[RSS'25] OpenVLA-OFT
: Fine-Tuning Vision-Language-Action Models_Optimizing Speed and Success

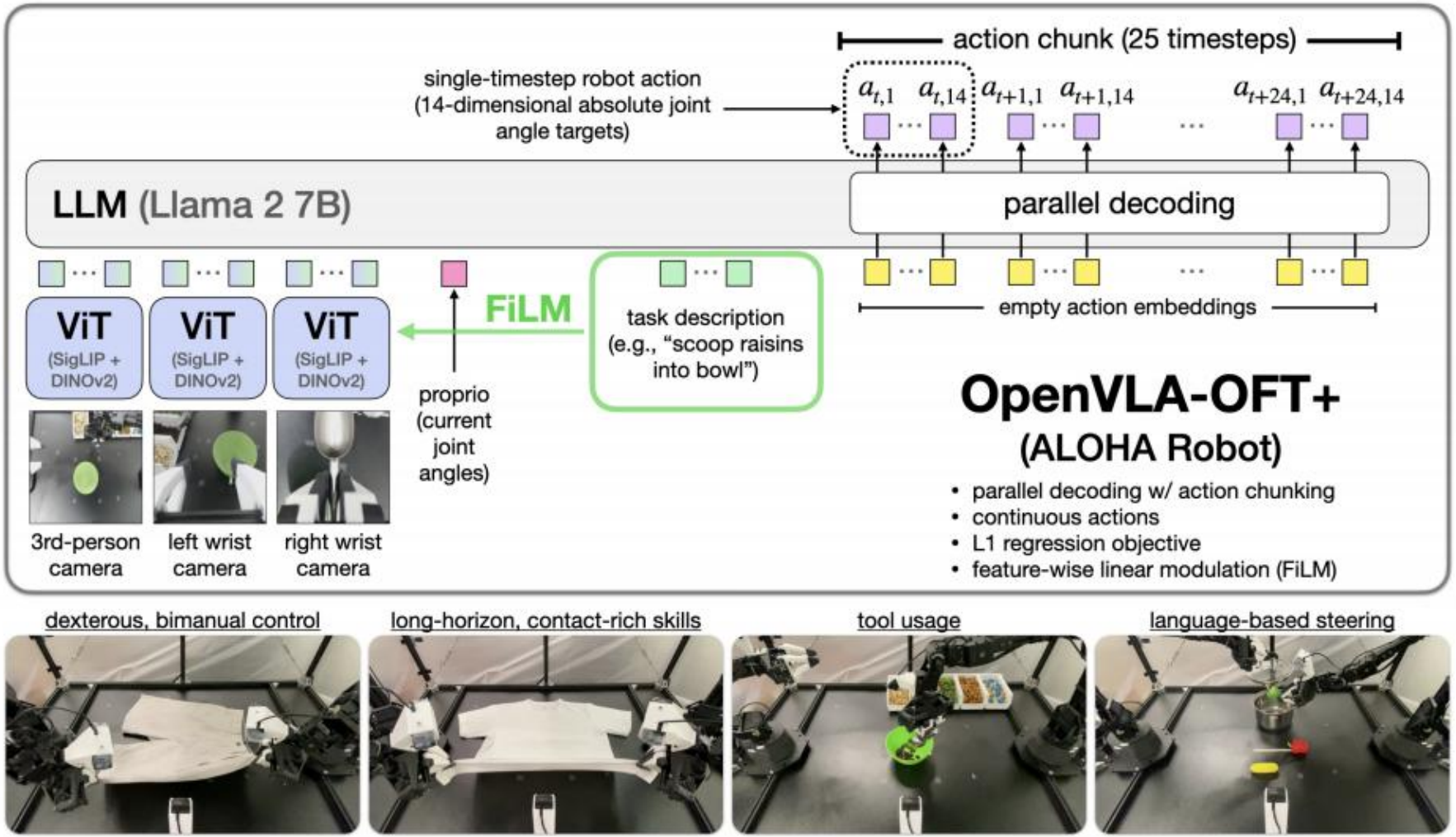

Fig. 1: OpenVLA-OFT+ on the bimanual ALOHA robot. Our Optimized Fine-Tuning (OFT) recipe enhances fine-tuned OpenVLA policies through improved inference efficiency, model quality, and input-output flexibility. The resulting OpenVLA-OFT+ policies execute diverse dexterous manipulation tasks on a real-world bimanual robot at high control frequencies (25 Hz). The "+" suffix indicates the integration of feature-wise linear modulation (FiLM) [37], which strengthens language grounding in tasks where accurate language understanding is critical for success.

<Introduction>

- The paper provides the first **comprehensive empirical study of VLA fine-tuning strategies**, identifying a simple yet powerful recipe—**OFT**—that achieves:
  - Near-perfect task success in simulation (97.1%)
  - Real-time bimanual control in real robots
  - Up to **43× faster inference**
- It thus establishes a **new standard for fine-tuning efficiency and effectiveness** in Vision-Language-Action models.

<Related Works>

- The paper **systematically unifies** insights from diverse research threads into a **single fine-tuning framework (OFT)** that:
  - Matches or exceeds prior performance across all categories,
  - Avoids complexity (no diffusion or RL),
  - And achieves **real-time, bimanual control** purely via **offline imitation learning**.

<Preliminary>

Section III establishes the **technical foundation and bottleneck** of current VLA training:

- OpenVLA's **autoregressive, discrete, next-token setup** severely limits speed and efficiency.
- **Action chunking** improves stability but is **computationally prohibitive**.
- This motivates the authors to design a **parallel, continuous, L1-based fine-tuning scheme (OFT)** that will be elaborated in later sections.

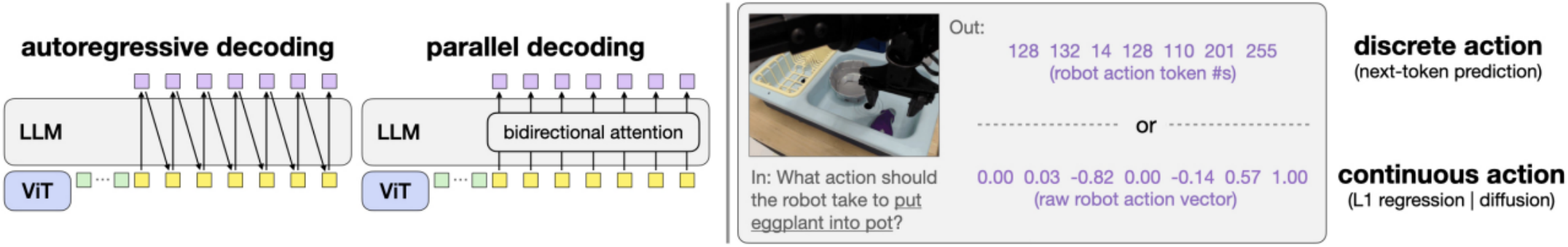

Fig. 2: Key design decisions for VLA fine-tuning. Left: Comparison between autoregressive decoding, which generates actions sequentially, and parallel decoding, which leverages bidirectional attention and generates all actions in a single forward pass. **Right**: Comparison between discrete action tokens with next-token prediction and continuous action values with L1 regression or diffusion modeling objectives. The original OpenVLA training scheme includes autoregressive decoding, discrete actions, and next-token prediction.

<Method>

- **Parallel decoding** removes the autoregressive bottleneck, making VLA fine-tuning feasible for real-time (25–50 Hz) control.
- **Continuous actions + L1 regression** achieve the best trade-off between expressiveness and efficiency.
- **FiLM integration** solves the "language-following" issue in complex visual environments.
- These insights collectively form the foundation of **OFT (Optimized Fine-Tuning)** — a recipe that yields *fast, accurate, and language-grounded* robotic control policies.

<Experiments>

- **Parallel decoding + chunking** is the single most impactful change, boosting speed and accuracy simultaneously.
- **Continuous actions + L1 loss** offer diffusion-level performance at fraction of the cost.
- **Multimodal flexibility** (scales to extra sensors) comes "for free" thanks to PD efficiency.
- The final **OpenVLA-OFT** achieves **SOTA performance on LIBERO** and is practical for real-time robot control.

<Experiments (Real-World ALOHA datasets)>

- **OFT+ bridges the simulation-to-real gap.**
  It adapts a single-arm sim-pre-trained VLA to a bimanual real-robot without re-architecture.
- **Fine-tuning > pretraining scale.**
  Careful adaptation (parallel decoding, continuous L1, FiLM) beats larger models with massive bimanual data.
- **Real-time efficiency is achieved without sacrificing accuracy.**
  Parallel decoding enables >25 Hz control on a 7.5 B transformer.
- **FiLM is critical for language grounding.**
  Eliminating it collapses instruction following to random chance.
- **OFT+ offers a general template** for adapting any VLA/VLM to new robots via lightweight offline fine-tuning—efficient, interpretable, and real-world viable.