CLAP: Direct VLM-to-VLA Adaptation via Language-Action Grounding

Yuri Ishitoya[1,2,†,‡] Jeremy Siburian[1,3,†,‡] Masashi Hamaya[1] Haruto Suzuki[1,†] Kuniaki Saito[1] Toshihiko Fukushima[1] Cristian C. Beltran-Hernandez[1] Mai Nishimura[1,‡]

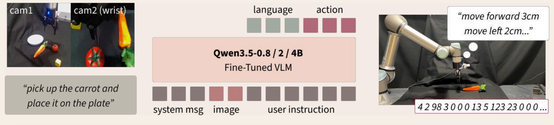
[1]OMRON SINIC X Corporation [2]Ochanomizu University [3]The University of Tokyo † Work done during an internship at OMRON SINIC X. ‡ Equal contribution.

👋 Turn any VLM into your own VLA with just 4× H100

SINIC X

Code
Models
Video

Overview



Instruction	*"pick up the carrot and place it on the plate"*
Language-action	`move forward 3 cm, move left 2 cm, …, close gripper`
Action tokens	`4 2 98 3 0 0 0 13 5 123 23 0 0 0 …`

Causal Language-Action Prediction: prepend a short language-action description. Causal attention conditions every action token on it.

- ✓ No arch. change
- ✓ No action expert
- ✓ No annotation

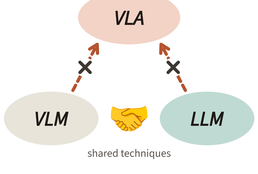
90.8%	+14.9 pt	60%	4–5 h
LIBERO avg. (2B)	vs. VLA-0 (2B)	UR5e (VLA-0 [1]: 15%)	fine-tuning, 1 node

Challenges in VLM-to-VLA Adaptation

Can we make VLA research as accessible as VLM research?

Ecosystem fragmentation

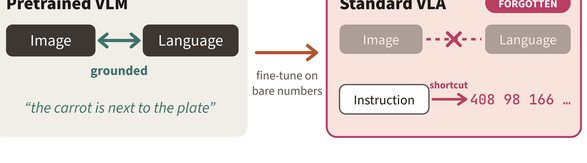
Advances in LLMs and VLMs cannot be directly applied to VLAs.



- × More model families
- × Model distillation
- × Speculative decoding
- × LLM serving (vLLM, llama.cpp)

Distribution mismatch

Fine-tuning on **bare numbers** breaks the VLM's **image–language bond**.



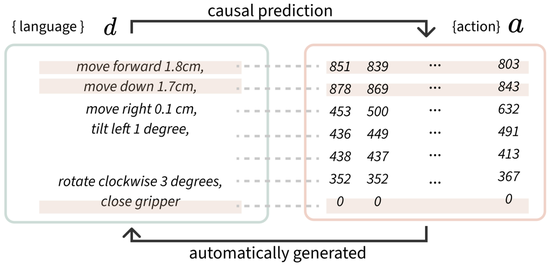
Causal Language-Action Prediction

Zero-modification architecture

CLAP keeps the pretrained VLM exactly as it is: **no action expert, no new vocabulary, no architectural change**. Actions are written as integer text tokens and predicted by the VLM's own language-model head, as in VLA-0 [1].

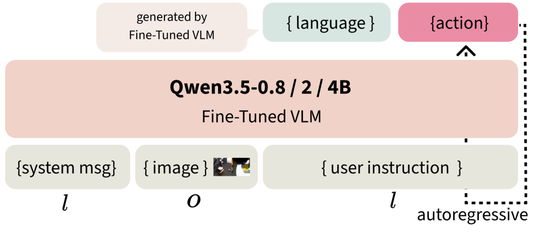
Training: language-action template

A fixed template turns the ground-truth action chunk $\tilde{a}$ into a description $\tilde{d} = T(\tilde{a})$; the VLM learns to output $\tilde{d}$, then $\tilde{a}$.



Inference: autoregressive generation

The prefix is stripped; action tokens decode into an 8-step chunk.



Example model output (LIBERO)

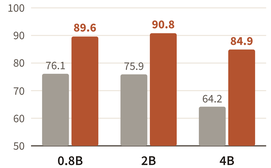
Instruction: *"grasp cube and drop it onto the plate"*

```
<think> move back 0.5 cm, move up 3.5 cm, move left 0.4 cm, tilt left 1
degrees, tilt back 1 degrees, rotate counterclockwise 3 degrees, open
gripper </think>
408 980 166 319 505 491 1000 407 980 159 312 496 486 1000 406 980 …
```

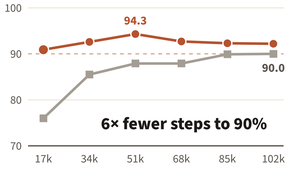
Efficient learning on LIBERO

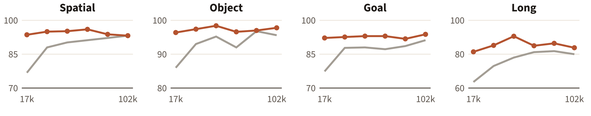
VLA-0 (repro.) CLAP (ours)

Avg. success (%) by model size · 17k steps



Avg. success (%) vs. training steps · 2B





6× fewer steps to reach **90%+** (4–5 h)
same-backbone VLA-0 needs 30 h+ to reach 90.0%

Effect of the language prefix

An action-grounded language prefix performs best

Ablation · 2B · LIBERO avg. success (%) axis from 70%

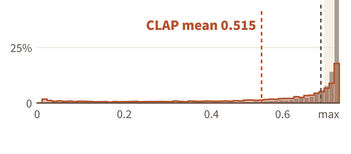
Action-token variants · no prefix		
VLA-0 [1]	*none*	75.9
Decimal strings [2]	`"0" "." "2" "0" "5" …`	81.1
Prefix variants		
Random words	`cavern cobblestone cardigan …`	79.0
Delta notation	`dx: +6.18, dy: -2.09, …`	88.7
Language prefix (CLAP)	`move forward 1.8cm, move down …`	**90.8**

The prefix anchors the model to the base VLM

Per-token JS divergence from frozen Qwen3.5-2B on generic VQA prompts (256 images).

Tokens whose prediction still overlaps the base VLM
CLAP **67.1%** vs VLA-0 **31.3%**
the rest are nearly disjoint (JS ≈ max)

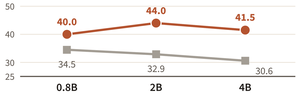
% of tokens by JS divergence — VLA-0 CLAP



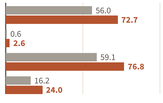
More robust on LIBERO-PRO

VLA-0 CLAP

Avg. success (%) by model size



2B, by perturbation type (%)



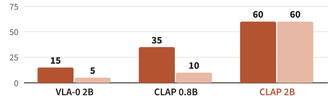
CLAP wins at every scale (up to **+11.1 pt**) and on every perturbation type; object relocation stays unsolved for both.

Real robot (UR5e)



120 teleop demos · trained from the vanilla VLM · OOD adds unseen distractors

Pick-and-place success (%), 20 trials — ID OOD



2B CLAP keeps **60%** even with unseen distractors; same-backbone VLA-0 drops to 5%.

Inference speed (RTX 4090, h=8)	default [s]	compiled [s]	compiled [Hz]
VLA-0 2B [1]	3.88	1.44	5.55
CLAP 0.8B	4.23	1.32	**6.06**
CLAP 2B	4.31	1.95	**4.11**
CLAP 4B	6.01	3.87	2.07

Discussion & Future Work

Does the action follow the words? Prefix-action consistency is not enforced; probing whether action tokens truly attend to the plan is an open question.

Next steps. Expand to more VLM families and analyze how VLA conversion changes VLM capabilities; bring in the whole LLM/VLM ecosystem: distillation, KV caching, speculative decoding, and more.

References

[1] A. Goyal et al., "VLA-0: Building State-of-the-Art VLAs with Zero Modification," arXiv, 2025.
[2] S. Grover et al., "Enhancing Generalization in Vision-Language-Action Models by Preserving Pretrained Representations," IROS, 2026.

Acknowledgement

This work was supported by JST PRESTO Grant Number JPMJPR2518 and the ABCI 3.0 "Development Acceleration Use" program (AIST and AIST Solutions).