되었으므로, 가중치는 로드하지 않고 토큰 임베딩과 Q-Former만 학습



ID	Token
0	<pad>
1	<bos>
...	
N+1	<3D>
N+2	<action>

T5 Vocabulary




```
{
  "id": "robot_data_001",
  "instruction": "The initial scene is <scene>[Scene_Embedding]</scene>.
                  Pick up the apple.",
  "output": "I will execute the following steps:
                  <ACT_START> <aloc128> <arot64> <gripper1>
                  <ACT_SEP> <aloc130> <arot64> <gripper0> <ACT_END>",
  "task_type": "action_prediction",
  "metadata":
                  {"object_name": "apple",
                  "location_token": "<loc102>"}
}
```

Flatten & Tokenizer

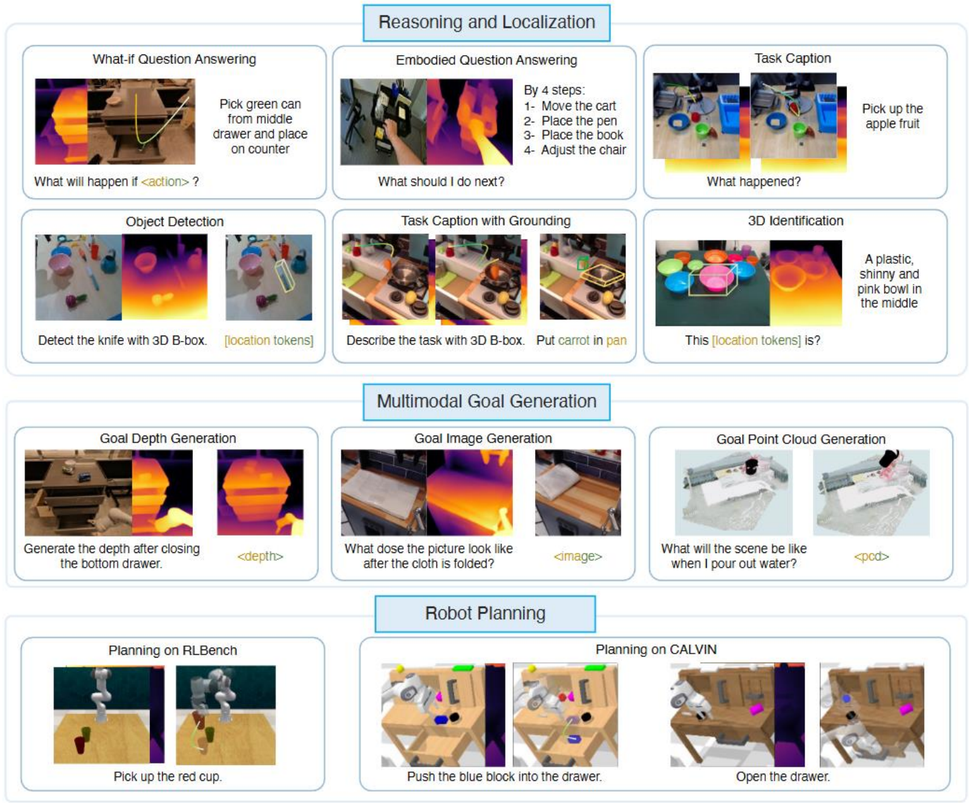
[The, initial, scene, is, <scene>, (3D_Embeddings), </scene>, Pick, up, ...<ACT_START>~<ACT_END>]

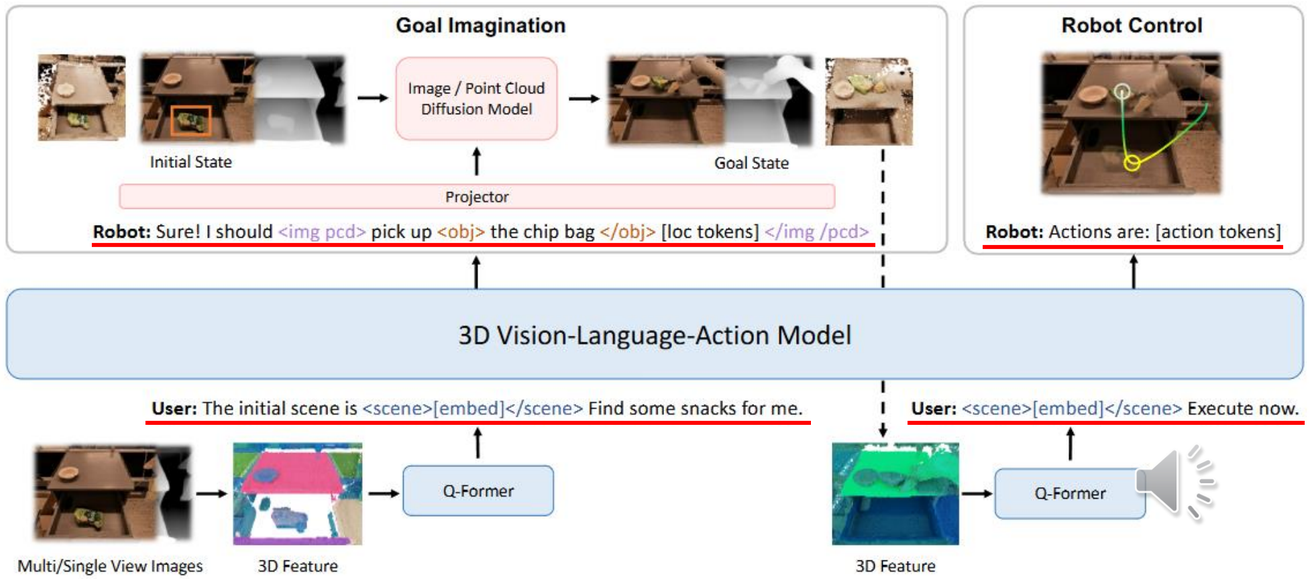
Embedding Vec

Methods (1)

3D-VLA

- **Methodology : Interaction Tokens**
  - 3D 공간의 동적 이해와 로봇의 물리적 상호작용을 위해 LLM vocabulary 확장
  - 조작 대상을 특정하는 <obj> 토큰, 3D 위치를 표현하는 <loc> 토큰, 다이내믹한 3D 장면을 인코딩하는 <scene> 토큰 도입
  - 7 DoF 로봇 팔의 제어 명령(<aloc>, <arot>, <gripper>)을 별도의 액션 토큰으로 분절하여 명령-수행의 연결성 확보





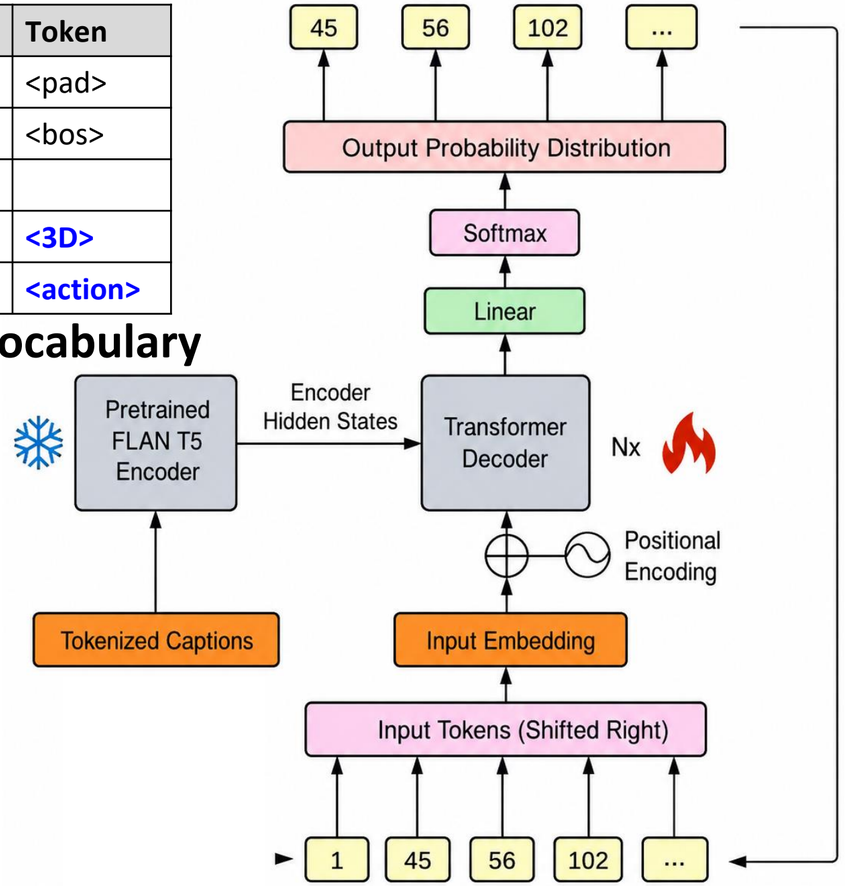
Methods (1)

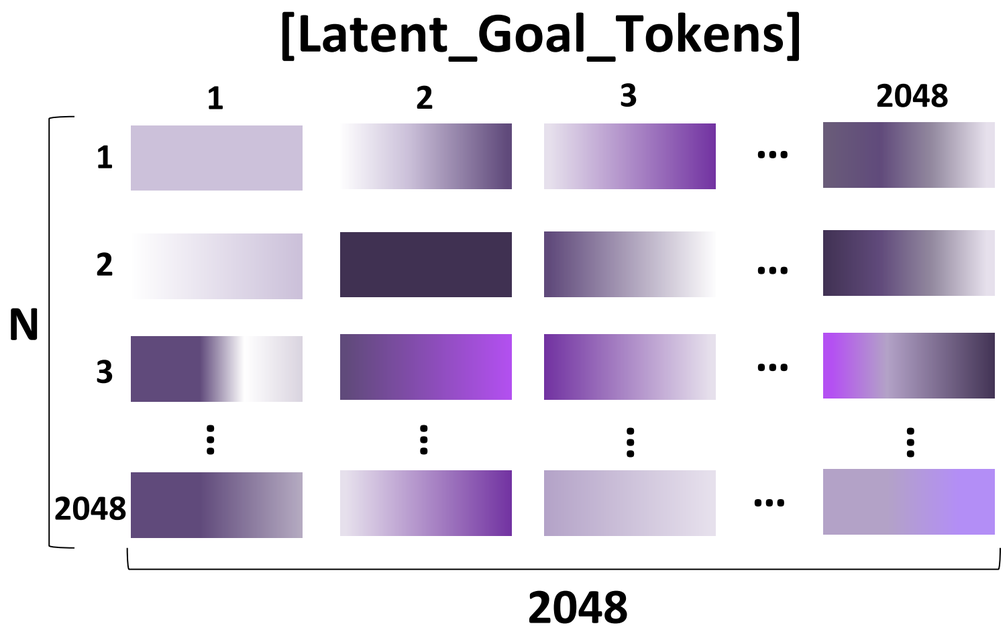
3D-VLA

- **Methodology : Backbone (3D LLM + Blip2-FlanT5)**
  - 3D 공간 이해가 필요하지만 기존 VLM 구조와 호환되지 않음 → 멀티뷰 특징을 Q-Former로 압축하는 3D-LLM 방식 채택
  - 로봇 데이터가 대규모가 아니므로 scratch learning 불가 → 시각·언어 정렬이 끝난 BLIP2-FlanT5XL을 백본으로 재활용
  - 단, 3D-LLM은 정적 장면 위주 사전 학습 되었으므로, 가중치는 로드하지 않고 토큰 임베딩과 Q-Former만 학습

ID	Token
0	<pad>
1	<bos>
...	
N+1	<3D>
N+2	<action>

T5 Vocabulary





3D-VLA answer :

"Sure! I should pick up the <obj> apple </obj> at <loc102>. I will first imagine the goal state.
<image> [Latent_Goal_Tokens] </image> <pcd> **[Latent_Pcd_Tokens]** </pcd>.
Now I will execute: <ACT_START> <aloc...>..."

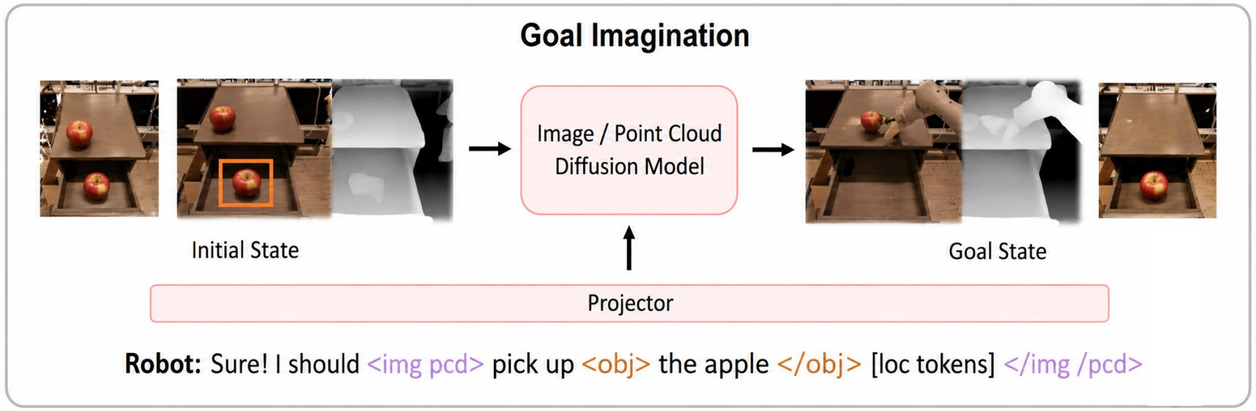
Methods (1)

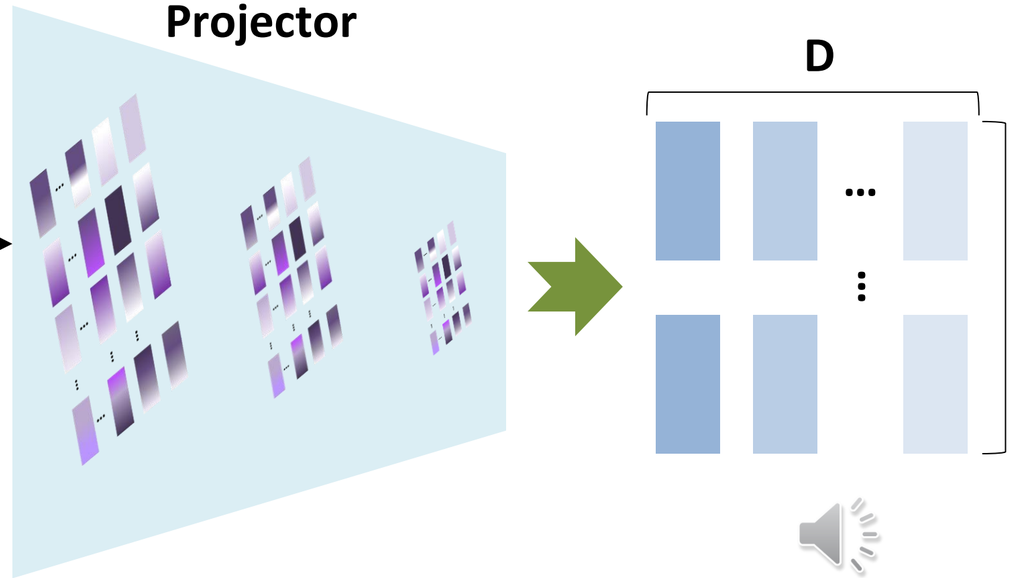
3D-VLA

- **Methodology : Injecting Goal Generation Ability into 3D-VLA (Projector)**
  - 행동 예측을 넘어 미래의 물리적 변화를 '상상'하는 Generative World Model 구현
  - 기존 범용 비디오 확산 모델의 한계(뷰 왜곡, 데이터 부족)를 극복하기 위해 로봇 데이터셋 기반의 전용 Diffusion Model 사전 학습
  - Interaction token & 3D-LLM 단계에서 확보한 모델의 상황 인지 답변(잠재 표현)으로 디퓨전 모델을 학습 [N×2048]
  - 단순한 텍스트 생성을 넘어, 미래의 RGB-D 이미지와 포인트 클라우드를 출력값으로 생성함으로써 계획(Planning) 능력 강화

3D-VLA answer :

"Sure! I should pick up the <obj> apple </obj> at <loc102>. I will first imagine the goal state.
<image> [Latent_Goal_Tokens] </image> <pcd> **[Latent_Pcd_Tokens]** </pcd>.
Now I will execute: <ACT_START> <aloc...>..."

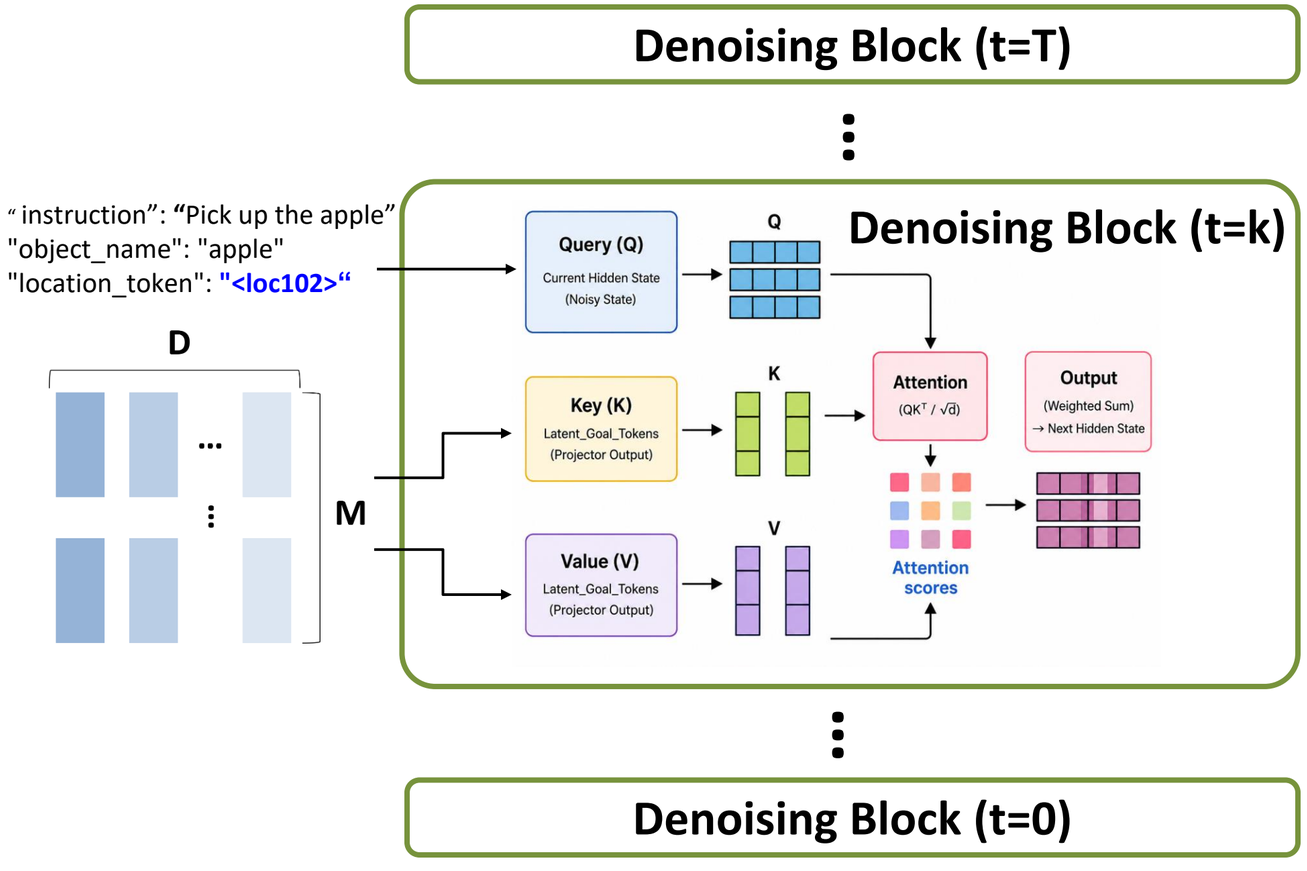




Methods (1)

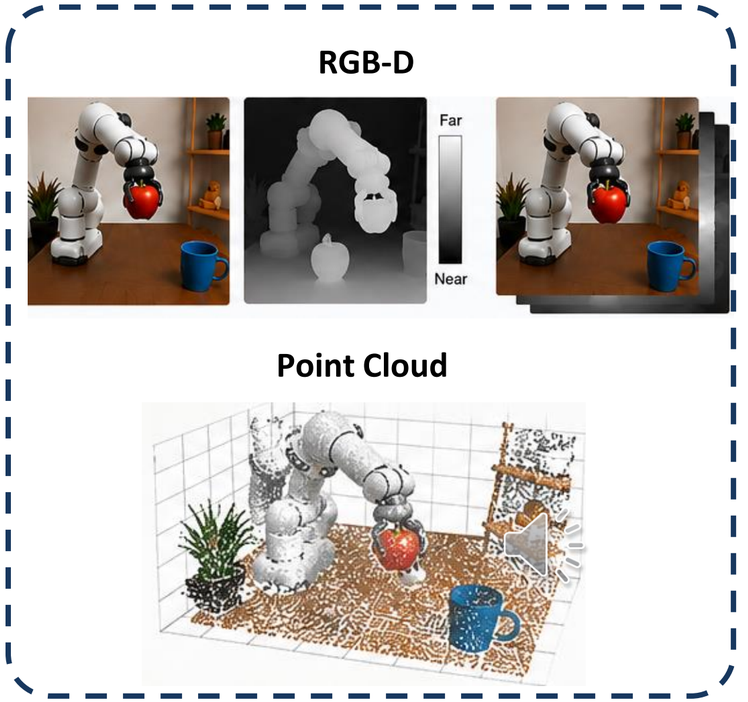
3D-VLA

- **Methodology : Injecting Goal Generation Ability into 3D-VLA (Diffusion model)**
  - LLM의 언어적 잠재 공간과 Diffusion Model의 생성 공간 사이의 의미적 간극 해소
  - Transformer 기반의 Projector를 설계하여 LLM에서 추론된 고수준의 목적(Goal) 특징을 생성 모델의 조건(Conditioning)으로 매핑
  - LoRA를 활용해 기존 가중치 보존 및 효율적인 정렬을 수행하여 파라미터 업데이트 최소화



VAE Decoder

Output : Goal state



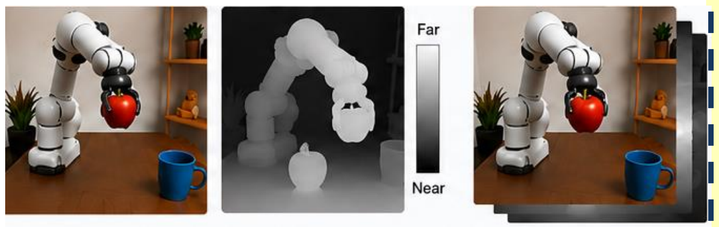
Methods (1)

3D-VLA

- **Methodology : Action Decoding (Robot Control)**
  - 3D 장면 이해와 목표 상상 결과를 바탕으로 실시간 로봇 제어 명령을 직접 생성
  - 복잡한 7 DoF 로봇 작업을 256개의 discrete 액션 토큰으로 변환하여 LLM이 텍스트 생성하듯 순차적으로 추론
  - 히스토리 데이터에 의존하지 않는 Open-loop 제어 방식을 통해, 새로운 환경에서도 언어 지시만으로 즉각적인 행동 수행 가능

Output : Goal state

RGB-D



Point Cloud

VIT

```
#Text embed : "The initial scene is.."
[ 0.012, -0.45, 0.88, ..., -0.03 ], # "The"
[ -0.23,  0.15, 0.04, ...,  0.92 ], # "initial"
[  0.05, -0.11, 0.22, ..., -0.55 ], # "scene"
[  0.10,  0.33, -0.1, ...,  0.12 ], # "is"

#3D embed<scene>[Initial_Scene_Embed]</scene>
[ 0.55, -0.72, 0.11, ..., 0.33 ],  # <Vector_V1>
[ 0.91,  0.02, -0.56, ..., -0.19 ], # <Vector_V2>
...
[ 0.44, -0.12,  0.05, ..., 0.77 ],  # <Vector_V32>

#Instruction embed : "Pick up the apple."
[ -0.08,  0.41, -0.22, ...,  0.15 ], # "Pick“
[  0.19, -0.33,  0.55, ..., -0.04 ], # "up“
[ -0.51,  0.08,  0.12, ...,  0.88 ], # "the“
[  0.66, -0.21, -0.19, ..., -0.31 ], # "apple“
[ -0.11,  0.05,  0.33, ...,  0.02 ], # "."

#Goal embed : <image>[Goal_Image_Embed]</image>
[ 0.88, -0.15, -0.42, ..., -0.22 ], # <Vector_G1>
[ 0.21,  0.66, -0.33, ...,  0.41 ], # <Vector_G2>
...
[ -0.1,  0.88,  0.14, ..., -0.09 ]  # <Vector_G32>
```

3D VLA(LLM) Model

Methods (1)

3D-VLA

- **Methodology : Action Decoding (Robot Control)**
  - 3D 장면 이해와 목표 상상 결과를 바탕으로 실시간 로봇 제어 명령을 직접 생성
  - 복잡한 7 DoF 로봇 작업을 256개의 discrete 액션 토큰으로 변환하여 LLM이 텍스트 생성하듯 순차적으로 추론
  - 히스토리 데이터에 의존하지 않는 Open-loop 제어 방식을 통해, 새로운 환경에서도 언어 지시만으로 즉각적인 행동 수행 가능

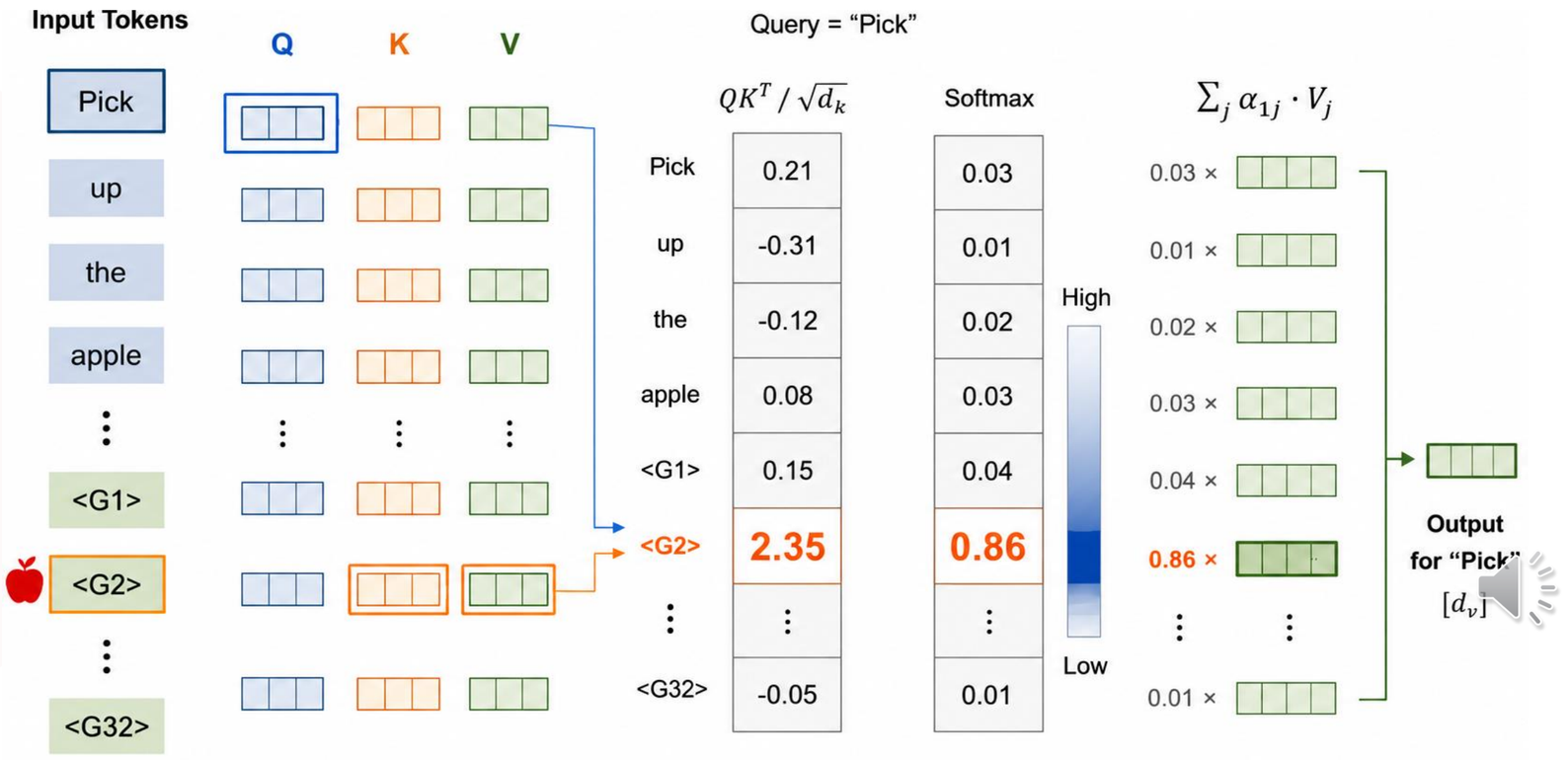
```
#Text embed : "The initial scene is.."
[ 0.012, -0.45, 0.88, ..., -0.03 ], # "The"
[ -0.23,  0.15, 0.04, ...,  0.92 ], # "initial"
[  0.05, -0.11, 0.22, ..., -0.55 ], # "scene"
[  0.10,  0.33, -0.1, ...,  0.12 ], # "is"

#3D embed<scene>[Initial_Scene_Embed]</scene>
[ 0.55, -0.72, 0.11, ..., 0.33 ],  # <Vector_V1>
[ 0.91,  0.02, -0.56, ..., -0.19 ], # <Vector_V2>
...
[ 0.44, -0.12,  0.05, ..., 0.77 ],  # <Vector_V32>

#Instruction embed : "Pick up the apple."
[ -0.08,  0.41, -0.22, ...,  0.15 ], # "Pick“
[  0.19, -0.33,  0.55, ..., -0.04 ], # "up“
[ -0.51,  0.08,  0.12, ...,  0.88 ], # "the“
[  0.66, -0.21, -0.19, ..., -0.31 ], # "apple“
[ -0.11,  0.05,  0.33, ...,  0.02 ], # "."

#Goal embed : <image>[Goal_Image_Embed]</image>
[ 0.88, -0.15, -0.42, ..., -0.22 ], # <Vector_G1>
[ 0.21,  0.66, -0.33, ...,  0.41 ], # <Vector_G2>
...
[ -0.1,  0.88,  0.14, ..., -0.09 ]  # <Vector_G32>
```

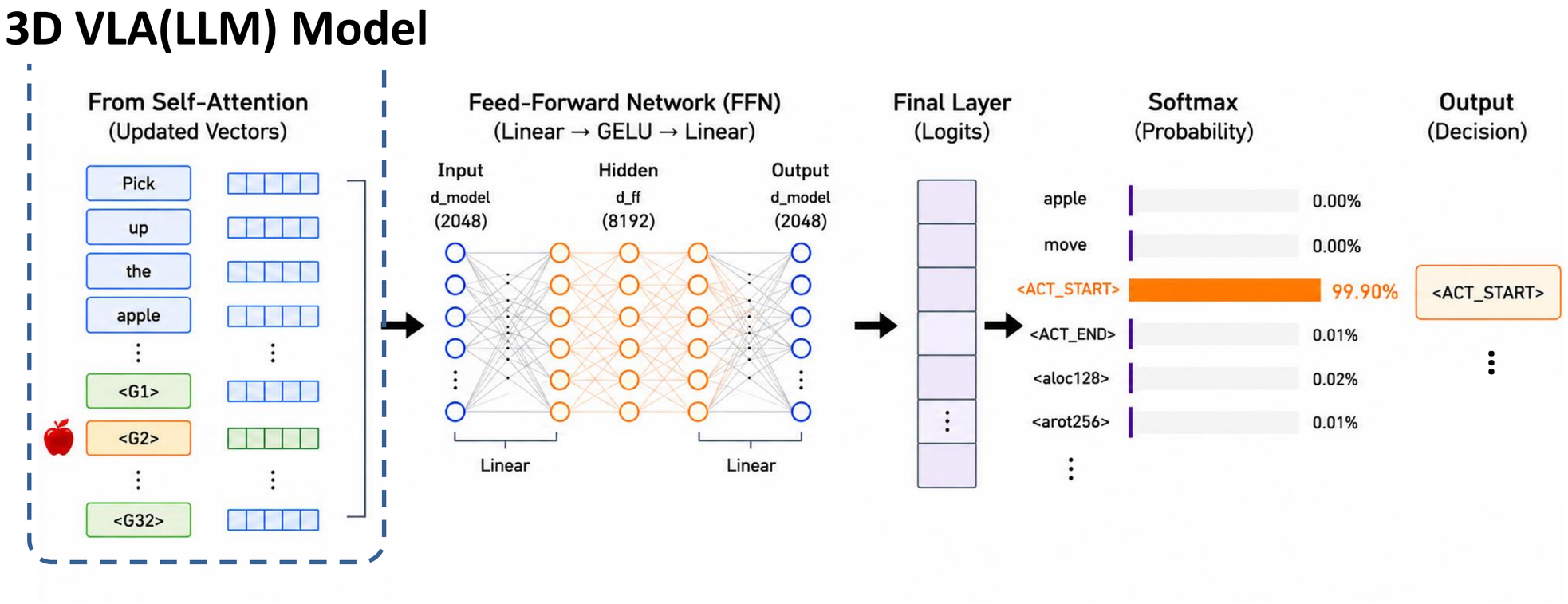
3D VLA(LLM) Model



Methods (1)

3D-VLA

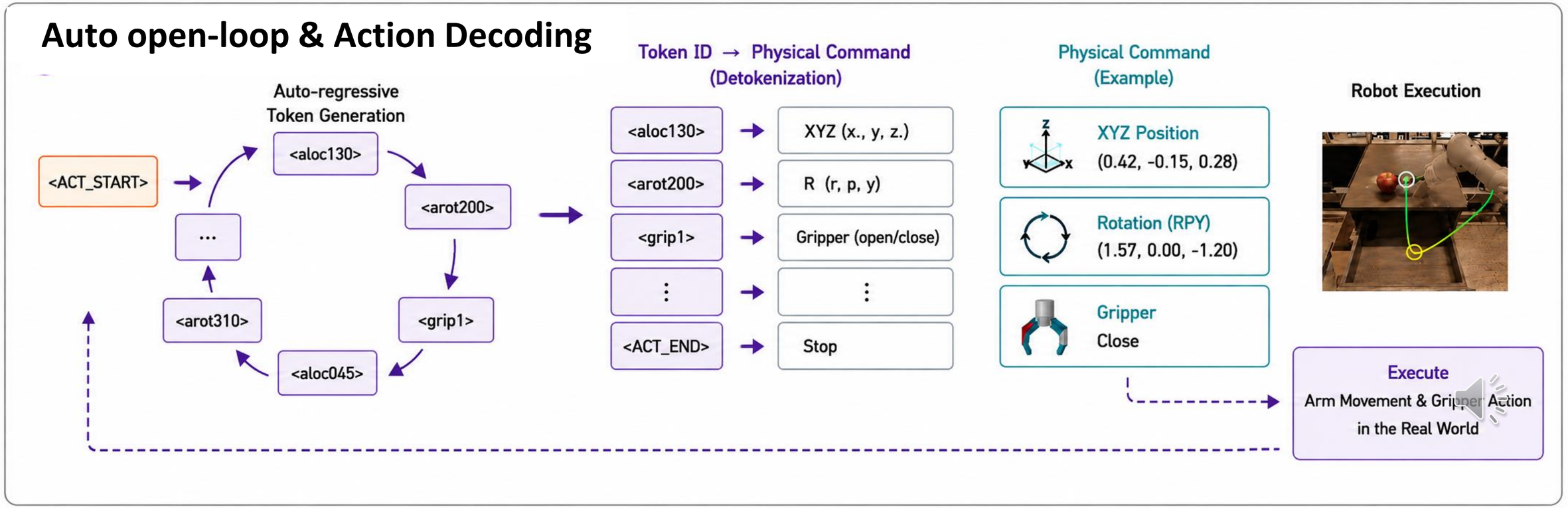
- **Methodology : Action Decoding (Robot Control)**
  - 3D 장면 이해와 목표 상상 결과를 바탕으로 실시간 로봇 제어 명령을 직접 생성
  - 복잡한 7 DoF 로봇 작업을 256개의 discrete 액션 토큰으로 변환하여 LLM이 텍스트 생성하듯 순차적으로 추론
  - 히스토리 데이터에 의존하지 않는 Open-loop 제어 방식을 통해, 새로운 환경에서도 언어 지시만으로 즉각적인 행동 수행 가능



Methods (1)

3D-VLA

- **Methodology : Action Decoding (Robot Control)**
  - 3D 장면 이해와 목표 상상 결과를 바탕으로 실시간 로봇 제어 명령을 직접 생성
  - 복잡한 7 DoF 로봇 작업을 256개의 discrete 액션 토큰으로 변환하여 LLM이 텍스트 생성하듯 순차적으로 추론
  - 히스토리 데이터에 의존하지 않는 Open-loop 제어 방식을 통해, 새로운 환경에서도 언어 지시만으로 즉각적인 행동 수행 가능



Methods (1)

3D-VLA

- **Experiments : 3D 추론과 물체 위치 찾기**
  - 로봇 상황 QA · 행동 설명 · 물체 위치 찾기 등 다섯 과제로 평가
  - 행동 설명(Task Caption) 점수 34.88로 같은 조건의 BLIP2(3.16)를 크게 앞섬
  - 물체 위치 찾기 정확도(IoU) 29.33으로 CoVLM(19.81) · Kosmos-2(10.92)보다 우수
  - 3D-LLM이 낮은 점수를 낸 것은 로봇 전용 3D 데이터가 따로 필요함을 보여줌

Tasks	Models	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGH-L	EM@1
Embodied QA	3D-LLM*	1.05	0.38	0.15	0.02	12.96	0.91	0.00
	BLIP2 $OPT_{2.7B}$*	7.39	3.17	0.03	0.02	3.87	7.40	3.03
	BLIP2 $FlanT5_{XL}$*	22.84	16.17	12.50	10.11	11.41	32.01	10.31
	$OpenFlamingo_{4B}$*	9.50	6.51	5.14	4.29	6.84	10.40	1.21
	$LLaVA_{7B}$*	11.66	8.06	6.01	4.58	12.59	14.17	5.67
	BLIP2 $FlanT5_{XL}$	37.31	27.20	20.32	15.48	17.80	38.92	15.35
	3D-VLA	**48.34**	**38.55**	**31.72**	**26.80**	**23.72**	**49.33**	**24.53**
Task Caption	3D-LLM*	0.78	0.16	0.07	0.05	0.57	1.33	0.00
	BLIP2 $FlanT5_{XL}$*	8.50	2.07	0.35	0.00	3.40	8.45	0.00
	$OpenFlamingo_{4B}$*	7.61	1.64	0.37	0.00	4.74	9.36	0.00
	$LLaVA_{7B}$*	2.63	0.69	0.16	0.00	2.63	4.65	0.00
	BLIP2 $FlanT5_{XL}$	22.05	11.40	5.72	3.16	8.72	26.12	7.75
	3D-VLA	**55.69**	**45.88**	**39.39**	**34.88**	**27.57**	**62.01**	**29.34**
What-if QA	BLIP2 $FlanT5_{XL}$	28.23	11.47	4.49	0.06	8.27	28.41	5.85
	3D-VLA	**53.09**	**40.94**	**34.34**	**29.38**	**26.83**	**52.82**	**14.7**
Dense Caption	3D-LLM*	0.52	0.22	0.16	0.13	0.34	0.64	0.00
	BLIP2 $FlanT5_{XL}$	36.17	24.72	18.06	13.96	17.83	40.56	13.10
	3D-VLA	**51.90**	**42.83**	**38.11**	**34.62**	**25.25**	**55.91**	**39.49**

Table 1. Evaluation on reasoning ability using held-in data. ∗ denotes zero-shot transfer results without training on our pre-train datasets.

Methods	IoU	Acc@25	Acc@50
Kosmos-2 (w/ GT Depth)	10.92	12.73	3.85
CoVLM (w/ GT Depth)	19.81	25.39	16.61
3D-VLA	**29.33**	**42.26**	**27.09**

Table 2. Localization results on held-in robotics datasets.

Methods (1)

3D-VLA

- **Experiments : 목표 생성과 행동 계획**
  - 학습에 쓰지 않은 4000개 영상으로 목표 이미지 생성 품질을 평가
  - 이미지 품질(PSNR) 17.21로, 같은 로봇 데이터로 학습한 Instruct-P2P(16.67)보다도 높음
  - 예측한 물체 위치 정보를 빼면 점수가 떨어져, 중간 단서가 실제로 도움이 됨을 확인
  - RLBench 조작은 앞서지만 CALVIN에서 5개 명령 연속 수행은 0으로, 긴 작업에는 한계

Method	PSNR ↑	CLIP Sim ↑	SSIM ↑	FID ↓
Instruct-P2P	14.41	0.909	0.389	0.309
SuSIE	15.20	0.898	0.549	0.182
NeXT-GPT	8.86	0.199	0.153	0.432
Instruct-P2P*	16.67	**0.941**	0.628	0.178
3D-VLA w/o Pred BBox	17.02	0.919	0.632	**0.173**
3D-VLA	**17.21**	0.920	**0.636**	0.177

Table 3. RGB image goal generation results. ∗ denotes the model is trained on our pretrained dataset.

	Put Knife	Take Umbrella	Pick up Cup	Pick up Cup (unseen)
LanCon-Learn	28.8	45.6	23.2	-
LanCon-Learn w/ His.	32.2	50.8	**44.2**	-
3D-VLA	**68**	**52**	40	24

Table 5. Evaluation of action planning on RLBench dataset.

Methods (1)

3D-VLA

- **Experiments : 생성된 목표 장면 정성적 평가**
  - 배경은 그대로 두고 지시받은 물체만 바꾸어 목표 장면을 그려냄
  - 생성된 결과가 정답 장면과 겉모습 · 의미 모두에서 비슷하게 맞음
  - 학습에 없던 인터넷 · 일상 사진에서도 안정적으로 동작
  - RLBench에서는 목표 이미지와 3D 포인트클라우드를 함께 생성

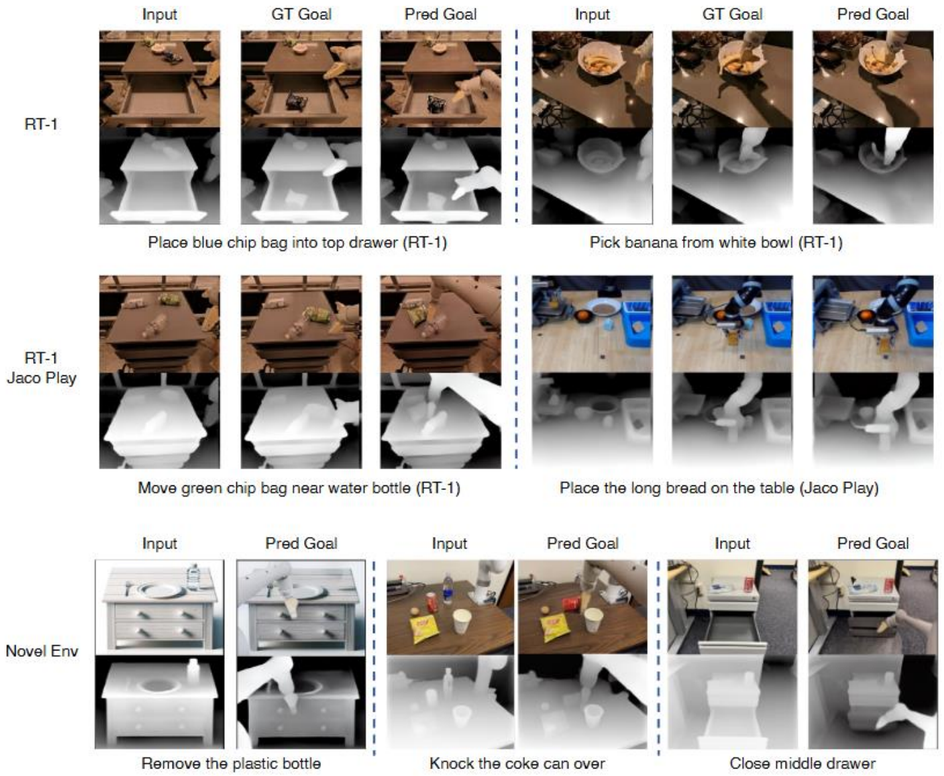


Figure 3. Visualization of generated RGB-D goal images. The results in the first row are sampled from the test set of held-in training data while the second row is the unseen environments gathered from the Internet or daily life.

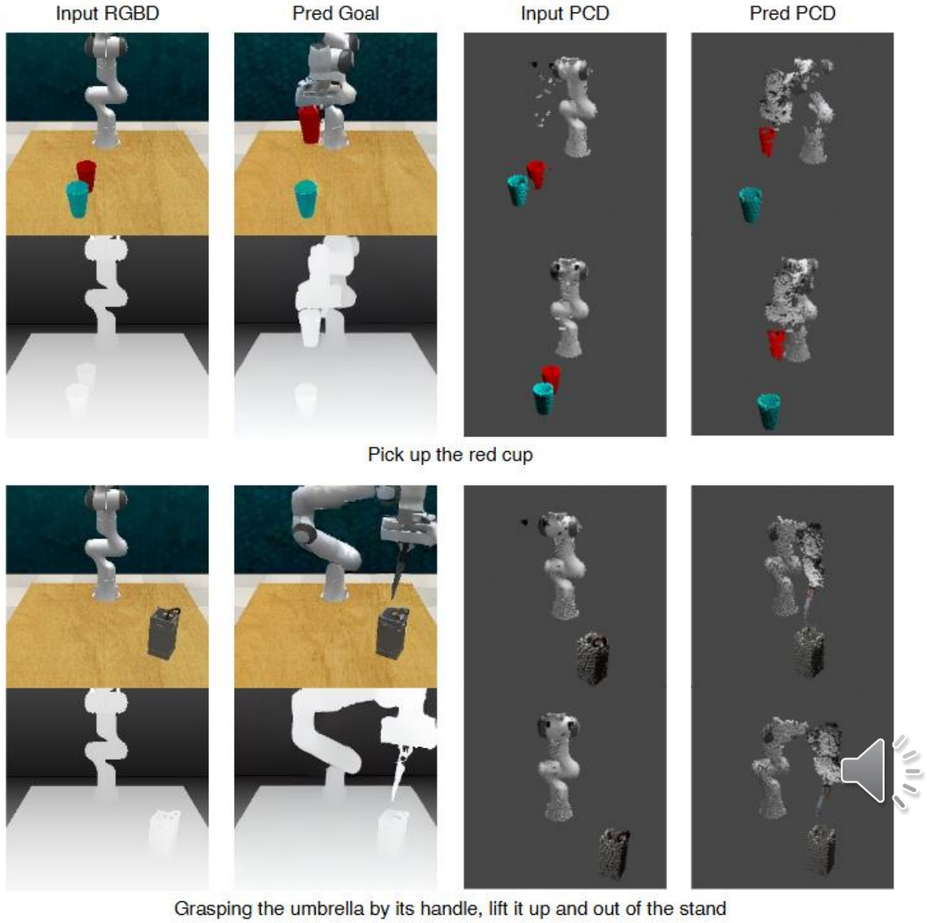
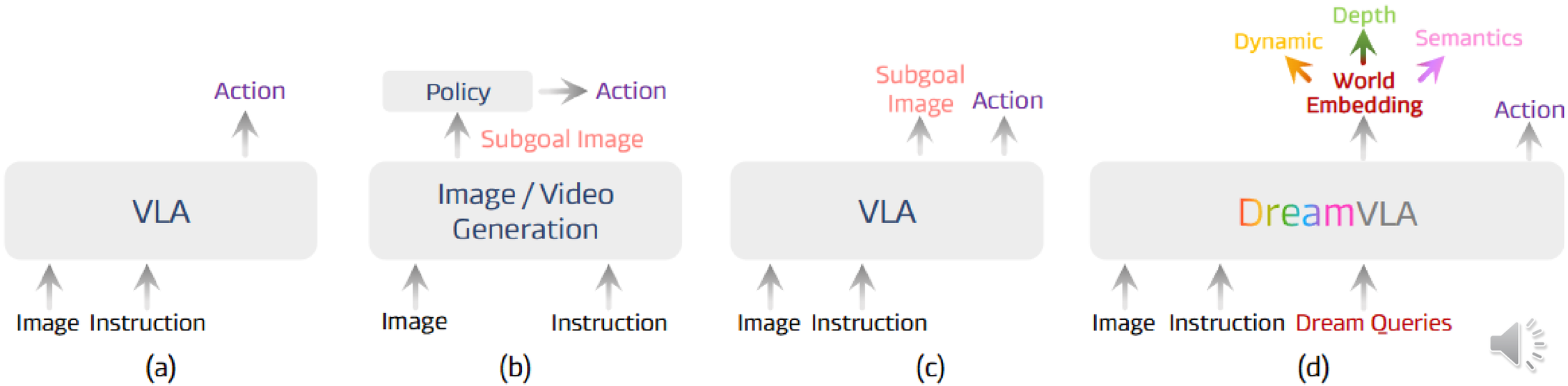


Figure 6. Visualization of generated RGB-D goal images and goal point cloud. (RLBench)

Methods (2)

DreamVLA

- **DreamVLA : A VLA Dreamed with Comprehensive World Knowledge (NeurIPS 2025)**
  - (a) 기본 VLA — 화면과 지시를 보고 곧바로 행동을 냄
  - (b) 별도의 영상 생성 모델이 미래 장면을 만들어주면, 그것을 보고 행동을 냄
  - (c) 행동에 앞서 중간 목표 이미지를 한 장 예측하는 방식
  - **(d) DreamVLA — 미래 장면을 통째로 그리지 않고 움직임 · 깊이 · 의미만 골라서 예측**



Zhang, W., Liu, H., Qi, Z., Wang, Y., Yu, X., Zhang, J., ... & Jin, X. (2026). Dreamvla: a vision-language-action model dreamed with comprehensive world knowledge. *Advances in Neural Information Processing Systems*, *38*, 24195-24228.

Methods (2)

DreamVLA

- **Motivation**
  - (1) 중복 픽셀 정보 — 예측 이미지와 현재 관측이 크게 겹쳐 예측이 비효율적[1)]
  - (2) 공간 정보 부재 — 환경에 대한 명시적 3D 지식이 없음[2)]
  - (3) 고수준 지식 예측 부재 — 시맨틱 등 미래 상태의 고수준 이해가 누락[3)]
  - **DreamVLA는 세가지 한계에 각각 대응하는 세 종류의 world knowledge를 예측하도록 설계**

① Redundant pixel information

프레임 대부분이 현재 관측과 중복 – 학습이 비효율적

② Lake of spatial information

테스크 환경의 3차원 구조에 대한 고차원 지식 부재

③ Lake of high-level forecasting

특정 물체에 집중하기 위한 semantic 정보가 누락

기존 VLA

1 Dynamic region-based forcasting

전체 프레임 대신 CoTracker로 동역학 정보를 예측

2 Depth-aware forecasting

프레임별 depth 특징을 예측해 3차원 공간 맥락 정보를 확보

3 High-level foundation features

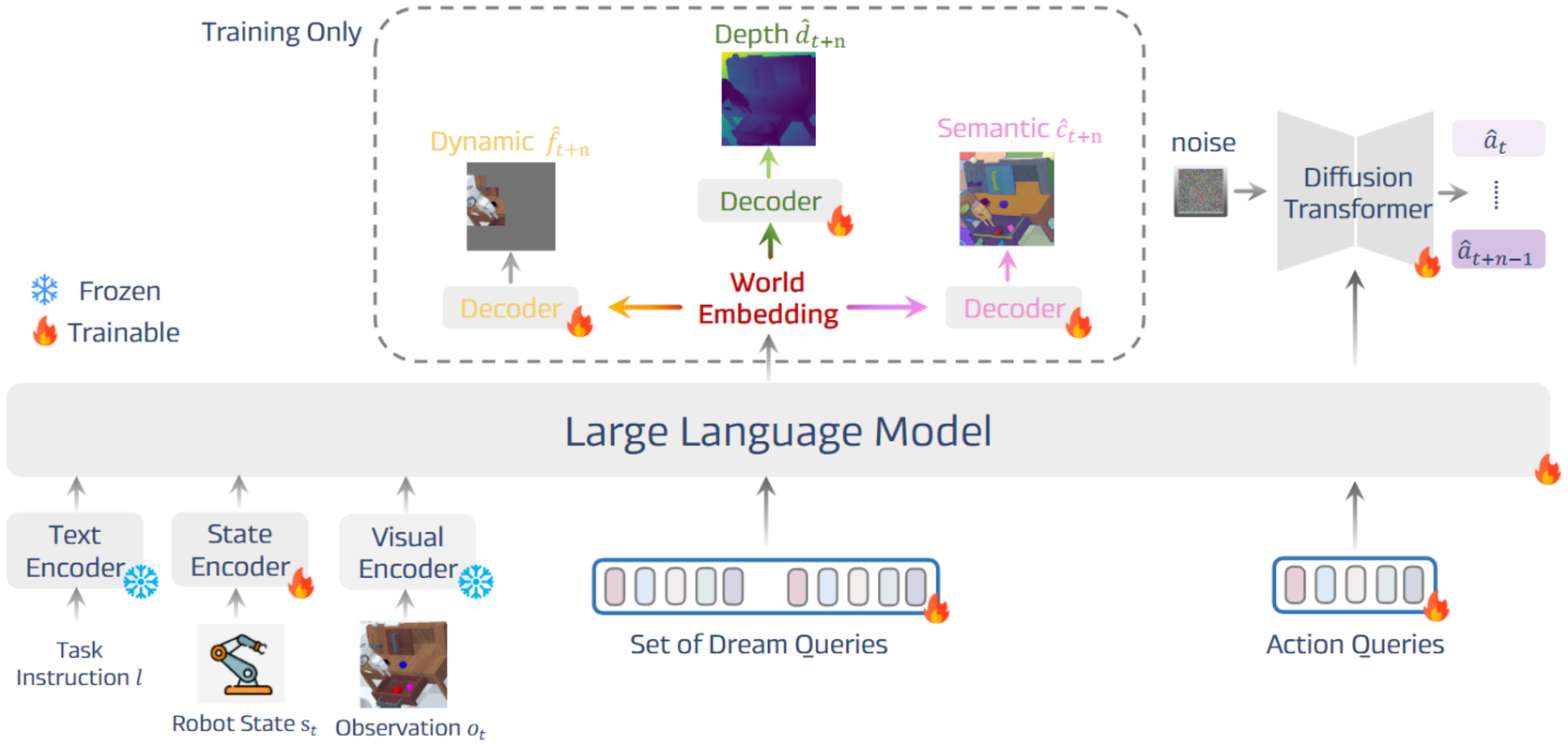
DINOv2, SAM과 정렬된 미래 semantic 정보를 예측 및 학습

DreamVLA

Methods (2)

DreamVLA

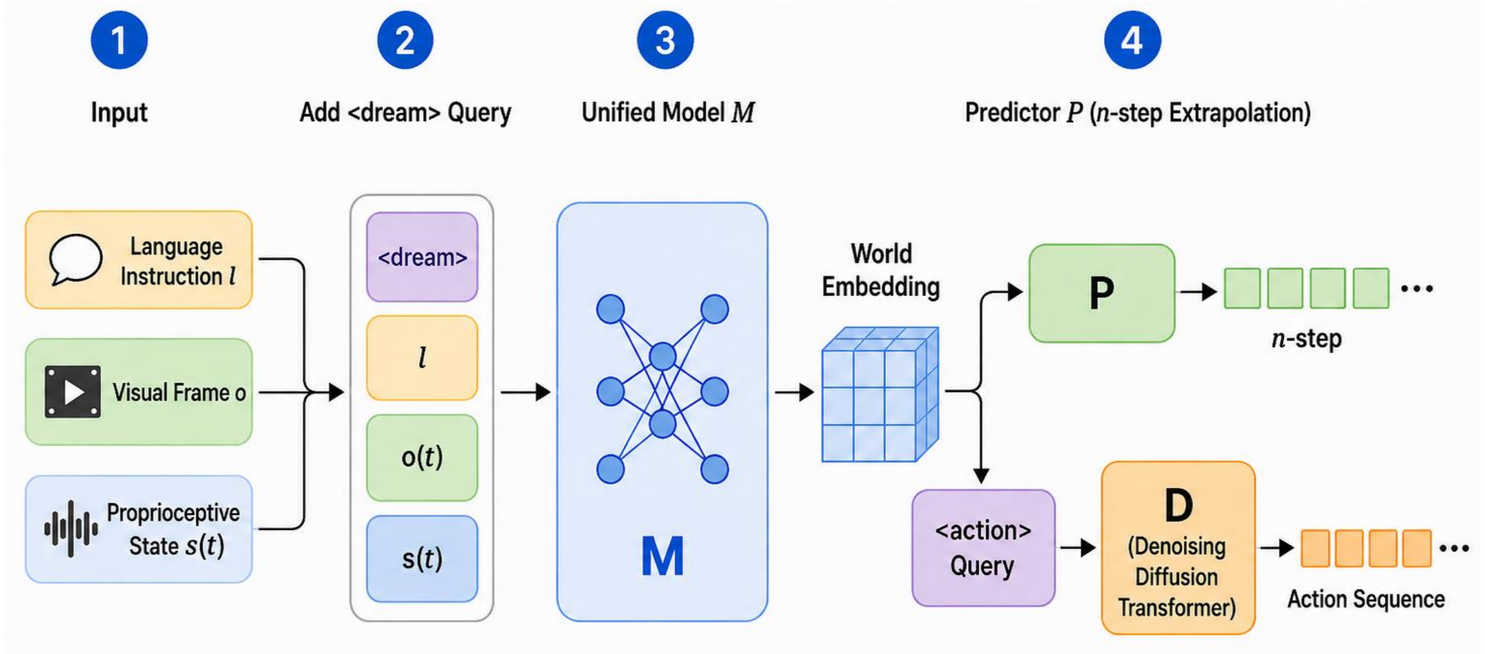
- **Model Architecture & Pipeline**
  - 언어는 CLIP 텍스트 임베딩, 시각은 MAE 사전학습 ViT-B, 고유수용 상태는 conv + FC로 인코딩
  - 학습 가능한 **<dream>**(동적 · 깊이 · 시맨틱 3개 쿼리)과 **<action>** 쿼리를 임베딩 뒤에 추가
  - GPT-2 기반 LLM이 구조화된 causal / non-causal attention으로 모달리티와 쿼리를 통합
  - 경량 출력 헤드가 world embedding을 복원하지만, 추론 시에는 디코더를 건너뛰어 연산을 절약



Methods (2)

DreamVLA

- **Problem Definition and Notation**
  - VLA 추론을 역동역학(inverse dynamics) 문제로 정식화 — 역동역학이란 현재 상태 <-> 미래 상태 사이의 행동을 거꾸로 푸는 문제
    - 「나중 장면」 자리에 **아직 오지 않은 미래를 스스로 예측해 넣음**
  - 매 시점 t에 언어 지시 l, 시각 프레임 o_t, 고유수용 상태 s_t 의 세 신호를 입력받음
  - ①~③ 위 3개의 신호에 <dream> 쿼리를 붙이면, 통합 모델 $M(\cdot)$ 이 미래 정보를 압축한 world embeding을 산출
  - ④ 예측기 $P(\cdot)$ 가 n스텝 뒤의 움직임 ·깊이 · 의미를 뽑고(Extrapolation), <action> 쿼리와 디퓨전 트랜스포머 $\mathrm{D}(\cdot)$ 가 행동 시퀀스를 생성



World Embedding

$$w_{t+n} = M(l, o_t, s_t | \langle \text{dream} \rangle)$$

Knowledge Projection

$$\hat{p}_{t+n} = P(w_{t+n}) = (\hat{f}_{t+n}, \hat{d}_{t+n}, \hat{c}_{t+n})$$

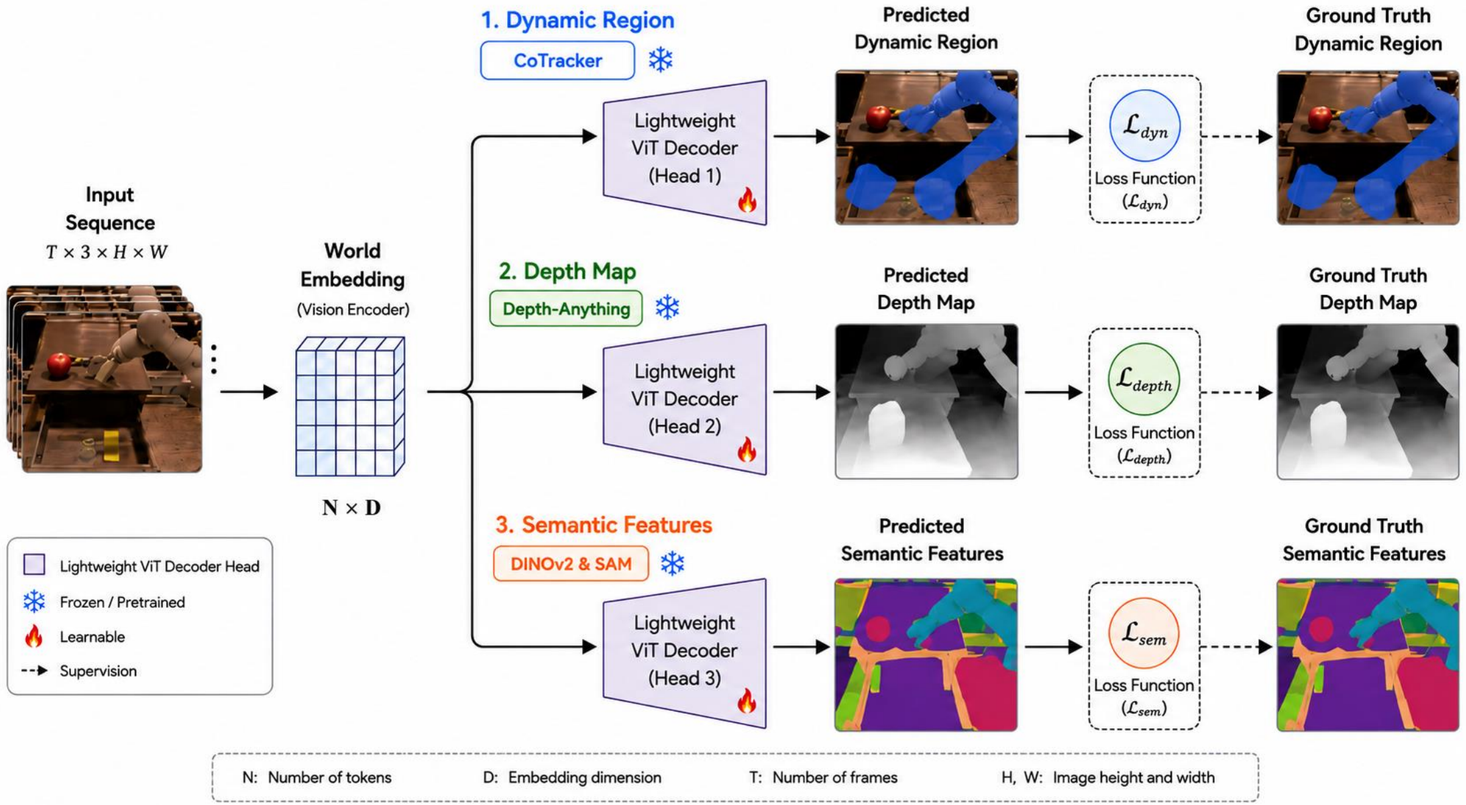
Action Generation(Decoding)

$$\hat{a}_{t:t+n-1} = D\big(M(l, o_t, s_t, \langle \text{dream} \rangle | \langle \text{action} \rangle)\big)$$

Methods (2)

DreamVLA

- **Comprehensive Knowledge Forecasting : Dynamic · Depth · Semantics**
  - Dynamic : CoTracker 마스크로 움직이는 영역만 표시해 그 부분에 집중(ROI 계산)
  - Depth : Depth-Anything의 예측을 정담으로 삼아 멀고 가까움의 상대적 깊이를 학습
  - Semantic : DINOv2 · SAM 특징으로 각 물체의 중요도를 인지
  - 위 3개의 모달에 대해 경량 ViT 디코더가 각각의 손실로 학습



Dynamic Loss

$$L_{\text{dyn}} = \frac{1}{|D|} \sum_{x_i \in D} \mathbb{E}_{z \sim Q_\phi(z|x_i)} \left[-\log P_\psi (x_i) \odot \text{Mask} | z_i \right]$$

Depth Loss

$$L_{\text{depth}} = \frac{1}{HW} \sum_{i,j} \left\| \hat{d}_{i,j,t+n} - \alpha d_{i,j,t+n} \right\|^2$$

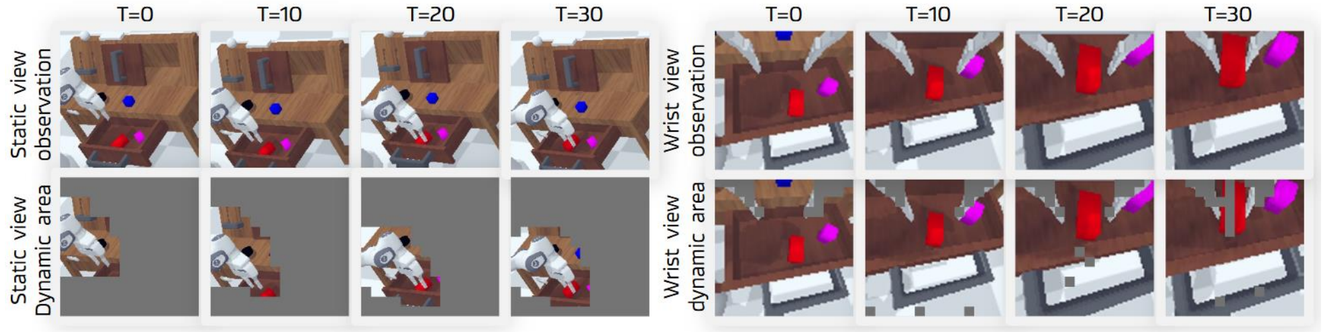
Semantic Loss

$$L_{\text{sem}} = -\log \frac{\exp(\hat{c}_{t+n}^{\top} c_{t+n} / \tau)}{\sum_k \exp(\hat{c}_{t+n}^{\top} c_k / \tau)}$$

Methods (2)

DreamVLA

- **Comprehensive World Knowledge (1) : Dynamic Region**
  - 화면 전체나 픽셀 흐름 전부가 아니라 「곧 움직일 곳」만 골라서 학습
  - CoTracker로 로봇 팔 · 물체와 함께 움직이는 픽셀을 찾아 그 영역만 복원
  - 가려진 부분을 되살리는 문제로 두고, 그 확률을 최대로 만드는 방식으로 정식화
  - 그 결과 손실은 「가려진 부분을 얼마나 잘 맞혔는가」로 단순해짐



Generative ELBO

$$\log P\left(x_i|\tilde{x}_i\right) \geq \mathbb{E}_{z_i \sim Q_\phi(z|x_i)}\left[\log P_\psi\left(x_i|z_i\right)\right] - D_{KL}[z, P_\theta(z|\hat{z}_i)]$$

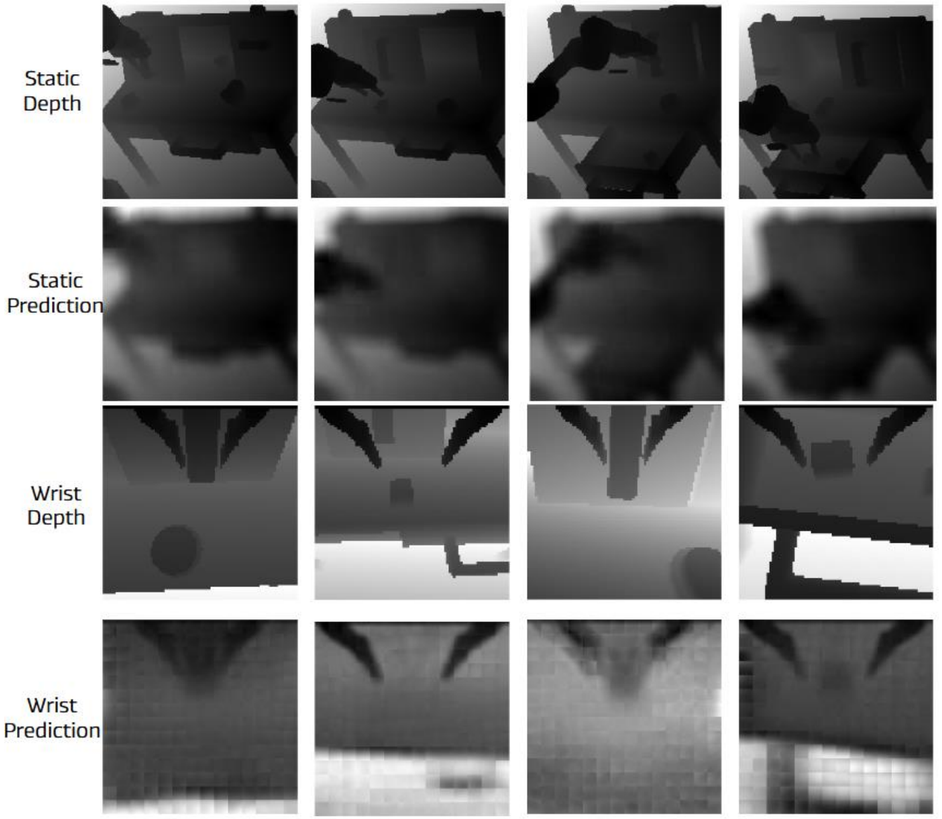
Dynamic Loss

$$L_{\text{dyn}} = \frac{1}{|D|}\sum_{x_i \in D} \mathbb{E}_{z \sim Q_\phi(z|x_i)}\left[-\log P_\psi\left(x_i\right) \odot \text{Mask}|z_i\right]$$

Methods (2)

DreamVLA

- **Comprehensive World Knowledge (2) : Depth**
  - 깊이가 어떻게 변할지 알면 어디로 가야 하고 무엇을 피해야 하는지 알 수 있음
  - 깊이 센서가 있으면 실측값을, 없으면 Depth-Anything의 예측을 정답 대신 사용
  - 카메라마다 거리 기준이 달라지는 문제를 계수 α로 맞춰준 뒤 오차를 계산
  - 정확한 거리보다 앞뒤 순서를 지키는 것이 목표 (집기 · 충돌 판단에 필요)



Depth Loss

$$L_{\text{depth}} = \frac{1}{HW}\sum_{i,j}\left(\hat{d}(i,j)_{t+n} - \alpha d(i,j)_{t+n}\right)^2$$

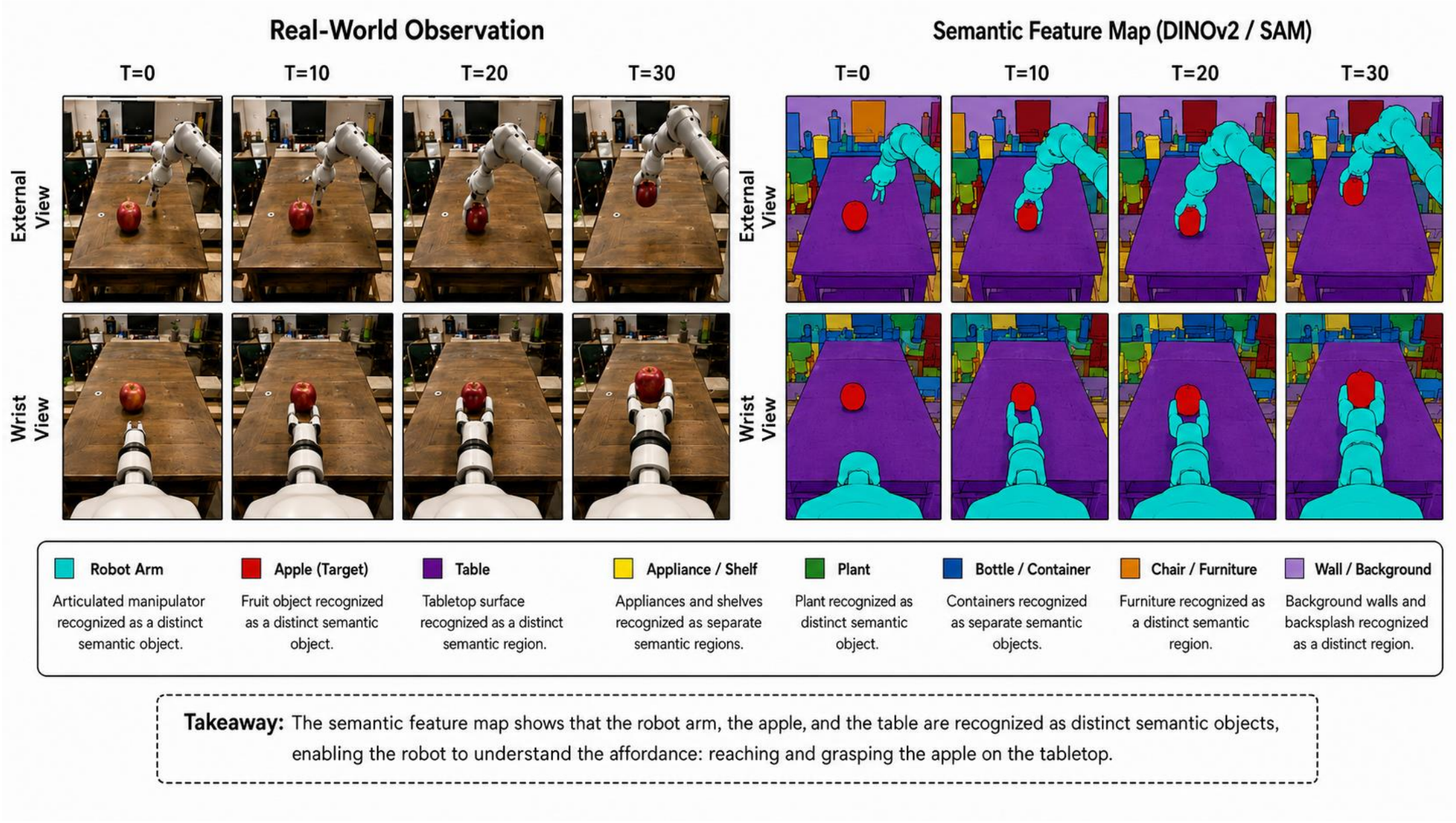
Scale Norm

$$\alpha = \frac{\sum_{i,j}\hat{d}\,(i,j)_{t+n}\cdot d(i,j)_{t+n}}{\sum_{i,j} d\,(i,j)^2_{t+n}}$$

Methods (2)

DreamVLA

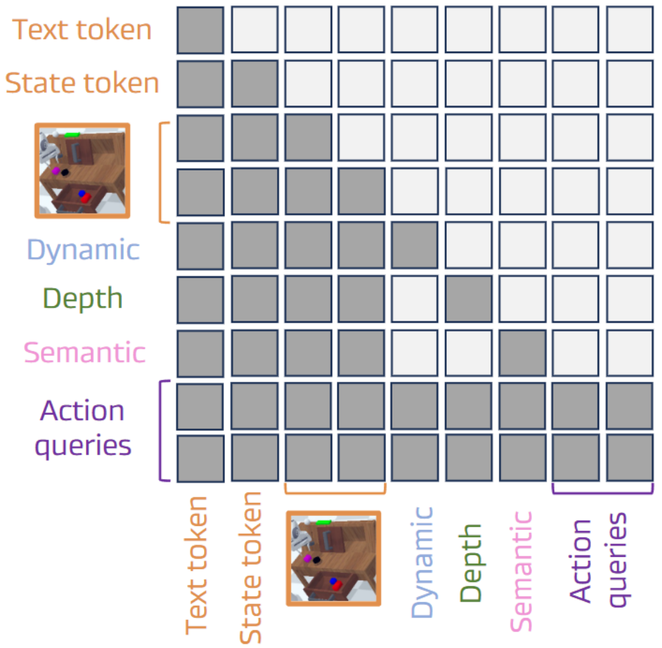
- **Comprehensive World Knowledge (3) : Semantic**
  - 아래 그림 중, 우측은 좌측 원본과 달리 로봇 팔, 사과 등의 각 객체가 서로 다른 색으로 분리
  - DINOv2는 물체의 종류를, SAM은 물체의 경계를 알려주고, 이 둘을 합쳐 미래 시점의 의미를 예측
  - 여러 타임 스텝에 따라 로봇 팔이 움직여도, 각 물체의 색은 유지되어, 객체의 고유 식별 정보를 계속 추적
  - 즉,「테이블 위 사과를 향해 뻗어 집는다」같은 지시어에서 각 객체의 중요도를 식별하고 객체 추적 가능



Methods (2)

DreamVLA

- **Block-wise Structured Attention**
  - <dream>은 동적 · 깊이 · 시맨틱 세 서브쿼리로 분해됨
  - 서로 자유롭게 attend하면 고주파 flow가 깊이 추론을 오염시키고 시맨틱이 모션에 스며듦
  - 각 서브쿼리는 공유된 시각 · 언어 · 상태 토큰만 참조하고 세 쿼리 간 직접 연결은 마스킹
  - <dream>과 <action> 모두 과거로 제한된 causal attention — MoE의 specialist routing과 유사



마스킹 규칙

- 각 서브쿼리는 공유된 시각 · 언어 · 상태 토큰만 참조
- 세 서브쿼리(dynamic · depth · semantic) 사이의 직접 연결은 차단
- <dream> 과 <action> 모두 과거 컨텍스트로 제한된 causal attention

※ 공유 토큰 열은 모두 열려 있으므로 block-diagonal 구조가 아님

설계 효과

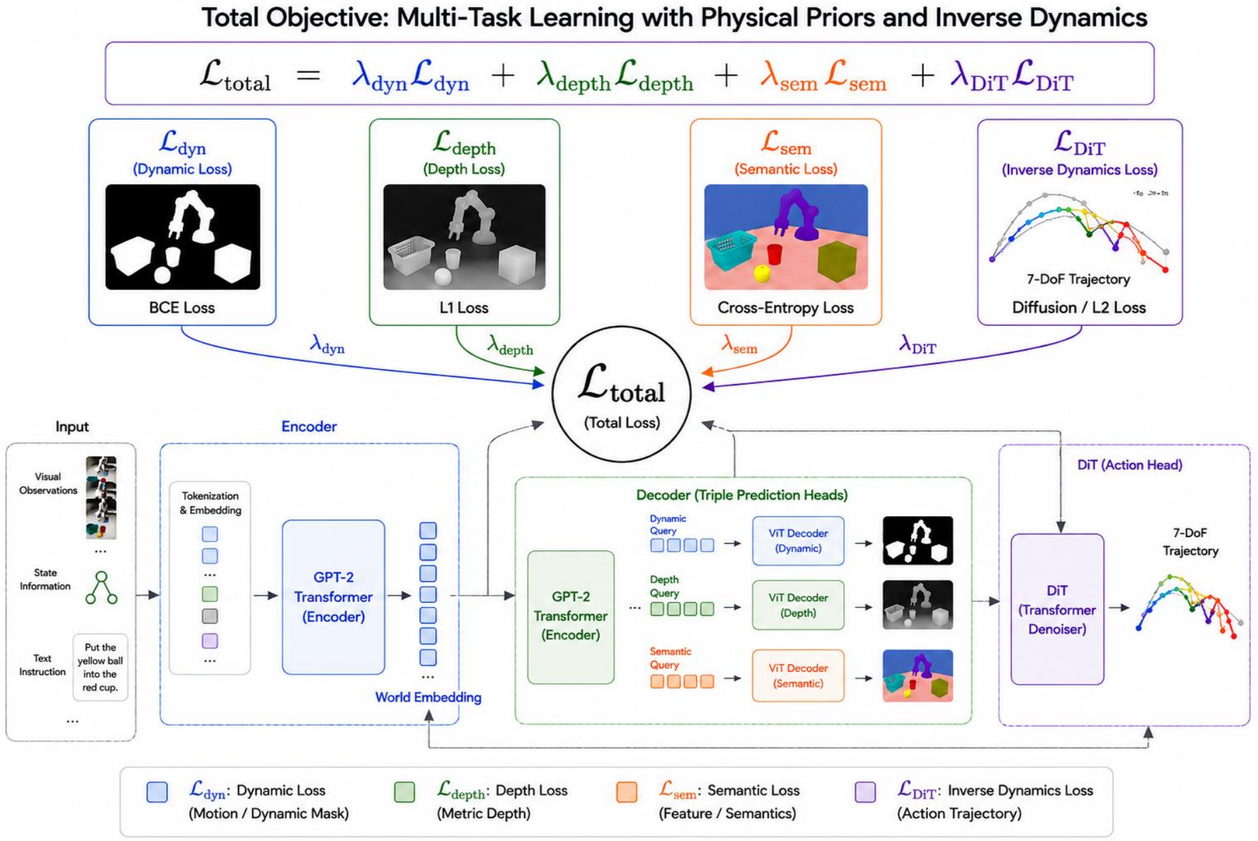
- ✓ 깨끗한 미래 world knowledge 를 행동 예측에 공급
- ✓ 표현 잡음이 줄어 강건성 향상
- ✓ 시간적 일관성 유지

Mixture-of-Experts 의 specialist routing 과 유사한 구조

Methods (2)

DreamVLA

- **Inverse Dynamics via Denoising Diffusion Transformer**
  - 고전적 역동역학은 두 관측 사이의 행동을 추론 → 본 논문은 n스텝 행동 시퀀스로 확장
  - world 임베딩과 action 임베딩이 같은 잠재 공간·유사 통계를 공유 → 미래 지식과 행동 정보가 유사한 형태 → 단순 MLP 헤드로는 분리 불가
  - 따라서 DiT를 행동 헤드로 사용해 반복 self-attention과 디노이징으로 가우시안 노이즈를 행동 궤적으로 변환
  - 전체 손실은 네 항의 가중합 — 예측 손실 4개 / 행동 손실 1개



DiT Denoising

$$L_{\text{DiT}} = \mathbb{E}_{\tau,\epsilon}\left\|\epsilon - \epsilon_\theta\left(\sqrt{\bar{\alpha}_\tau}a_{t:t+n-1} + \sqrt{1-\bar{\alpha}_\tau}\epsilon, \tau, c\right)\right\|^2$$

Total Loss

$$L = \lambda_{\text{dyn}}L_{\text{dyn}} + \lambda_{\text{depth}}L_{\text{depth}} + \lambda_{\text{sem}}L_{\text{sem}} + \lambda_{\text{DiT}}L_{\text{DiT}}$$

Methods (2)

DreamVLA

- **Experiments (1) : 실험 설정 및 CALVIN ABC-D**
  - 8× A800에서 학습, 갈래마다 쿼리 9개 · 확산 10스텝 · 20 epoch
  - CALVIN과 DROID로 먼저 넓게 학습한 뒤, 목표 데이터로 미세조정
  - A · B · C 환경에서 배우고 한 번도 못 본 D 환경에서 평가 (명령 5개 연속 수행)
  - 평균 4.44개를 연속 수행해 VPP(4.29) · Seer(4.28) · RoboVLM(4.25)을 모두 앞섬

Table 1: **CALVIN ABC-D results.** We present the average success computed over 1000 rollouts for each task and the average number of completed tasks to solve 5 instructions consecutively (Avg. Len.). DreamVLA shows significant superiority over baselines. The best results are **bolded**.

Method	Task completed in a row					
	1	2	3	4	5	Avg. Len. ↑
Roboflamingo [32]	82.4	61.9	46.6	33.1	23.5	2.47
Susie [120]	87.0	69.0	49.0	38.0	26.0	2.69
GR-1 [16]	85.4	71.2	59.6	49.7	40.1	3.06
3D Diffusor Actor [95]	92.2	78.7	63.9	51.2	41.2	3.27
OpenVLA [1]	91.3	77.8	62.0	52.1	43.5	3.27
RoboDual [121]	94.4	82.7	72.1	62.4	54.4	3.66
UNIVLA [122]	95.5	85.8	75.4	66.9	56.5	3.80
π_0 [34]	93.8	85.0	76.7	68.1	59.9	3.84
CLOVER [123]	96.0	83.5	70.8	57.5	45.4	3.53
UP-VLA [59]	92.8	86.5	81.5	76.9	69.9	4.08
Robovlm [39]	98.0	93.6	85.4	77.8	70.4	4.25
Seer [58]	96.3	91.6	86.1	80.3	74.0	4.28
VPP [51]	95.7	91.2	86.3	81.0	75.0	4.29
DreamVLA	**98.2**	**94.6**	**89.5**	**83.4**	**78.1**	**4.44**

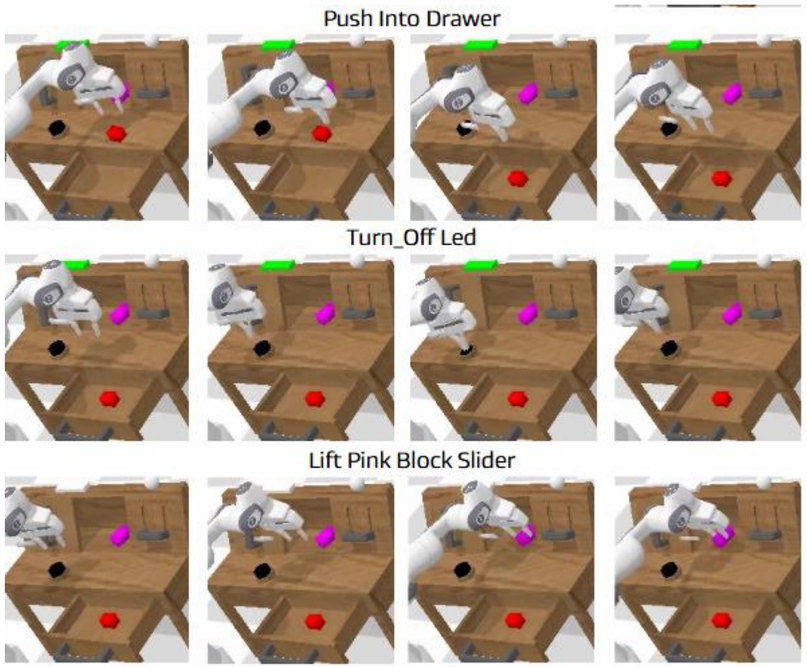


Figure 7: **Qualitative results** of the CALVIN long horizon task.

Methods (2)

DreamVLA

- **Experiments (2) : LIBERO 벤치마크**
  - LIBERO-Spatial / Object / Goal / Long 네 트랙, 각 10개 태스크와 50개 인간 원격조작 시연
  - 공간 추론 · 물체 중심 조작 · 목표 완수를 각각 겨냥한 구성으로, LIBERO-90 사전학습 후 테스크별 파인튜닝
  - **평균 92.6%로 기존 OpenVLA(76.5) · SpatialVLA(78.1) · CoT-VLA(69.0)를 능가**
  - Spatial 97.5 · Object 94.0 · Long 89.5 — 특히 공간 추론과 장기 태스크에서 강점[4)]

Table 2: **The extended LIBERO experiments.** DreamVLA achieves the best or competitive performance across all tracks compared to previous approaches. The best results are **bolded**.

Methods	Scores (%)				Average
	Spatial	Object	Goal	Long	
Diffusion Policy [92]	78.3	92.5	68.3	50.5	72.4
Octo [15]	78.9	85.7	84.6	51.1	75.1
OpenVLA [1]	84.7	88.4	79.2	53.7	76.5
SpatialVLA [38]	88.2	89.9	78.6	55.5	78.1
CoT-VLA [60]	87.5	91.6	87.6	69.0	81.1
DreamVLA	**97.5**	**94.0**	**89.5**	**89.5**	**92.6**

Methods (2)

DreamVLA

- **Experiments (3) : Real-world (Franka Panda)**
  - Franka Panda와 RealSense D415 2대(3인칭 · 손목)로 pick · place · drawer 세 태스크 구성
  - DROID로 사전학습 후 태스크당 100개 시연으로 파인튜닝, 시행당 최대 20회 연속 시도 허용
  - 전체 평균 76.7%로 Diffusion Policy(50.8) · Octo-Base(45.0) · OpenVLA(35.0)를 상회
  - 특히 Drawer는 67.5%로 베이스라인(35.0~37.5%) 대비 약 두 배

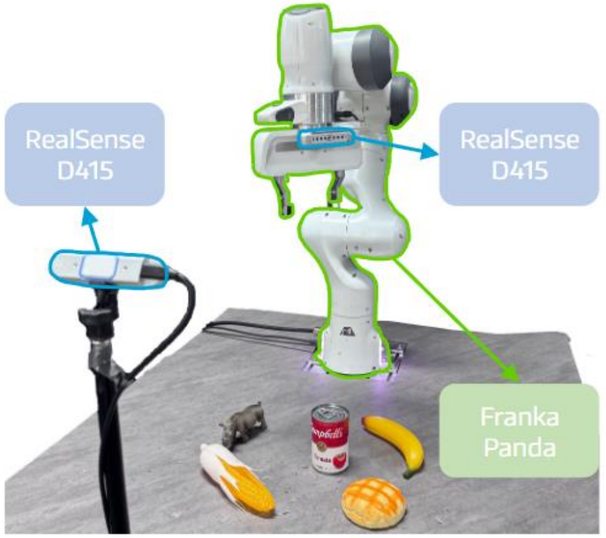


Figure 5: Real-world experiment setup.

Table 3: **Real-world evaluation** with the Franka Robot across three tasks.

Method	Pick			Place			Drawer			Task (All)
	Bottle	Doll	Avg.	Banana	Chili	Avg.	Open	Close	Avg.	Avg.
Diffusion Policy [92]	50.0	70.0	60.0	65.0	45.0	55.0	15.0	60.0	37.5	50.8
Octo-Base [15]	50.0	60.0	55.0	40.0	50.0	45.0	20.0	50.0	35.0	45.0
OpenVLA [1]	50.0	40.0	45.0	20.0	30.0	25.0	40.0	30.0	35.0	35.0
DreamVLA	**85.0**	**80.0**	**82.5**	**80.0**	**80.0**	**80.0**	**70.0**	**65.0**	**67.5**	**76.7**

Methods (2)

DreamVLA

- **Ablation Q1 : 각 modal(정보 종류) 요소의 기여도**
  - 동적 영역 · 깊이 · DINO · SAM 네 종을 각각 단독으로 학습시켜 기여도 비교
  - 동적 영역 단독이 4.32로 가장 크게 기여 — 마스크가 곧 변할 픽셀을 지목해 행동 의미론과 정렬함에 기인
  - 반면 깊이(2.98) · DINO(3.25) · SAM(3.13) 단독은 Vanilla VLA(3.64)보다 오히려 낮음
  - 큰 회귀 손실과 고차원 특징 매칭이 최적화를 지배해 태스크 관련 특징을 희석시키기 때문 → 정작 중요한 특징을 간과함

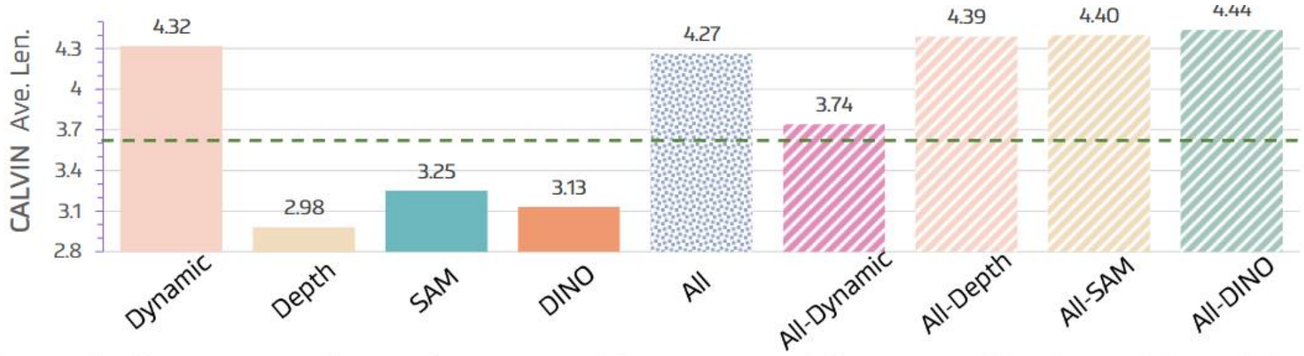


Figure 6: CALVIN ABC-D performance with respect to different combinations of knowledge prediction. All=all of five models, and All-X=taking X out of All.

Table 4: **Performance comparison** between predicting the optical flow and the dynamic region. Notably, the * denotes that this result is from [58].

Method	Task completed in a row					
	1	2	3	4	5	Avg. Len. ↑
Vanilla VLA*	93.0	82.4	72.3	62.6	53.3	3.64
+ dynamic region	97.6	92.6	87.5	80.4	73.7	4.32
+ depth	**98.3**	94.3	88.5	82.0	77.2	4.40
+ semantics	98.2	**94.6**	**89.5**	**83.4**	**78.1**	**4.44**

Methods (2)

DreamVLA

- **Experiments (3) : 정성 결과 및 추론 지연**
  - 동적 영역만 지도했음에도 장면 전체를 의미 있게 복원 — 긴 시퀀스에서 대부분 영역이 한 번은 동적이 됨
  - 깊이 예측은 MAE 계열 디코더 특성상 패치 단위로 거칠지만 공간 인식과 행동 정교화에 기여
  - 실기에서도 pick bottle / doll, place banana / chili 네 태스크를 안정적으로 수행
  - RTX 4090 기준 전체 91ms(11Hz)이며 dream query 추가 비용(지연 시간)은 3ms(3.4%)에 불과

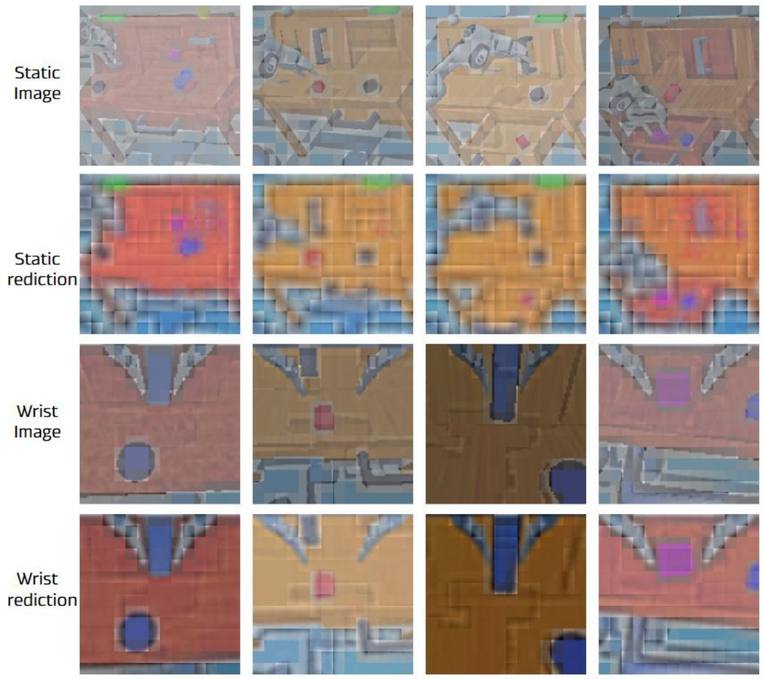


Figure 8: **Visualization results** of the dynamic region predictions.

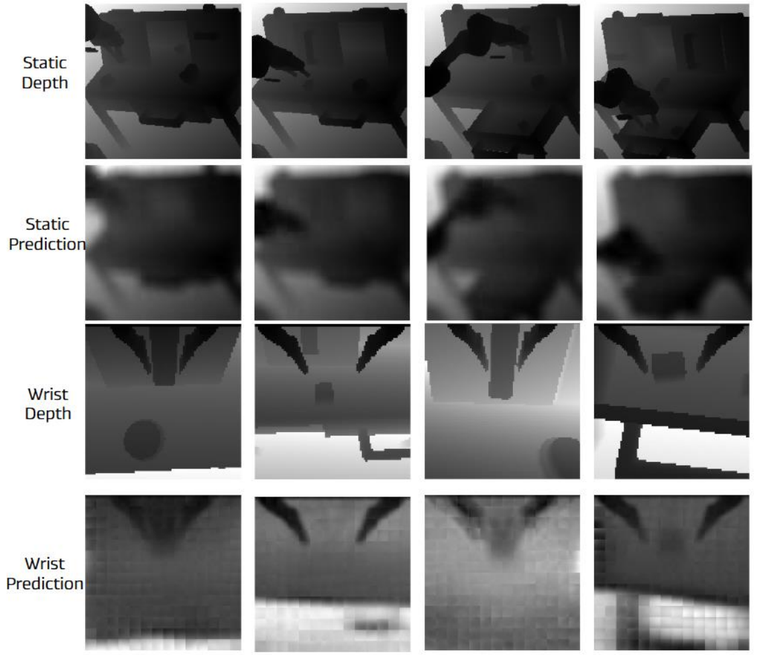


Figure 9: **Visualization results** of the depth maps.

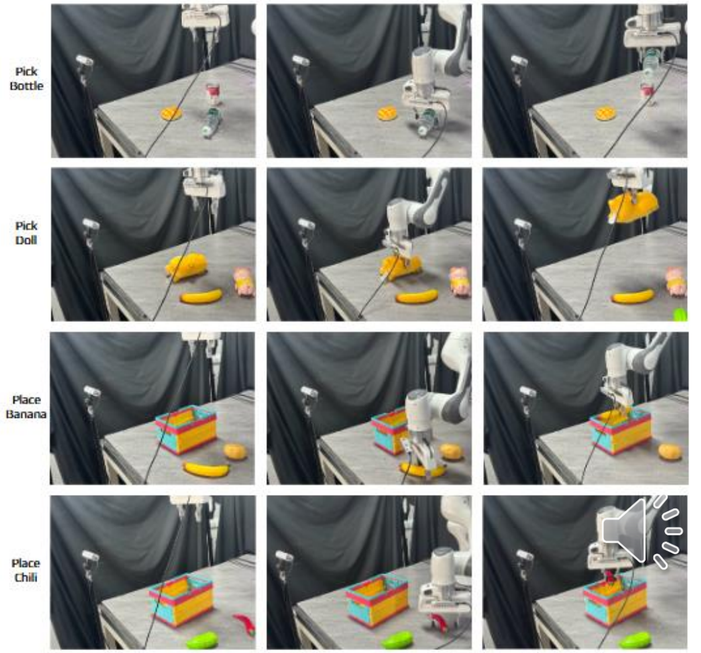


Figure 10: **Qualitative results** of real world language-grounded manipulation.

Conclusion

Summary

01

3D-VLA (2024)

- 목표 장면을 스스로 그려 행동의 기준으로 삼은 첫 VLA 월드 모델
- 2D 로봇 영상을 3D로 되살려 2M 규모 학습 데이터를 만든 것이 실질적 기여

02

DreamVLA (2025)

- 장면 전체가 아니라 움직임 · 깊이 · 의미만 골라 예측 — 「무엇을 예측할까」를 문제로 삼음
- 지식이 서로 섞이지 않게 어텐션을 막고, 확산 모델로 행동을 생성

03

VLA using World Model

- 예측을 정책 안에 넣는 단계를 지나, 월드 모델을 정책이 뛰노는 「환경」으로 쓰는 단계로 이동
- 최신(2026) 연구인 WMPO · WorldVLA · UniVLA · villa-X · Genie Envisioner 등이 위 흐름에 속함

Thank You

Data Mining
Quality Analytics