



[week2] PaLM-E: An Embodied Multimodal Language Model

PaLM-E : An Embodied Multimodal Language Model



Abstract

실제 감각 정보를 언어 모델에 통합한 구체화된 언어모델
→ 시각적 질문 응답 & 캡션과 같은 작업 수행을 위해 시각 및 text 를 포함
한 다양한 감각 정보를 입력 받는다.
⇒ 이 encoding은 end-to-end로 학습된다.

562B의 거대한 파라미터 수를 가지고 robotics task에서 학습되어 OK-VQA
에서 SOTA를 달성한 visual-language generalist 이다.
+) 모델 크기를 증가시키면서 일반 언어 기능을 유지한다.

Introduction

LLM은 다양한 도메인에서 좋은 성능을 보이고 있다.

그러나 실제 세계의 문제에 대해서 한계점을 가지고 있다. (visual & physical sensor)

(이것은 LLM이 textual한 input만 받기 때문일 수도 있다.)

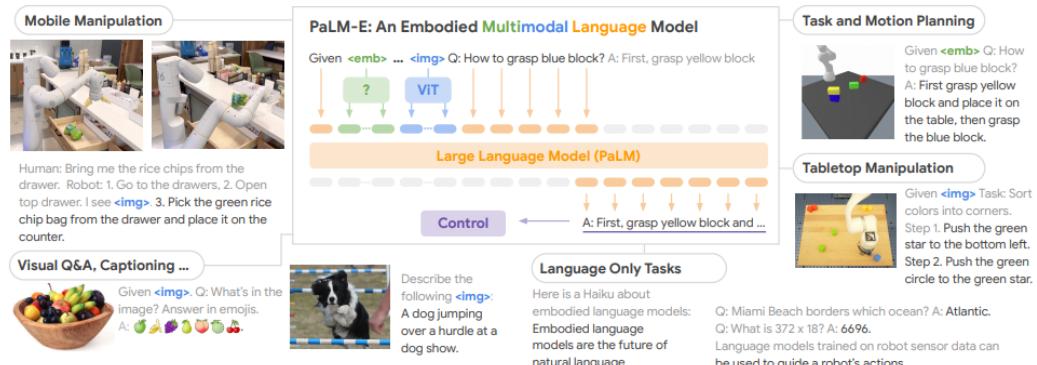


Figure 1: PaLM-E is a single general-purpose multimodal language model for embodied reasoning tasks, visual-language tasks, and language tasks. PaLM-E transfers knowledge from visual-language domains into embodied reasoning – from robot planning in environments with complex dynamics and physical constraints, to answering questions about the observable world. PaLM-E operates on *multimodal sentences*, i.e. sequences of tokens where inputs from arbitrary modalities (e.g. images, neural 3D representations, or states, in green and blue) are inserted alongside text tokens (in orange) as input to an LLM, trained end-to-end.

→ 본 모델은 일반적인 vision-language task(visual-question-answering)에 대해 SOTA를 달성했다.

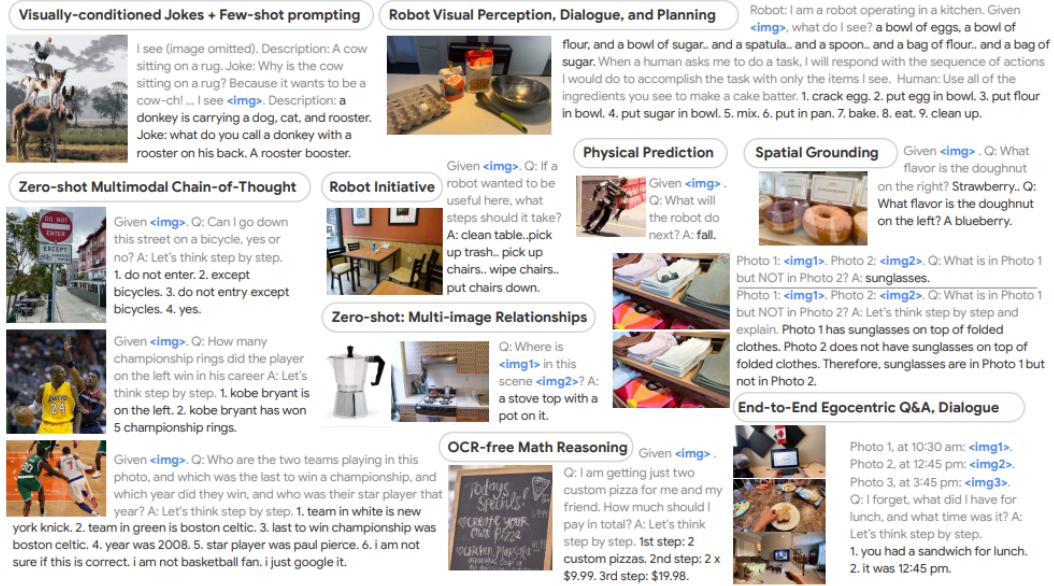


Figure 2: PaLM-E-562B can do *zero-shot multimodal chain-of-thought reasoning*, can tell visually-conditioned jokes given an image, and demonstrates an array of robot-relevant multimodal-informed capabilities including perception, visually-grounded dialogue, and planning. PaLM-E also generalizes, zero-shot, to multi-image prompts despite only being trained on single-image prompts. PaLM-E can also perform math given an image with textually-interleaved handwritten numbers. In addition, the model can perform, zero-shot, question and answering on temporally-annotated egocentric vision, similar to what was shown in (Zeng et al., 2022) but end-to-end all in one model.

how? ⇒ 구체화된 agent의 센서 모달리티로부터 continuous inputs을 직접 통합한다.

→ 언어 모델이 실제 세계에서 더 근거있는 추론을 할 수 있도록 한다.

image와 같은 인풋은 language token과 같은 방식으로 같은 latent 임베딩에 임베딩되고, transformer 기반의 LLM에서 text와 같은 방식으로 self-attention layer를 통해 처리된다.

(image와 같은 모달리티도 text와 동일한 방식으로 처리)

(→ 사전학습된 LLM의 인코더에 continuous input을 주입하여 진행)

[main contribution]

(1) 멀티모달 LLM에 구체화된 데이터를 섞어서 학습이 될 수 있도록 한다.

(2) 현재 SOTA 모델 (general-purposed visual-language model / zero-shot)은 구체화된 추론 문제를 성공적으로 다루지 못한다.

(3) 본 논문에서는 neural scene representations과 entity labeling multimodal tokens 같은 새로운 아키텍처를 제시한다.

(4) PaLM-E가 vision, language generalist로 경쟁력을 갖춤을 보인다.

(5) 언어 모델 크기를 확장하면 치명적 망각을 줄이면서 멀티모달 fine-tuning이 가능함을 보인다.

PaLM-E : An Embodied Multimodal Language Model

- main 아키텍처 아이디어

: 이미지나 state estimate, sensor 모달리티와 같은 연속적이고 구체화된 observation을 사전학습된 언어모델의 language embedding space에 주입한다.

→ 연속적인 observation을 언어 토큰의 임베딩 공간과 동일한 곳에 시퀀스 벡터로 인코딩한다.

⇒ prefix나 프롬프트가 주어지면 텍스트를 완성되게 생성하고, text와 continuous observation으로 구성된 문장을 얻는다.

: 질문에 대한 답변이나 로봇이 실행할 내용에 대한 텍스트를 출력하게 된다.

- decoder-only-LLM

$$p(w_{1:L}) = \prod_{l=1}^L p_{\text{LM}}(w_l | w_{1:l-1}),$$

: 토큰 시퀀스를 표현하는 text들의 확률을 예측하도록 학습된 모델
(pLM : large transformer network)

- Prefix-decoder-only LLM

$$p(w_{n+1:L}|w_{1:n}) = \prod_{l=n+1}^L p_{\text{LM}}(w_l|w_{1:l-1}).$$

: 모델 아키텍처를 변경하지 않고 prefix 또는 프롬프트에 따라 조건을 조정할 수 있는 자동 회귀 LM

→ prefix는 모델이 다음 토큰을 예측할 수 있는 context 제공

+) 프롬프트는 더 상세하게 설명하여 원하는 출력에 더 가깝게 예측이 가능하다

- **Multi-modal sentences: injection of continuous observations**

연속적인 observation을 얻어 임베딩 공간에 mapping하여 LLM에 주입

→ LLM의 prefix 형성을 위해 일반 텍스트 토큰과 같이 벡터 시퀀스에 continuous observation을 mapping하도록 인코더를 학습시킨다.

→ 정보 주입은 고정된 위치가 아니라 주변 텍스트 내에서 동적으로 수행된다.

(기존 LLM의 위치 인코딩을 재사용)

$$x_i = \begin{cases} \gamma(w_i) & \text{if } i \text{ a is text token, or} \\ \phi_j(O_j)_i & \text{if } i \text{ corresponds to observation } O_j. \end{cases}$$

- **Embodying the output: PaLM-E in a robot control loop**

: 멀티모달 문장 입력을 기반으로 텍스트를 생성하는 생성 모델

→ output은 구체화된 질문의 답변이나 장면 설명 task등의 솔루션으로 사용 가능하다.

: low level policy의 순서를 지정하고 제어하는 high-level policy로 이해할 수 있다.

(control loop에는 얻어 조건이 적용된다.)

Input & Scene Representations for Different Sensor Modalities

다양한 모달리티에 대해 다양한 아키텍처를 사용한다.

- **State estimation vectors**

: 포즈, 크기, 색상과 같은 정보를 포함한다.

→ MLP를 사용하여 state vector를 엄여 임베딩 공간에 mapping

- **Vision Transformer (ViT)**

: 이미지를 토큰 임베딩에 mapping

+) 논문에서는 ViT의 여러 변형을 고려

→ 언어모델과의 호환성을 고려하기 위해 학습된 affin transformation을 이용해서 임베딩을 투영한다.

- **Object-centric representations**

: 시각적 인풋을 LM에 주입 전에 별개의 객체로 분리한다. (시각적 인풋이 의미있는 단위로 사전 구조화가 되어 있지 않기 때문에)

→ grounded truth object instance mask를 이용해서 ViT representation을 별개의 객체 임베디으로 분해할 수 있다

⇒ 사전 훈련된 언어모델과 호환성이 향상된다.

- **Object Scene Representation Transformer (OSRT)**

: 새로운 뷰 합성 작업을 통해서 3D representation으로 학습 & 생성

→ 각 slot은 MLP를 사용하여 여러 임베딩으로 투영된다.

- **Entity referrals**

: 구체화된 계획 task에 대해 생성된 계획의 개체를 참조할 수 있다.

Training Recipes

- PaLM-E는 연속적인 observation, text, 인덱스가 포함된 샘플로 구성된 데이터 셋에 대해 학습되었다.

(텍스트에는 멀티 모달 문장으로 구성된 prefix 부분과 text 토큰만 포함된 예측 대상이 포함)

- 손실 함수 : 접두사가 아닌 개별 토큰에 대해 평균을 낸 크로스 엔트로피 loss
- 텍스트의 특수 토큰은 텍스트의 해당 위치에 인코더의 벡터를 삽입하여 대체되어 멀티모달문장을 형성
- PaLM-E는 디코더 전용 LLM인 PaLM의 사전 훈련된 8B, 62B 및 540B 변형을 기반으로 하며 사전 훈련 or 처음부터 훈련된 입력 인코더를 통해 연속적인 observation을 주입
- PaLM-E는 4B ViT, 22B ViT 또는 둘 다를 결합
→ 결과 모델은 각각 PaLM-E12B, PaLM-E-84B 및 PaLM-E-562B

+ 매개 변수가 고정되고 입력 인코더만 훈련되는 변형도 고려

→ 적절한 프롬프트가 제공될 때 LLM의 추론 기능을 활용하는 것이 목표

(soft 프롬프트의 한 형태)

Experiments

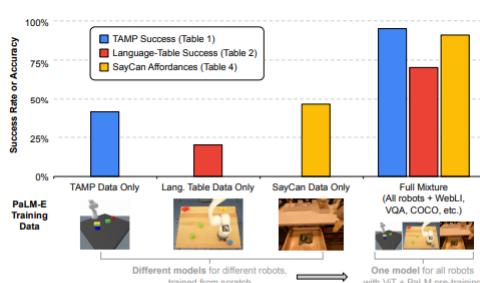


Figure 3: Overview of *transfer learning* demonstrated by PaLM-E across three different robotics domains, using PaLM and ViT pretraining together with the full mixture of robotics and general visual-language data provides a significant performance increase compared to only training on the respective in-domain data. See Tab. 1, Fig. 4, Tab. 2, Tab. 4 for additional data in each domain.

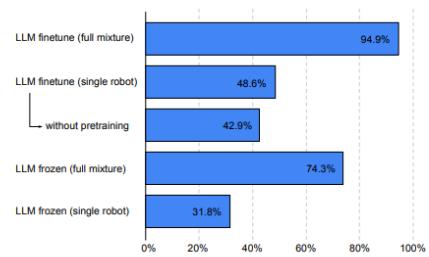


Figure 4: Planning success results in the TAMP environment (1% data) for PaLM-E-12B, comparing the effects of PaLM-E models (i) using the full training mixture, (ii) pre-training (ViT and PaLM), and (iii) freezing or finetuning the language model. Transfer from full mixture is particularly effective. Note that full mixture contains only 1% of the training data (320 examples each) for the tasks evaluated here. Shown is the mean of tasks p_1, p_2 .

Fig.3 : 다른 task와 dataset에 대해 동시에 학습된 것이 유의미한 성능 향상을 보여준다

Fig.4 : “full mixing”은 거의 2배의 성능 향상을 보인다.

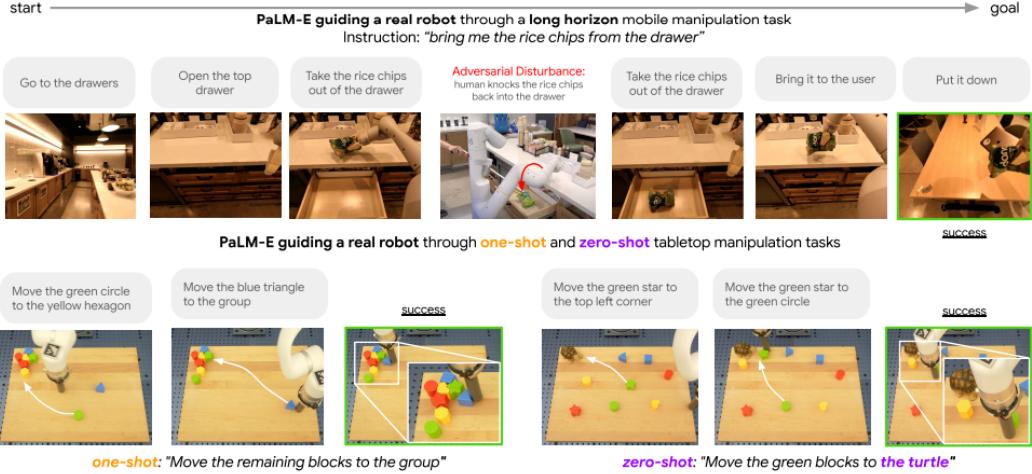


Figure 5: A single PaLM-E model directs the low-level policies of two real robots. Shown is a long-horizon mobile manipulation task in a kitchen, and one-shot / zero-shot generalization with a tabletop manipulation robot.

Object-centric	LLM pre-train	Embodied VQA	Planning					
			q ₁	q ₂	q ₃	q ₄	p ₁	p ₂
SayCan (oracle afford.) (Ahn et al., 2022)	✓	-	-	-	-	-	38.7	33.3
PaLI (zero-shot) (Chen et al., 2022)	✓	-	0.0	0.0	-	-	-	-
PaLM-E (ours) w/ input enc:								
State	✓(GT)	✗	99.4	89.8	90.3	88.3	45.0	46.1
State	✓(GT)	✓	100.0	96.3	95.1	93.1	55.9	49.7
ViT + TL	✓(GT)	✓	34.7	54.6	74.6	91.6	24.0	14.7
ViT-4B single robot	✗	✓	-	45.9	78.4	92.2	30.6	32.9
ViT-4B full mixture	✗	✓	-	70.7	93.4	92.1	74.1	74.6
OSRT (no VQA)	✓	✓	-	-	-	-	71.9	75.1
OSRT	✓	✓	99.7	98.2	100.0	93.7	82.5	76.2

Table 1: Comparison of different input representations on TAMP environment (in terms of success rates), where data from TAMP constitutes only 1% (i.e., 320 samples for p₁, p₂ each) of total training data size. PaLM-E outperforms both PaLI and SayCan on embodied VQA and planning tasks. Cross-domain *transfer* is observed, since the PaLM-E with ViT-4B trained on our full data mixture improves planning performance. OSRT, despite using no large-scale data, provides the most effective input encodings for learning. (GT) means ground-truth object-centric information provided. In all experiments, the LLM is frozen. The non-object centric ViT-4B variant utilizes color to reference objects, hence q₁ cannot be evaluated here. The LLM is frozen in these experiments (except for the case where it is not pre-trained). Sec. B.1 describes the tasks q₁-q₄, p₁, q₂.

Task 1. Q: There is a block that is closest to {i.e., top right corner}. Push that block to the other block of the same color.

Task 2. Q: How to sort the blocks by colors into corners?

Task 3. Q: How to push all the blocks that are on the {left/right} side together, without bringing over any of the blocks that are on the {right/left} side?

Table 3: Task prompts for Tab. 2.

Zero-shot Baselines						Task 1			Task 2			Task 3					
SayCan (oracle afford.) (Ahn et al., 2022)						0.0			-			-					
PaLI (Chen et al., 2022)						0.0			-			-					
PaLM-E-	trained on	from scratch	LLM+ViT pretrain	LLM frozen	Task finetune	# Demos			10	20	40	10	20	40	10	20	80
12B	Single robot	✓	✗	n/a	✓	20.0	30.0	50.0	2.5	6.3	2.5	11.3	16.9	28.3			
12B	Full mixture	✗	✓	✓	✗	-	-	20.0	-	-	36.3	-	-	29.4			
12B	Full mixture	✗	✓	✗	✗	-	-	80.0	-	-	57.5	-	-	50.0			
12B	Full mixture	✗	✓	✗	✓	70.0	80.0	80.0	31.3	58.8	58.8	57.5	54.4	56.3			
84B	Full mixture	✗	✓	✗	✗	-	-	90.0	-	-	53.8	-	-	64.4			

Table 2: Results on planning tasks in the simulated environment from Lynch et al. (2022).

Baselines			Failure det.	Affordance
PaLI (Zero-shot) (Chen et al., 2022)			0.73	0.62
CLIP-FT (Xiao et al., 2022)			0.65	-
CLIP-FT-hindsight (Xiao et al., 2022)			0.89	-
QT-OPT (Kalashnikov et al., 2018)			-	0.63
PaLM-E-12B	from scratch	LLM+ViT pretrain	LLM frozen	
Single robot	✓	✗	n/a	0.54
Single robot	✗	✓	✓	0.91
Full mixture	✗	✓	✓	0.91
Full mixture	✗	✓	✗	0.77
				0.91

Table 4: Mobile manipulation environment: failure detection and affordance prediction (F1 score).

Model	VQAv2		OK-VQA		COCO
	test-dev	test-std	val	Karpathy test	
<i>Generalist (one model)</i>					
PaLM-E-12B	76.2	-	55.5	135.0	
PaLM-E-562B	80.0	-	66.1	138.7	
<i>Task-specific finetuned models</i>					
Flamingo (Alayrac et al., 2022)	82.0	82.1	57.8†	138.1	
PaLI (Chen et al., 2022)	84.3	84.3	64.5	149.1	
PaLM-E-12B	77.7	77.9	60.1	136.0	
PaLM-E-66B	-	-	62.9	-	
PaLM-E-84B	80.5	-	63.3	138.0	
<i>Generalist (one model), with frozen LLM</i>					
(Tsimpoukelli et al., 2021)	48.4	-	-	-	
PaLM-E-12B frozen	70.3	-	51.5	128.0	

Table 5: Results on general visual-language tasks. For the generalist models, they are the same checkpoint across the different evaluations, while task-specific finetuned models use different finetuned models for the different tasks. COCO uses Karpathy splits. † is 32-shot on OK-VQA (not finetuned).



Figure 6: Results on general language tasks (NLG = natural language generation): increasing scale leads to less catastrophic forgetting between a corresponding PaLM-E model and its inherited PaLM model. See full suite of tasks and results in Tab. 8.

• Data efficiency

: PaLM-E는 다른 모델에 비해 robotics 측면의 아주 적은 데이터로도 해당 task에 대한 학습을 충분히 해낸다.

→ 이것은 미래의 large-scale visual data에 적용할 수 있을 것이다.

• Retaining language capabilities

: 멀티모달 학습 과정에서 모델의 언어능력을 유지할 수 있는 2가지 방법을 보여 준다

1) LLM을 고정시키고 인코더만 학습시키기

→ 이것은 종종 robotics task에서 어려울 수 있으므로 2번째로 대체 가능하다.

2) 전체 모델을 end-to-end로 학습시키기 (모델 scale을 증가시킬수록 기존의 언어모델의 성능에 가까워진다.)

Conclusion

- 멀티모달 정보를 사전학습된 LM 모델의 임베딩 스페이스에 임베딩하면서 구체화된 언어모델을 구축했다.
- 실험 결과를 통해 VQA와 캡셔닝 task에서 vision-language 모델의 SOTA 달성을 보였다.
- 로봇의 시뮬레이션과 실제 세계를 control하기 위해 PaLM-E 구축했다.
- 다양한 task를 섞어서 학습 시키면서 성능을 향상 시켰다
- 멀티모달을 학습시키면서도 언어 모델의 언어적 능력을 유지했다는데 의미가 있다.