CVPR JUNE 3-7, 2026 DENVER COLORADO

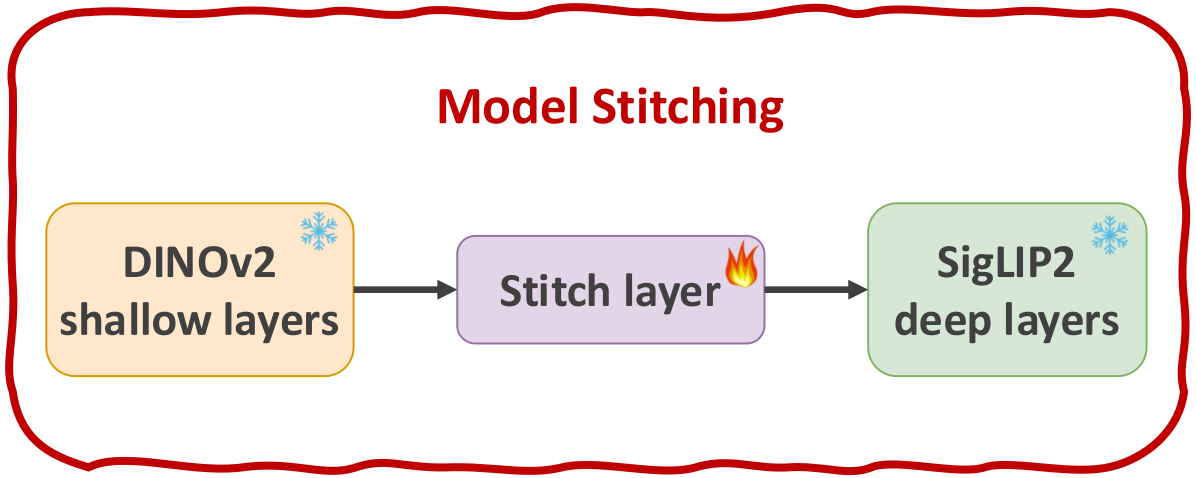
Revisiting Model Stitching In the Foundation Model Era

Zheda Mai[1,3], Ke Zhang[3], Fu-En Wang[3], Zixiao Ken Wang[3], Albert Y. C. Chen[3], Lu Xia[3], Min Sun[3], Wei-Lun Chao[2], Cheng-Hao Kuo[3]

[1]The Ohio State University, [2]Boston University, [3]Amazon

Summary

Can heterogeneous Vision Foundation Models (VFMs) be connected, fused, and served more efficiently?

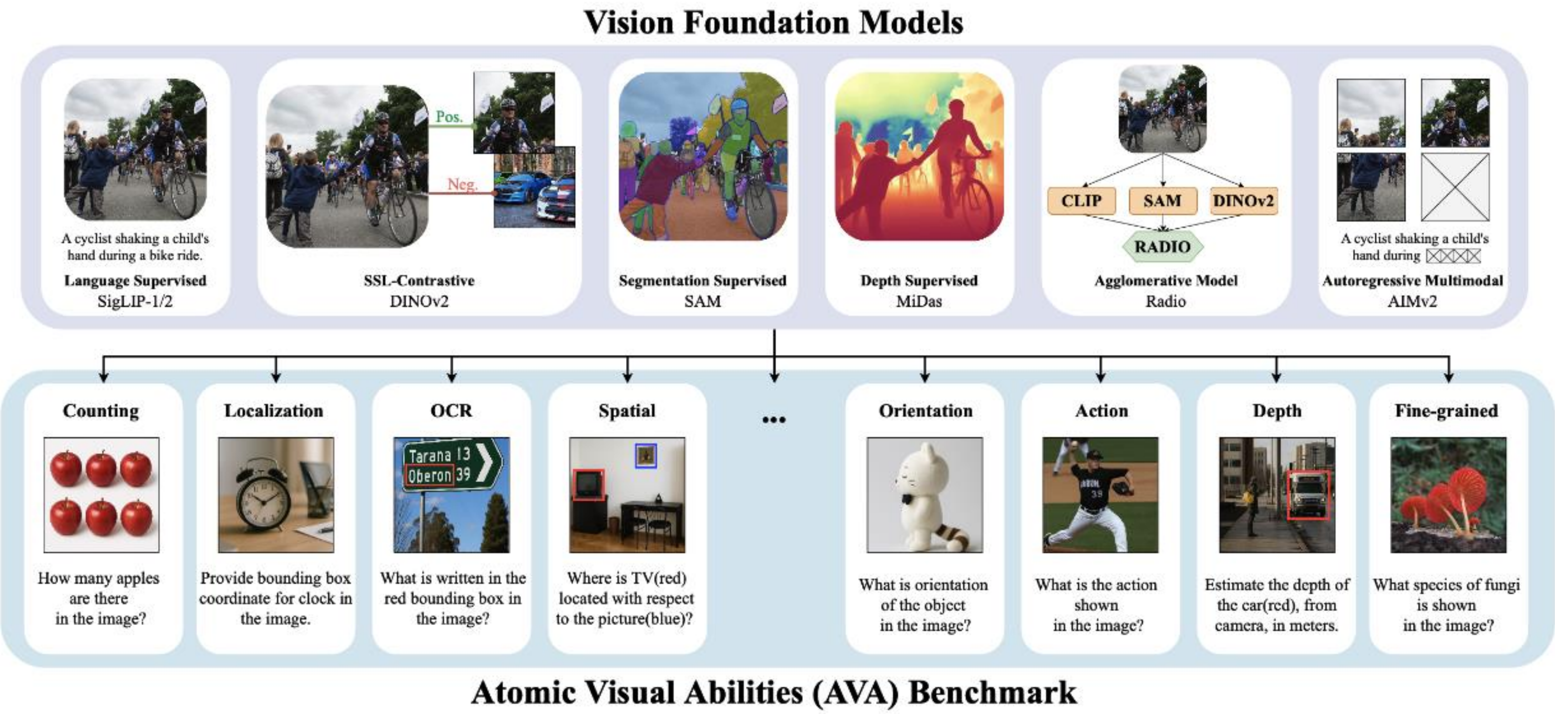


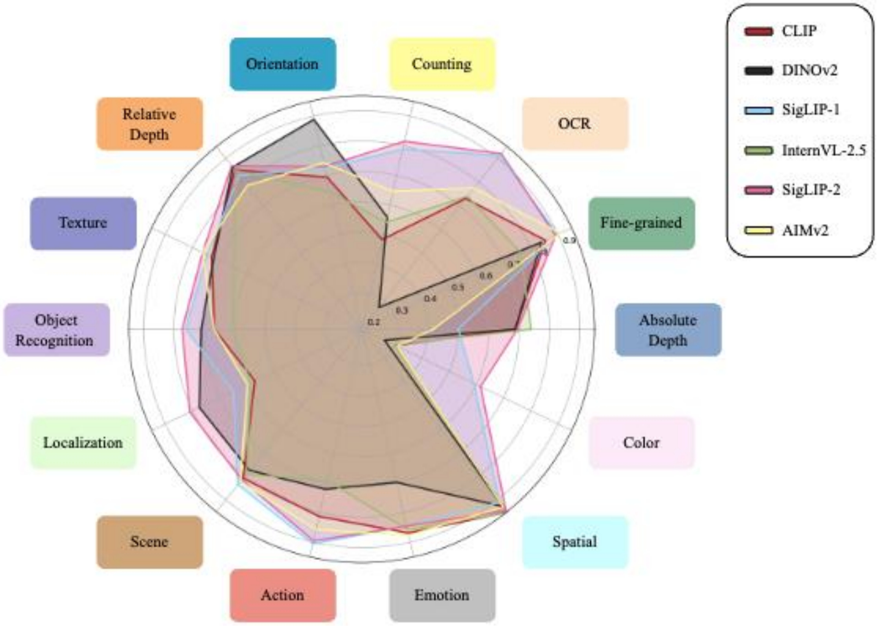
Key finding:

- With appropriate stitching, different VFMs are not just compatible — they can be complementary.
- Model Stitching is a practical recipe for fusing complementary VFM to **reduce cost** of multi-VFM systems.

CVPR JUNE 3-7, 2026 DENVER COLORADO

Vision Foundation Models (VFMs)





VFMs trained with **different** data, supervision and training objectives

Common strengths

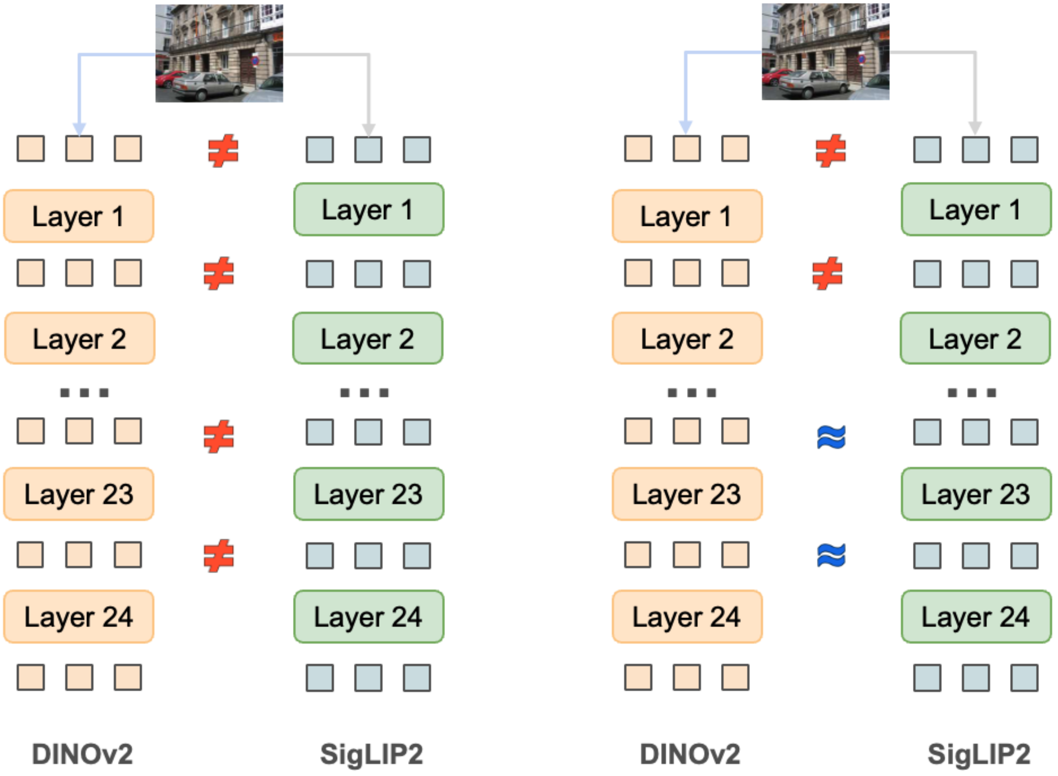
Unique strengths and weaknesses

AVA-Bench: Atomic Visual Ability Benchmark for Vision Foundation Models. **Mai et al., CVPR 2026**.

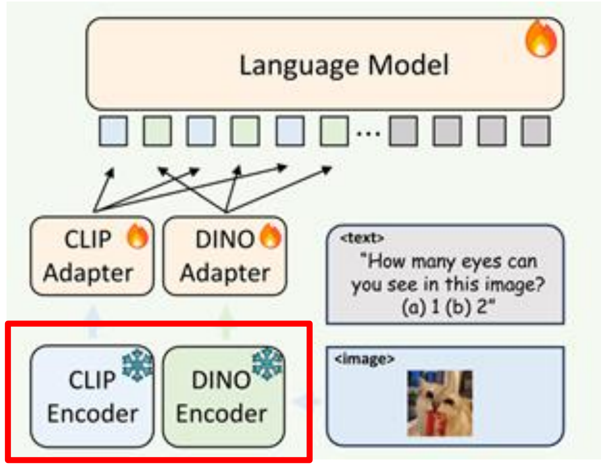
Are thy different end-to-end?

We may want to ask:

- Are VFMs fundamentally different end-to-end?
- Are their internal representations similar or compatible in some layers?



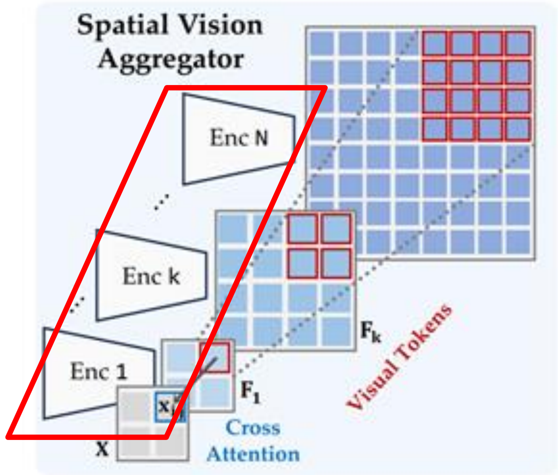
Multi-VFMs



Eyes Wide Shut? Exploring the Visual Shortcomings of Multimodal LLMs

Shengbang Tong[1] Zhuang Liu[2] Yuexiang Zhai[3] Yi Ma[3] Yann LeCun[1] Saining Xie[1]

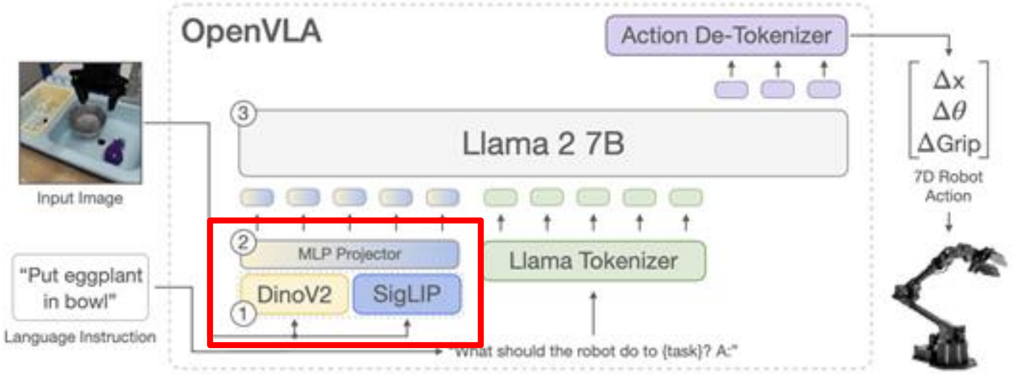
[1]New York University [2]FAIR, Meta [3]UC Berkeley



***Cambrian*-1: A Fully Open, *Vision-Centric* Exploration of Multimodal LLMs**

Shengbang Tong* Ellis Brown* Penghao Wu* Sanghyun Woo Manoj Middepogu Sai Charitha Akula Jihan Yang Shusheng Yang Adithya Iyer Xichen Pan Austin Wang Rob Fergus Yann LeCun Saining Xie[†]

New York University



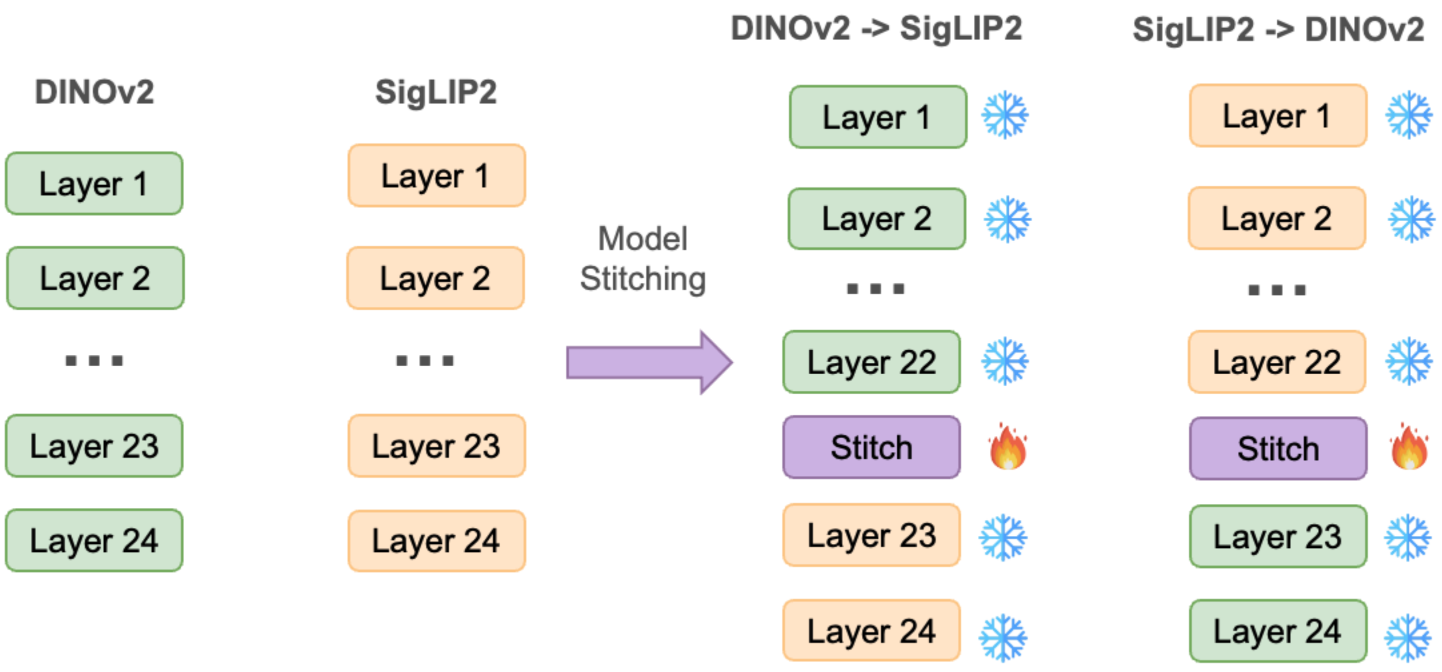
OpenVLA: An Open-Source Vision-Language-Action Model

Moo Jin Kim*,1 Karl Pertsch*,1,2 Siddharth Karamcheti*,1,3 Ted Xiao[4] Ashwin Balakrishna[3] Suraj Nair[3] Rafael Rafailov[1] Ethan Foster[1] Grace Lam Pannag Sanketi[4] Quan Vuong[5,†] Thomas Kollar[3] Benjamin Burchfiel[3] Russ Tedrake[3,6] Dorsa Sadigh[1] Sergey Levine[2] Percy Liang[1] Chelsea Finn[1]

With the better understanding of their internal representations, can we serve multiple VFMs more <u>efficiently</u>?

Model Stitching

Model Stitching: connect the **shallow layers** of model A to the **deep layers** of model B with a stitch layer (e.g., linear or MLP) to form a stitched model



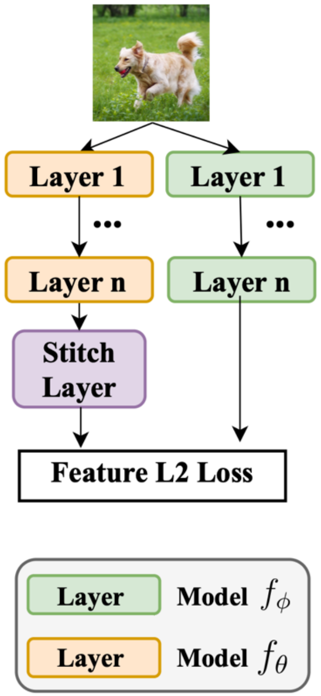
- Only stitch layer is trainable, all other layers in both models are frozen
- If the stitched model has **minimal performance drop**, it means their internal representations are similar

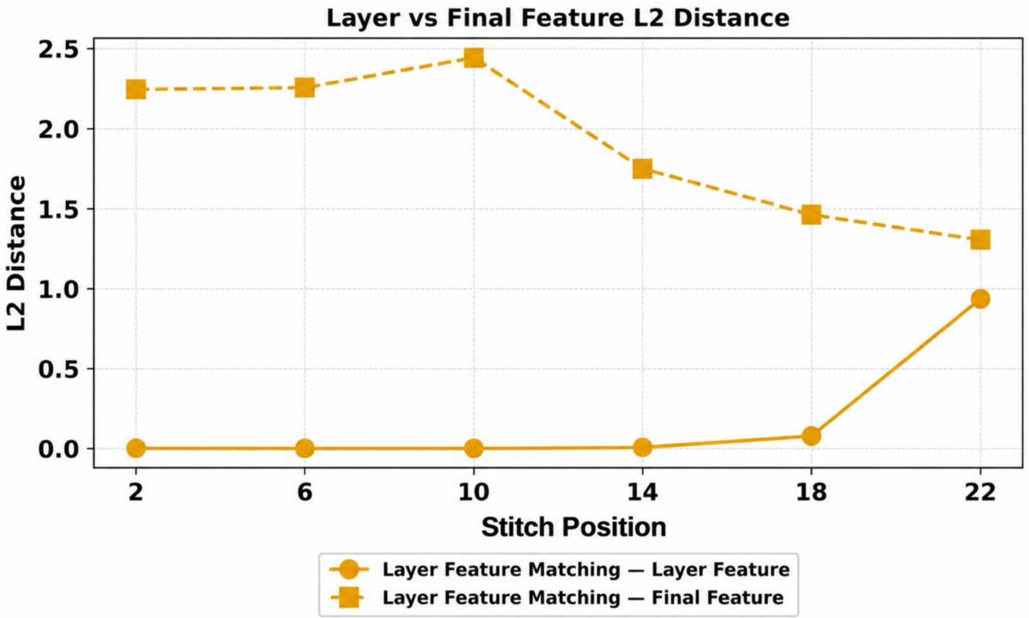
How to Train the Stitch Layer?

Naive stitching fails in the VFM regime

Layer Feature Matching:
match features at the stitch position

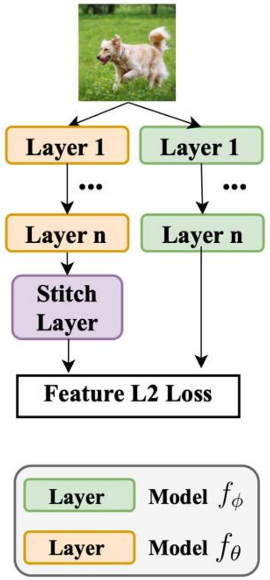
Problem: a small difference at the stitch position could accumulate and be amplified by the frozen target layers -> a pronounced final feature difference.



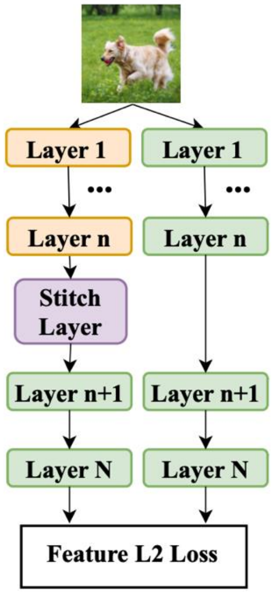


Final Feature Matching

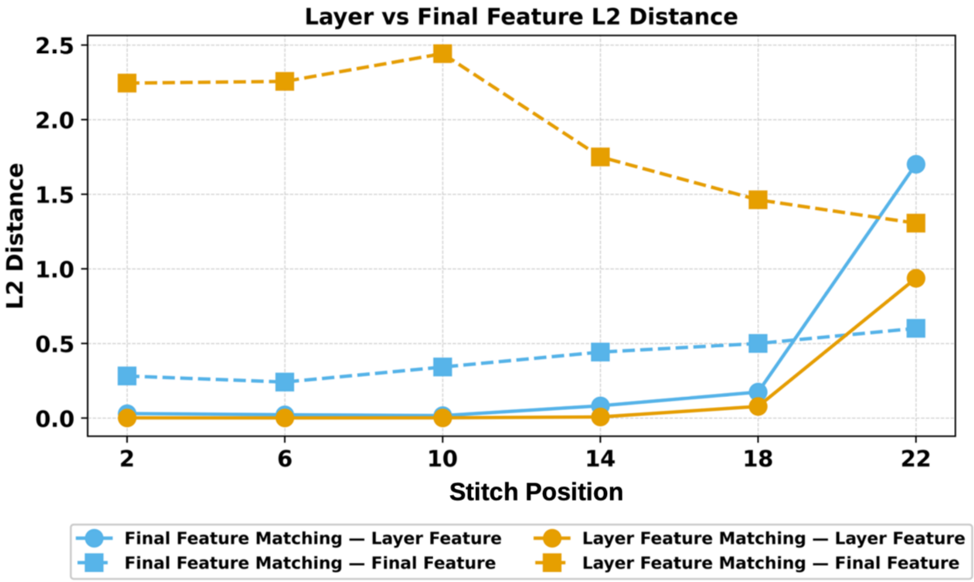
Final Feature Matching:
match final features between the stitched model and target model

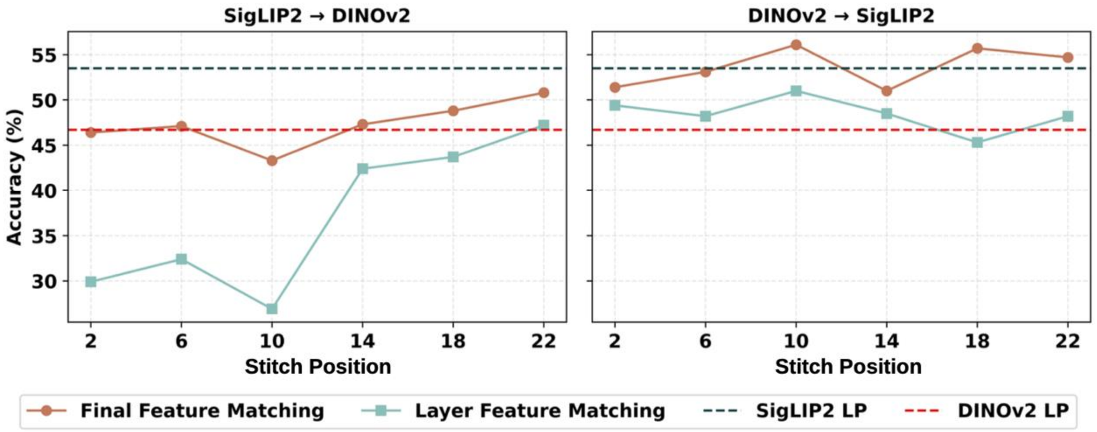


Layer Feature Matching



Final Feature Matching

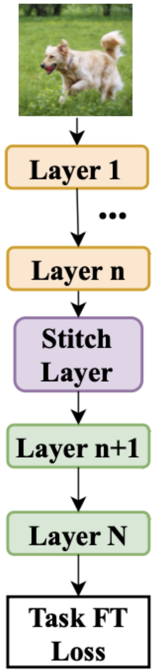




How to Train the Stitch Layer?

Naive stitching fails in the VFM regime

Task-Loss Training:
optimize the downstream objective only



Problem:

- When stitching **shallowly**, all post-stitch target layers are **frozen**
- Gradients must backprop through a long, fixed transformation to *update only the stitch layer*.
- Optimization challenges, especially under random initialization

Task Loss Training	DINOv2	SigLIP2	Layer Feature Matching	Final Feature Matching
25.1%	46.7%	53.4%	49.4%	51.4%

DINOv2->SigLIP2 stitched at layer 2 on fMoW

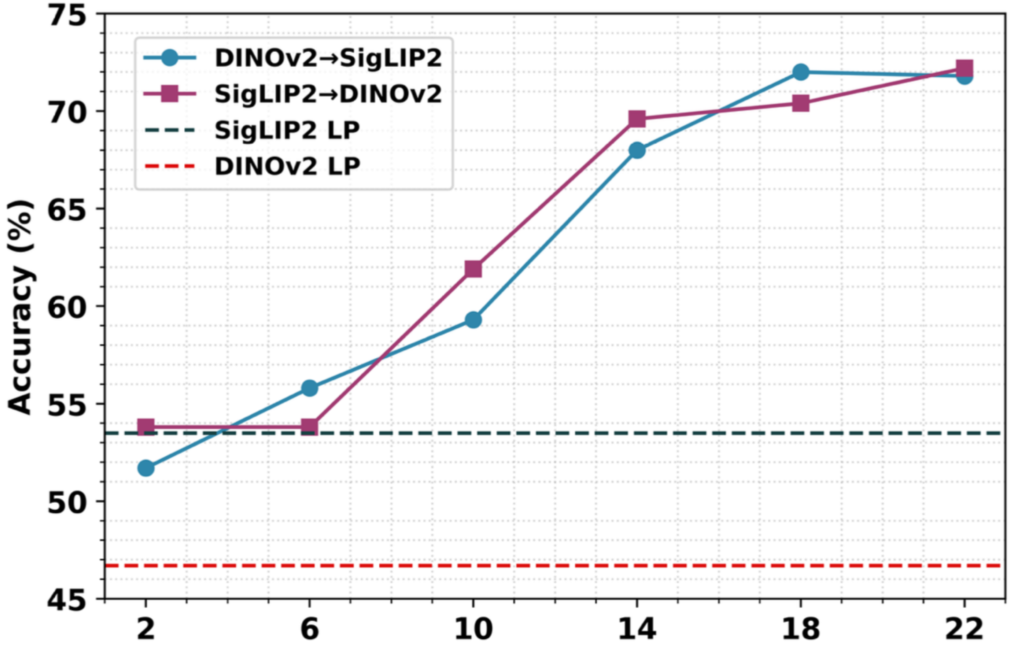
Our Stitching Recipe

Stage 1
Final Feature Matching Pretraining

Label-free pre-train: match the target model's final representation

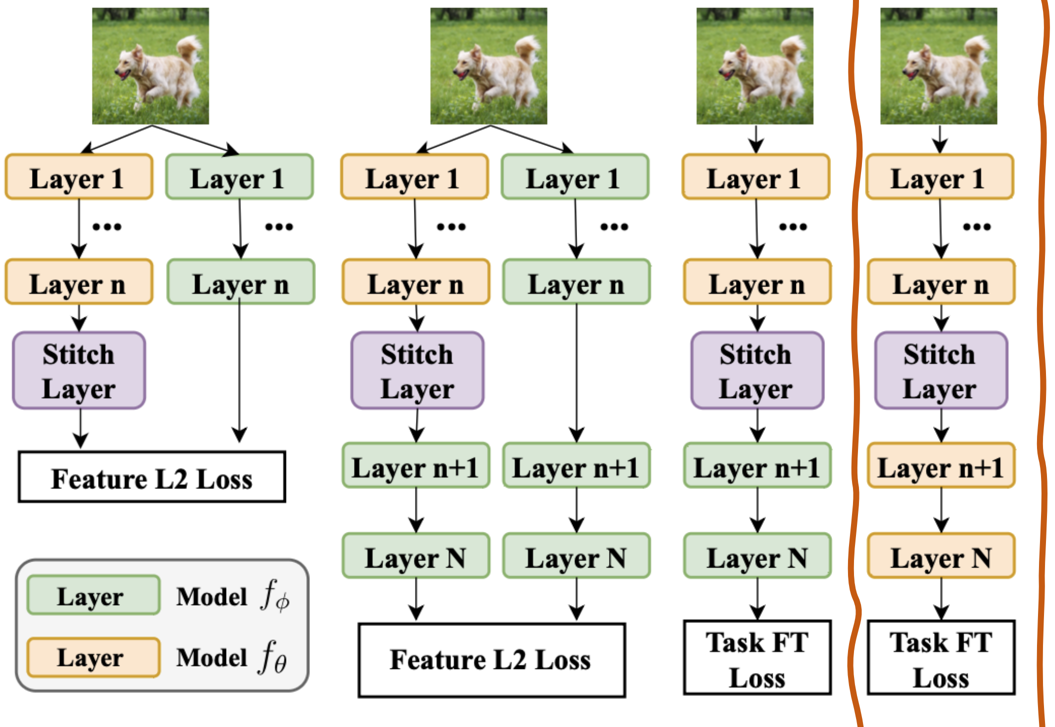
Stage 2
Task-Loss Fine-Tuning

Better adapt to the downstream task with pretraining



LP: Linear Probing

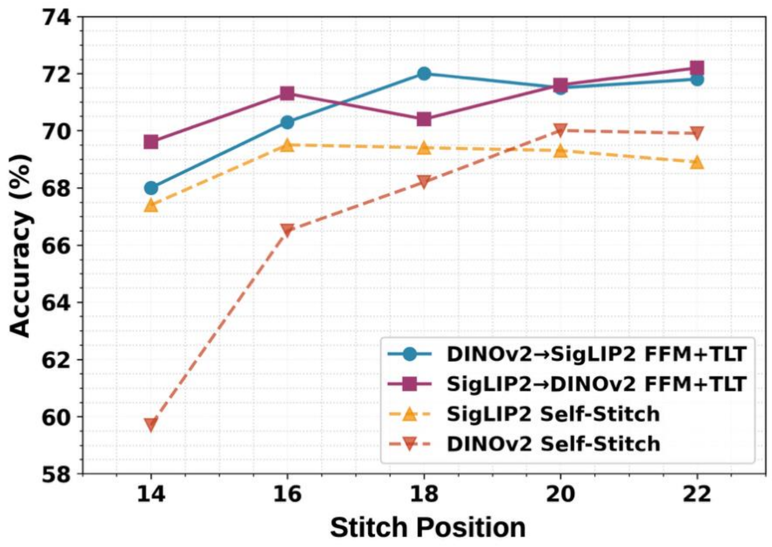
Knowledge Fusion, Not Just Capacity



(a) Layer Feature Matching (b) Final Feature Matching (c) Task Loss Training (d) Self Stitch

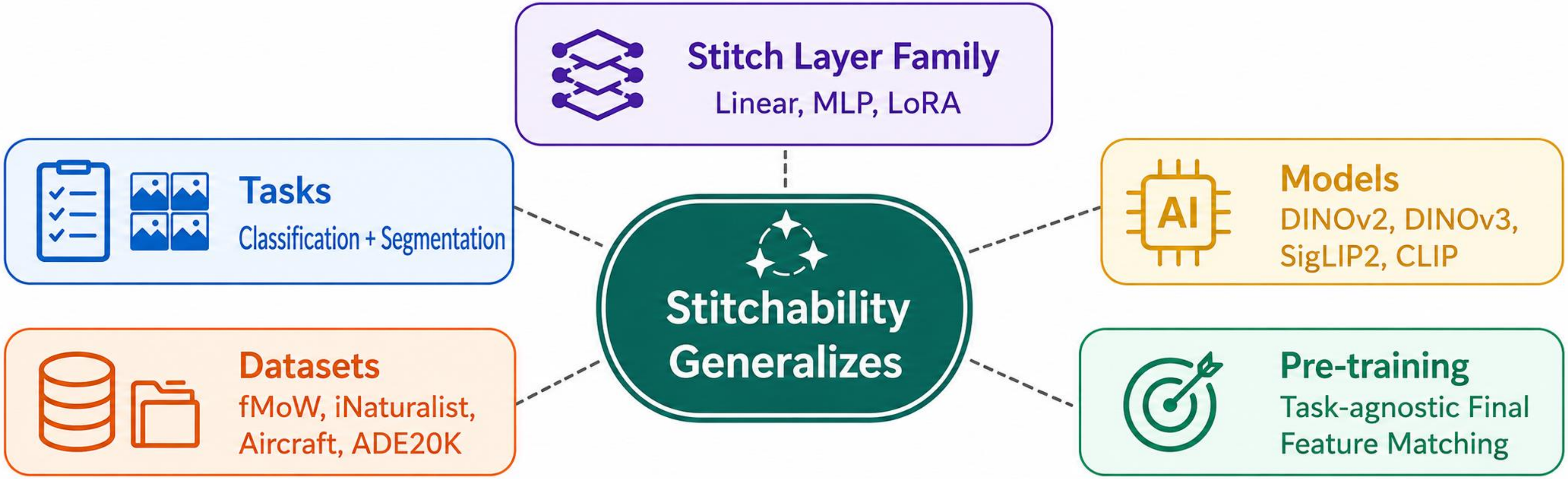
Self-Stitch Baseline:

- Insert same stitch layer into source-only and target-only models.
- Same stitch position, training data, and objective.
- Isolate whether improvement is only from the stitch layer.

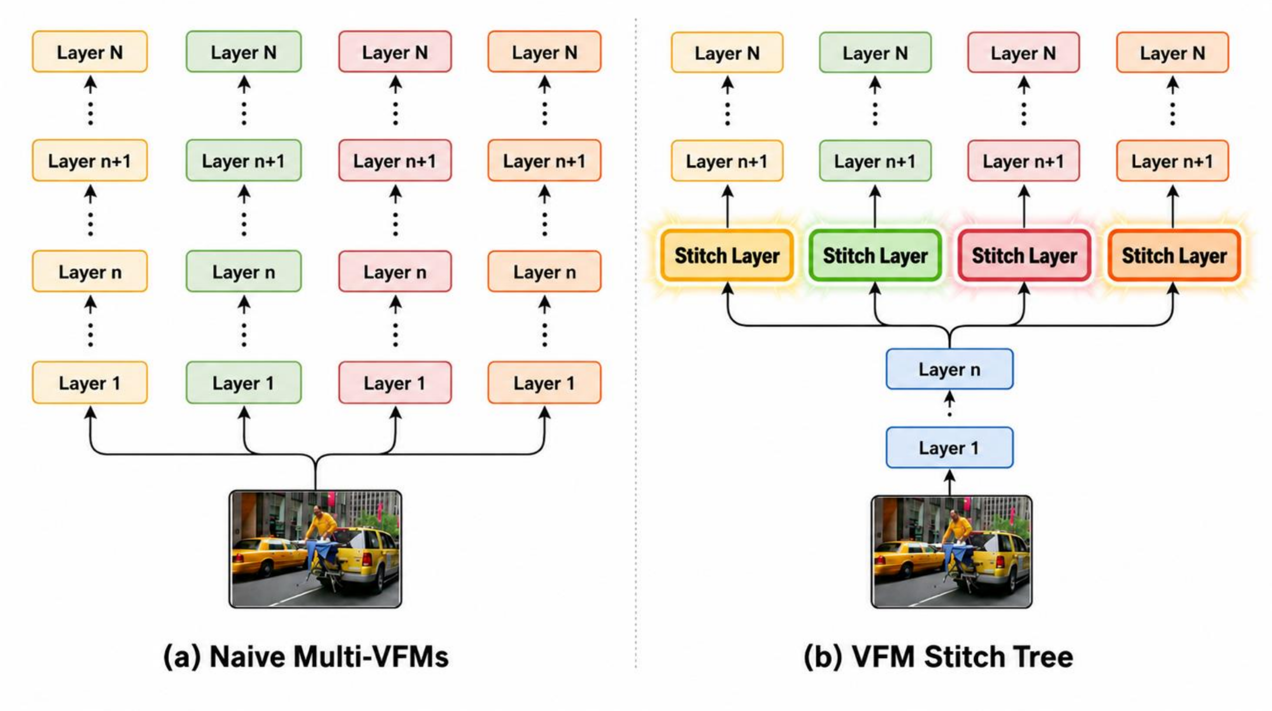


Stitching heterogeneous VFMs is not only feasible but also has the potential to fuse complementarity between them.

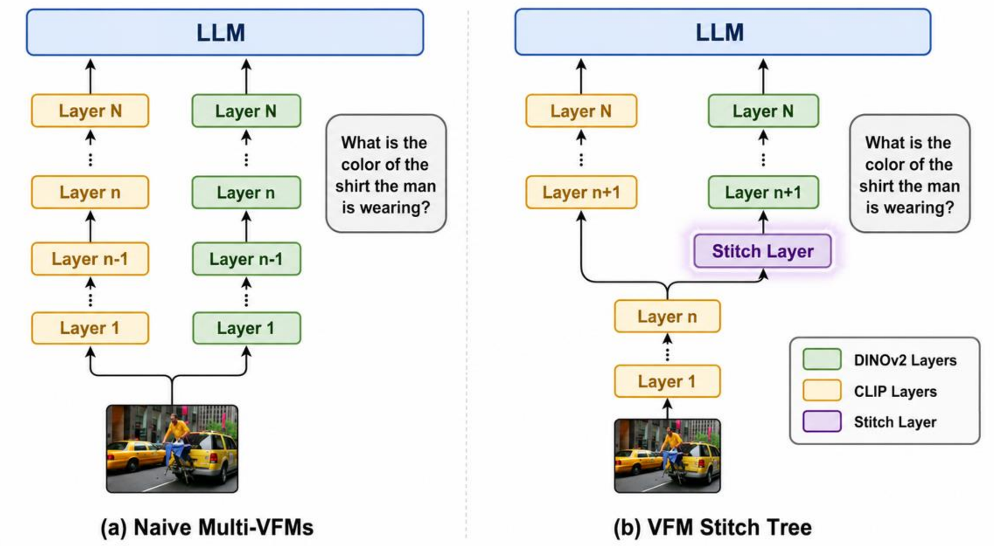
Generality: Not a One-off Result

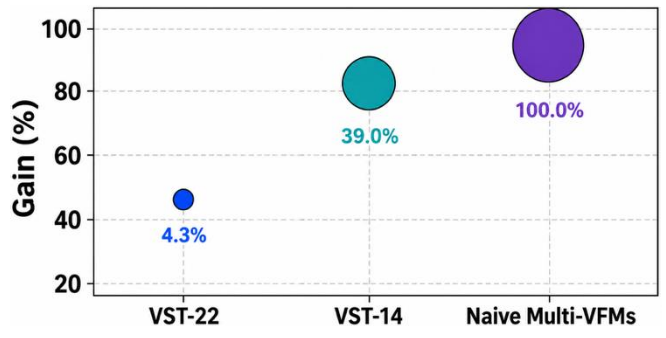


Application: VFM Stitch Tree (VST)



Share shallow layers, retain **specialized** deep branches





Example of VST in a MoF-LLaVA:
VST achieves much of the two-VFM gain with **lower** overhead.

Takeaways

1

Stitch Layer Training matters

Naive stitching fails for VFMs; Final Feature Matching creates a strong initialization;

2

Stitching fuses strengths

Cross-VFM stitches beat self-stitch controls, suggesting real complementarity.

3

Stitching enables systems

VFM Stitch Tree turns multi-VFM models into a controllable accuracy-latency trade-off.

Our work advances model stitching from a representation-analysis tool to a practical paradigm for integrating heterogeneous VFMs.