TempoRAL: Cross-Embodiment Temporal Abstraction Transfer from Human Demonstrations via Internal Reinforcement Learning in VLM Backbones

Anonymous Author[1]

***Abstract*— Vision-Language-Action (VLA) systems such as $\pi_{0.5}$ and GR00T adopt hierarchical architectures in which a Vision-Language Model (VLM) backbone decomposes instructions into subtasks executed by an action expert. We hypothesise that the VLM backbone—an autoregressive, causal transformer—develops *temporally-abstract representations* of manipulation subtasks in its residual stream, and that these representations are embodiment-invariant: a human grasping a cup and a robot grasping a cup share the same reach–approach–close–lift boundary pattern despite vastly different kinematics. Building on this insight we propose TempoRAL, a three-phase framework: (1) fine-tune the VLM backbone on human manipulation demonstrations so that manipulation-specific temporal structure is encoded in its internal representations, then freeze; (2) train a self-supervised meta-controller on the frozen VLM backbone's residual stream to discover subtask boundaries without any annotation; and (3) apply Internal Reinforcement Learning in the discovered abstract-action space so that a causal policy can compose novel subtask sequences for new tasks under sparse rewards. A key ablation compares a base VLM (pretrained only) against a human-data-fine-tuned VLM, isolating the contribution of human demonstrations to temporal abstraction formation. We argue that this constitutes a novel form of *temporal abstraction transfer*—going beyond the representation-level cross-embodiment alignment reported in prior work to transfer the *temporal structure* of manipulation from humans to robots.**

I. INTRODUCTION

Recent Vision-Language-Action (VLA) models such as π_0 [1], $\pi_{0.5}$ [2], and GR00T N1 [24] have demonstrated that a single policy can generalise across diverse manipulation tasks when conditioned on natural-language instructions. These models share a common two-tier architecture: a *VLM backbone*—an autoregressive, causal transformer that processes visual observations and language instructions to form high-level plans—and an *action expert* that translates these plans into low-level motor commands via flow matching or diffusion. $\pi_{0.5}$ exemplifies this: its VLM first decomposes an instruction into subtasks ("pick up the cup", "place cup under the machine"), which an action expert executes at up to 50 Hz.

A central yet under-explored question is whether the VLM backbone develops *internal temporal abstractions*—structured representations of when one manipulation subtask ends and another begins—that can be discovered and leveraged without explicit annotation. Kobayashi et al. [5] recently showed that autoregressive models trained on next-token prediction spontaneously develop such temporal abstractions in their residual stream activations. A meta-controller—a small hypernetwork that reads from and writes to the residual stream—can discover these abstractions in a purely self-supervised manner: its switching gate β_t learns to fire at semantically meaningful boundaries (e.g. sub-goal transitions) without any boundary supervision. Moreover, performing RL directly in this abstract-action space ("Internal RL") vastly outperforms token-level RL under sparse rewards.

A critical prerequisite of this approach is that the base model must be **autoregressive and causal**—temporal abstractions emerge from sequential next-token prediction over long horizons. In VLA systems, this condition is satisfied by the **VLM backbone** (e.g. Eagle-2 for GR00T, PaliGemma for π_0), which autoregressively processes episode-length observation sequences. Conversely, the action expert uses *diffusion or flow matching* with *bidirectional attention* over short action chunks (0.1–1 s), and therefore cannot form temporal abstractions—it is an executor ("how to move"), not a planner ("what to do next"). This distinction is supported by empirical evidence: GR00T N1 found that *middle-layer* VLM embeddings outperform final-layer embeddings for downstream policy success [24], and $\pi_{0.5}$ already predicts subtask text explicitly within the VLM [2].

We further hypothesise that *human demonstration data* plays a decisive role in forming manipulation-relevant temporal abstractions within the VLM backbone. The temporal boundary structure of manipulation—when one primitive ends and the next begins—is largely **embodiment-invariant**: a human grasping a cup and a robot grasping a cup share the same reach→approach→close→lift boundary pattern, differing only in low-level kinematics. Physical Intelligence's analysis of $\pi_{0.5}$ [4] demonstrated that at sufficient scale, human and robot latent representations spontaneously merge in the VLM's embedding space. However, whether this alignment extends to *temporal abstraction*—not just static representations but the *dynamic switching structure* between subtasks—remains unverified.

We propose **TempoRAL**, a framework that:

1) **Fine-tunes** the VLM backbone on human manipulation demonstrations to encode manipulation-specific temporal structure, then **freezes** it (§III-A).
2) **Discovers** subtask boundaries via a self-supervised meta-controller trained on the frozen VLM backbone's residual stream—requiring no boundary annotations (§III-B).
3) **Optimises** subtask composition through Internal RL

[1]Anonymous Institution

in the low-dimensional controller-code space, enabling novel subtask sequences under sparse rewards (§III-C).
4) **Ablates** the contribution of human data by comparing a base VLM (pretrained only) against a human-data-fine-tuned VLM, providing the first quantitative evidence for cross-embodiment *temporal abstraction transfer* (§IV).

II. Related Work

A. Vision-Language-Action Models

The VLA paradigm unifies visual perception, language understanding, and motor control in a single model. RT-2 [6] first showed that a VLM can be fine-tuned to output robot actions as text tokens. π_0 [1] introduced a flow-matching action head, enabling 50 Hz continuous control alongside a frozen PaliGemma backbone. $\pi_{0.5}$ [2] extended this with a hierarchical subtask-then-action architecture and demonstrated open-world generalisation. π_0-FAST [3] replaced flow matching with FAST tokenisation, achieving 5× faster training. OpenVLA [7] and Octo [8] provide open-source alternatives. GR00T N1 [24] introduced per-embodiment MLP adapters around a shared Diffusion Transformer, enabling a single model to control diverse humanoid morphologies without zero-padding. All these systems treat the subtask–action interface as a fixed design choice; none exploit the VLM backbone's internal temporal representations to discover subtask boundaries from data.

B. Hierarchical Reinforcement Learning and Temporal Abstraction

The Options framework [15] formalised temporal abstraction in RL as policies with initiation sets, internal policies, and termination conditions. Option-Critic [16] learns options end-to-end. HIRO [17] and HAM [18] use goal-conditioned sub-policies. CompILE [19] discovers segments via variational inference. Most recently, *Internal RL* [5] was proposed showing that a meta-controller trained on a frozen autoregressive model's residual stream discovers temporal abstractions that align with ground-truth sub-goals—and that RL in the resulting abstract-action space vastly outperforms token-level RL under sparse rewards. Internal RL has only been demonstrated in gridworld and MuJoCo locomotion; its potential for VLM backbones in VLA systems and real-world robot manipulation is unexplored.

C. Learning Robot Policies from Human Data

Robot demonstration data is scarce ($\sim 10^5$ trajectories in Open X-Embodiment [9]) compared to human interaction data ($\sim 10^8$ samples in datasets like Ego4D [10], UniHand [11], Something-Something V2 [12]). LAPA [13] learns latent actions from human videos via a VQ-VAE and transfers them to robot control, outperforming OpenVLA while requiring 30× less compute. MT-π [14] uses motion tracks as a cross-embodiment action representation, achieving 86.5% real-world success. Being-H0 [11] trains a dexterous VLA on 165 M human hand samples. Physical Intelligence's own analysis of $\pi_{0.5}$ revealed an *emergent* alignment between human and robot representations at scale [4]: as model capacity increases, latent clusters for human hands and robot grippers spontaneously merge. Prior work transfers *low-level actions* or *visual features*; we transfer the *temporal abstraction structure*—discovered unsupervised from the VLM backbone's residual stream—which is more abstract and therefore more embodiment-invariant. Critically, while [4] showed that human and robot *representations* converge at scale, whether this alignment extends to *temporal abstractions* (the dynamic switching structure between subtasks) remains unverified.

Phase 1 (Fine-tune VLM θ, then freeze) → **Phase 2** (Train meta-controller ϕ; θ frozen) → **Phase 3** (Train RL policy ψ; θ, ϕ frozen)

Fig. 1: Training pipeline of TempoRAL. The VLM backbone is fine-tuned on human demonstrations and frozen; the meta-controller discovers subtask boundaries; Internal RL composes novel subtask sequences.

D. Subtask Decomposition and Task Planning

LLM-based planners such as SayCan [20], Inner Monologue [21], and Code-as-Policies [22] decompose instructions into executable steps. These approaches typically rely on a *fixed* affordance model or success detector to determine granularity. Voyager [23] learns a skill library but does not tie decomposition to executor capability. No existing method learns the decomposition granularity from the *planner's own internal temporal abstractions*.

III. TempoRAL Framework

TempoRAL consists of three sequential training phases (Figure 1). Throughout, we denote the VLM backbone's parameters as θ, the meta-controller's parameters as ϕ, and the Internal RL policy's parameters as ψ.

A. VLM Backbone Fine-Tuning on Human Demonstration Data

The goal of Phase 1 is to endow the VLM backbone with manipulation-specific temporal structure by fine-tuning it on human demonstration data, and then **freezing** it for all subsequent phases.

We target the VLM backbone rather than the action expert for a fundamental reason: temporal abstractions can only emerge in *autoregressive, causal* models that process long temporal sequences [5]. The VLM backbone (e.g. Eagle-2 for GR00T with 28 causal transformer layers, or PaliGemma for π_0 with 18 causal layers) satisfies this condition—it autoregressively processes episode-length observation-action sequences. In contrast, the action expert uses diffusion or flow matching with bidirectional attention over short action chunks (16–50 steps ≈ 0.1–1 s), which cannot form temporal abstractions spanning multi-second subtask transitions.

We argue that human manipulation data is a particularly effective fine-tuning source for three reasons.

a) Scale: Robot demonstration datasets contain on the order of 10^5 trajectories (e.g. Open X-Embodiment [9]: 970 K, BridgeV2: 60 K), collected via expensive teleoperation. Human manipulation datasets are orders of magnitude larger (Ego4D [10]: 3 670 hours; YouTube: effectively unbounded) and trivial to collect with commodity cameras.

b) Embodiment-invariant temporal structure: While low-level kinematics differ between a human hand (20+ DoF) and a parallel-jaw gripper (1 DoF), the *temporal boundary structure*—when one manipulation primitive ends and the next begins—is shared. Consider a "pick up cup" task: both human and robot execute `reach` → `pre-grasp` → `close` → `lift`, with transitions at kinematically analogous moments (contact initiation, force closure, vertical acceleration). This temporal structure is precisely what Phase 2's meta-controller will discover; fine-tuning on human data gives the VLM a richer prior over manipulation boundaries.

c) Emergent cross-embodiment alignment: Recent analysis of $\pi_{0.5}$ [4] demonstrated that at sufficient scale, VLA latent representations of human hands and robot grippers *spontaneously merge* in the VLM's embedding space. This provides direct empirical support for the viability of human-data fine-tuning.

The VLM backbone is a pretrained causal transformer (e.g. Eagle-2 with width $n_e = 2048$, depth 28; or PaliGemma with width $n_e = 2048$, depth 18). We fine-tune it on human manipulation demonstrations using next-token prediction over observation-action sequences:

$$\mathcal{L}(\theta) = \sum_{t=1}^{T} \Big[-\ln p_\theta(a_t \mid o_{1:t}) - \lambda \ln p_\theta(o_{t+1} \mid o_{1:t}) \Big] \quad (1)$$

where o_t are visual observations, a_t are actions, and $\lambda \geq 0$ weights an auxiliary observation-prediction loss that encourages world-model formation [5]. Fine-tuning is performed via LoRA [25] to keep the number of trainable parameters small (<1% of total), preserving the VLM's pretrained visual-language capabilities while injecting manipulation-specific temporal knowledge.

After this phase, θ **is frozen** for all subsequent phases. This is a critical design choice inherited from [5]: co-training the base model and meta-controller causes the temporal abstractions to collapse, as shown by the rate-distortion analysis in the original work.

To isolate the contribution of human data, we consider multiple conditions:

- **Condition A**: Base VLM (pretrained only, no fine-tuning) → freeze → Phase 2
- **Condition B**: VLM + human demonstration fine-tuning → freeze → Phase 2
- **Condition C** (optional): VLM + robot demonstration fine-tuning → freeze → Phase 2

Comparing A vs. B isolates whether human data enhances temporal abstraction formation in the VLM backbone. Comparing B vs. C tests whether human demonstrations produce temporal abstractions that are as effective as (or superior to) robot-specific data—the core of our cross-embodiment transfer hypothesis.

B. Meta-Controller Training (Self-Supervised)

The goal of this phase is to discover, from the frozen VLM backbone's internal representations, *where subtask boundaries naturally occur* —without any boundary annotations.

Given a demonstration trajectory $(o_{1:T}, a_{1:T})$, we perform a forward pass through the frozen VLM backbone and extract the residual-stream activations at a chosen layer l:

$$e_{t,l} = \text{ResidualStream}_\theta^{(l)}(o_{1:t}) \quad \in \mathbb{R}^{n_e} \quad (2)$$

where $n_e = 2048$ (the VLM hidden dimension). Based on the findings of [5] and the GR00T N1 middle-layer observation [24], we select l near mid-depth (e.g. layer 14 of 28 for Eagle, layer 9 of 18 for PaliGemma). Because the VLM backbone is a causal autoregressive transformer, it processes the full episode-length observation sequence—precisely the setting in which temporal abstractions have been shown to emerge [5].

The meta-controller ϕ consists of three components, following the architecture of [5]:

a) Encoder (bidirectional GRU): A recurrent encoder processes the *full* sequence of VLM residual-stream activations (non-causal access, using future information) to produce per-timestep latent statistics:

$$h_t = \text{BiGRU}_\phi(e_{1:T,l}) \quad (3)$$

$$\mu_t = W_\mu\, h_t + b_\mu \quad \in \mathbb{R}^{n_z} \quad (4)$$

$$\sigma_t^2 = \text{softplus}(W_\sigma\, h_t + b_\sigma) \quad \in \mathbb{R}^{n_z} \quad (5)$$

$$\tilde{z}_t \sim \mathcal{N}(\mu_t,\, \text{diag}(\sigma_t^2)) \quad (6)$$

where $n_z \ll n_e$ is the controller-code dimension (typically $n_z = 32$). Non-causal encoding during Phase 2 is justified by the variational information-theoretic argument of [5]: conditioning on the future allows the encoder to discover boundaries that *anticipate* upcoming transitions.

b) Switching Unit: A continuous gate $\beta_t \in [0, 1]$ determines whether to adopt a new controller code or persist with the previous one:

$$\beta_t = \sigma(W_\beta\, [e_{t,l};\, h_t;\, z_{t-1}] + b_\beta) \quad (7)$$

$$z_t = \beta_t \odot \tilde{z}_t + (1 - \beta_t) \odot z_{t-1} \quad (8)$$

When $\beta_t \approx 0$ the previous controller is maintained (same subtask); when $\beta_t \approx 1$ a fresh controller code is sampled (subtask boundary). A central finding of [5] is that β_t learns quasi-binary, sparse switching patterns *without explicit regularisation*, aligned with ground-truth sub-goal changes.

c) Decoder (hypernetwork): The controller code z_t is decoded into a low-rank linear controller $U_t \in \mathbb{R}^{n_e \times n_e}$ that modifies the residual stream via an additive intervention:

$$U_t = B_t A_t \quad (9)$$

$$B_t = f_B(z_t) \quad (B_t \in \mathbb{R}^{n_e \times r}) \quad (10)$$

$$A_t = f_A(z_t) \quad (A_t \in \mathbb{R}^{r \times n_e}) \quad (11)$$

$$e'_{t,l} = e_{t,l} + U_t e_{t,l} \quad (12)$$

where $r \ll n_e$ is the rank (typically $r = 32$) and f_B, f_A are learned linear maps. This factorisation keeps the number

of trainable parameters small ($\sim$2 M) relative to the frozen VLM backbone (1.7–2 B parameters).

The meta-controller is trained by minimising a variational lower bound on the action-prediction log-likelihood under the controlled VLM residual stream:

$$\mathcal{L}(\phi) = \sum_{t=1}^{T} \Big[\underbrace{-\ln p_{\theta,\phi}(a_t \mid o_{1:t}, z_{1:t})}_{\text{action prediction}} + \alpha \underbrace{D_{\mathrm{KL}}\big(\mathcal{N}(\mu_t, \sigma_t^2) \,\|\, \mathcal{N}(0, I)\big)}_{\text{prior regularisation}} \Big] \tag{13}$$

The hyperparameter α controls the rate–distortion trade-off: larger α pushes z_t toward the prior, producing coarser (more abstract) boundaries; smaller α permits finer segmentation.

d) Frozen VLM backbone is essential: If θ is co-trained with ϕ, the VLM learns to "absorb" the controller's influence, and β_t degenerates to uniform switching [5]. Freezing θ creates an information bottleneck that forces β_t to capture genuine temporal structure.

1) Emergent Subtask Boundaries: After training, β_t exhibits sparse, quasi-binary firing patterns that correspond to manipulation-phase transitions—without any boundary supervision. These boundaries reflect the VLM backbone's *internal* notion of "where one coherent behavioural plan ends and another begins", which is precisely the temporal abstraction structure we aim to discover and transfer across embodiments.

C. Internal RL for Novel Task Composition

Phase 2 discovers temporal boundaries in a self-supervised manner, but does not optimise them for *task success*. Phase 3 closes this loop by treating the entire VLA system (VLM backbone + meta-controller + action expert + environment) as the environment and applying RL in the abstract controller-code space.

We construct an "internal" MDP whose state and action spaces live inside the VLM backbone's representations:

$$\text{State:} \quad o_t^{\text{int}} = e_{t,l} \quad \text{(VLM residual stream at layer } l\text{)} \tag{14}$$

$$\text{Action:} \quad z_t \in \mathbb{R}^{n_z} \quad \text{(controller code)} \tag{15}$$

$$\text{Reward:} \quad r_t = \begin{cases} 1 & \text{if task completed successfully} \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

The "environment dynamics" are the composition of: (i) the decoder producing U_t from z_t, (ii) the frozen VLM backbone propagating the controlled residual stream to the action expert, (iii) the action expert generating motor commands, and (iv) the physical (or simulated) world executing those commands. From the RL agent's perspective, all of this is a black box that maps z_t to $(o_{t+1}^{\text{int}}, r_t)$.

During Phase 3 the continuous β_t is discretised via a Heaviside step function:

$$\beta_t^{\text{bin}} = H(\beta_t - \beta_{\text{thresh}}) \tag{17}$$

When $\beta_t^{\text{bin}} = 0$, the previous z_{t-1} is reused and the RL policy is *not queried*—the VLM backbone continues with the same abstract action. When $\beta_t^{\text{bin}} = 1$, the RL policy emits a new z_t. This achieves **temporal contraction**: if a trajectory of T primitive steps has M switch points ($M \ll T$), the RL policy makes only M decisions, drastically reducing the search space and improving credit assignment.

The non-causal BiGRU encoder from Phase 2 is replaced by a *causal* policy π_ψ that can only observe past and present:

$$z_t \sim \pi_\psi(z_t \mid e_{1:t,l}) \tag{18}$$

implemented as a causal GRU followed by a Gaussian policy head. The decoder (hypernetwork) weights from Phase 2 are reused and frozen; only ψ is trained.

We optimize ψ via policy gradient with relative advantage estimation:

$$\nabla_\psi J = \mathbb{E}\left[\sum_{m=1}^{M} A_m \, \nabla_\psi \ln \pi_\psi(z_{t_m} \mid e_{1:t_m,l}) \right] \tag{19}$$

where $\{t_1, \ldots, t_M\}$ are the switch points and A_m is the advantage at switch m. Because M is small (typically 3–8 per episode), the gradient variance is dramatically lower than token-level policy gradients, which is the key advantage of Internal RL over standard RL fine-tuning.

The RL policy learns to emit controller codes z_t that compose manipulation subtasks in novel sequences: each z_t steers the VLM backbone's planning representation through a coherent manipulation phase, and a switch is triggered precisely when a new subtask should begin. Because z_t operates in the VLM's abstract planning space rather than in the raw action space, novel combinations of subtasks—including sequences not seen during pretraining—become reachable through compositional generalisation in the controller-code space [5].

IV. PROPOSED EXPERIMENTS

We outline the experimental protocol designed to validate our hypothesis that human demonstration data induces transferable temporal abstractions in VLM backbones.

A. Experimental Setup

a) VLM backbones: We target two VLA architectures: (1) GR00T N1.6 with Eagle-2 VLM (Qwen3-1.7B, 28 layers, $n_e = 2048$), and (2) π_0 with PaliGemma VLM (Gemma 2B, 18 layers, $n_e = 2048$). For cost-effective validation, we also consider Qwen2.5-VL-7B (28 layers, $n_e = 3584$) with LoRA fine-tuning.

b) Human demonstration data: Egocentric manipulation videos with extracted end-effector trajectories via 3D hand tracking. Each trajectory is an observation-action sequence $(o_{1:T}, a_{1:T})$ where o_t is a camera frame and a_t is the hand/end-effector pose.

c) Meta-controller configuration: Controller-code dimension $n_z = 16$, hypernetwork rank $r = 32$, intervention at mid-depth (layer $l = L/2$), KL weight α swept over $[0.001, 0.1]$.

B. Ablation: Base VLM vs. Human-Fine-Tuned VLM

The central experiment compares three conditions on the same meta-controller training pipeline:

Condition	VLM state	Fine-tune data
A (Base)	Pretrained only	None
B (Human)	+ LoRA fine-tune	Human demonstrations
C (Robot)	+ LoRA fine-tune	Robot demonstrations

TABLE I: Ablation conditions for isolating the contribution of human data to temporal abstraction formation.

a) Metric 1: Linear probing accuracy: For each condition, we train linear classifiers on the frozen VLM's residual stream at each layer to decode the current ground-truth subtask label. Higher mid-layer probing accuracy indicates richer temporal abstraction representations [5].

b) Metric 2: β_t alignment with subtask boundaries: We measure the alignment between the meta-controller's switching gate β_t and ground-truth subtask transition times using normalised mutual information (NMI). Sparse, quasi-binary β_t patterns aligned with true transitions indicate successful temporal abstraction discovery.

c) Metric 3: Internal RL success rate: For post-training tasks—novel subtask sequences not seen during pre-training or meta-controller training—we compare the success rate of Internal RL across conditions A, B, and C. Following [5], we also compare against raw-action RL (GRPO) and Internal RL without temporal abstraction ($\beta_t = 1$ forced) as baselines.

C. Expected Outcomes

- $\mathbf{A} < \mathbf{B}$: Human fine-tuning significantly improves temporal abstraction quality, validating that human data encodes manipulation-relevant temporal structure in the VLM backbone.
- $\mathbf{B} \approx \mathbf{C}$: Human and robot demonstrations produce comparable temporal abstractions, providing strong evidence for *cross-embodiment temporal abstraction transfer*.
- $\mathbf{B} > \mathbf{C}$: If human data yields *superior* temporal abstractions (due to greater data diversity or more natural manipulation structure), this would constitute a particularly strong contribution.

Any of these outcomes provides novel empirical evidence beyond the representation-level cross-embodiment alignment reported in [4], extending it to the *temporal abstraction* level.

V. CONCLUSIONS

We presented **TempoRAL**, a framework that discovers and transfers temporal abstractions from human demonstrations to robot manipulation via the internal representations of VLM backbones. Our approach rests on three insights: (1) the VLM backbone—an autoregressive, causal transformer—is the correct site for temporal abstraction discovery in VLA systems, as it satisfies the prerequisites identified by [5] (unlike the diffusion-based action expert); (2) a meta-controller trained on the frozen VLM backbone's residual stream discovers subtask boundaries without supervision; and (3) Internal RL in the resulting abstract-action space enables compositional generalisation to novel subtask sequences under sparse rewards.

The central hypothesis of this work is that human demonstration data induces manipulation-specific temporal abstractions in the VLM backbone that are *embodiment-invariant*—extending beyond the representation-level cross-embodiment alignment reported in prior work [4] to the *temporal structure* of manipulation. Our proposed ablation (base VLM vs. human-fine-tuned VLM vs. robot-fine-tuned VLM) is designed to provide the first quantitative evidence for this claim.

a) Future work: Several directions remain. First, the meta-controller's β_t switching signal could serve as a *runtime re-planning trigger*: when β_t fires unexpectedly during deployment, it signals that the current subtask plan has become invalid, enabling the VLM to generate a revised decomposition. Second, the learned temporal abstraction structure could be distilled into a capability-aware prompting prior that guides the VLM to generate subtasks at the appropriate granularity without manual prompt engineering. Finally, scaling to additional embodiments (e.g. quadrupeds, dexterous hands) and validating on diverse hardware platforms is essential to confirm the generality of cross-embodiment temporal abstraction transfer.

APPENDIX

Appendixes should appear before the acknowledgment.

ACKNOWLEDGMENT

The preferred spelling of the word ÒacknowledgmentÓ in America is without an ÒeÓ after the ÒgÓ. Avoid the stilted expression, ÒOne of us (R. B. G.) thanks . . .Ó Instead, try ÒR. B. G. thanksÓ. Put sponsor acknowledgments in the unnumbered footnote on the first page.

References are important to the reader; therefore, each citation must be complete and correct. If at all possible, references should be commonly available publications.

REFERENCES

[1] Physical Intelligence. π_0: A Vision-Language-Action Flow Model for General Robot Control. *arXiv:2410.24164*, 2024.
[2] Physical Intelligence. $\pi_{0.5}$: A Vision-Language-Action Model with Open-World Generalization. 2025.
[3] Physical Intelligence. π_0-FAST: Efficient Action Tokenization for Vision-Language-Action Models. 2025.
[4] Physical Intelligence. Emergent Cross-Embodiment Alignment in Scaled VLA Models. Blog post, 2025.
[5] S. Kobayashi et al. Emergent temporal abstractions in autoregressive models enable hierarchical reinforcement learning. *arXiv:2512.20605*, 2025.
[6] A. Brohan et al. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. *arXiv:2307.15818*, 2023.
[7] M. Kim et al. OpenVLA: An Open-Source Vision-Language-Action Model. *arXiv:2406.09246*, 2024.
[8] Octo Model Team. Octo: An Open-Source Generalist Robot Policy. *arXiv:2405.12213*, 2024.
[9] Open X-Embodiment Collaboration. Open X-Embodiment: Robotic Learning Datasets and RT-X Models. *arXiv:2310.08864*, 2023.
[10] K. Grauman et al. Ego4D: Around the World in 3,000 Hours of Egocentric Video. *CVPR*, 2022.

[11] Being-H0 Team. Being-H0: Vision-Language-Action Pretraining from Large-Scale Human Videos. *arXiv:2507.15597*, 2025.
[12] R. Goyal et al. The "Something Something" Video Database for Learning and Evaluating Visual Common Sense. *ICCV*, 2017.
[13] S. Edwards et al. Latent Action Pretraining from Videos. *arXiv:2410.11758*, 2024.
[14] Cornell Team. Motion Tracks: A Unified Representation for Human-Robot Transfer in Few-Shot Imitation Learning. 2025.
[15] R. Sutton, D. Precup, and S. Singh. Between MDPs and semi-MDPs: A Framework for Temporal Abstraction in Reinforcement Learning. *Artificial Intelligence*, 112(1–2):181–211, 1999.
[16] P.-L. Bacon, J. Harb, and D. Precup. The Option-Critic Architecture. *AAAI*, 2017.
[17] O. Nachum et al. Data-Efficient Hierarchical Reinforcement Learning. *NeurIPS*, 2018.
[18] R. Parr and S. Russell. Reinforcement Learning with Hierarchies of Machines. *NeurIPS*, 1998.
[19] T. Kipf et al. CompILE: Compositional Imitation Learning and Execution. *ICML*, 2019.
[20] M. Ahn et al. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. *arXiv:2204.01691*, 2022.
[21] W. Huang et al. Inner Monologue: Embodied Reasoning through Planning with Language Models. *CoRL*, 2023.
[22] J. Liang et al. Code as Policies: Language Model Programs for Embodied Control. *ICRA*, 2023.
[23] G. Wang et al. Voyager: An Open-Ended Embodied Agent with Large Language Models. *arXiv:2305.16291*, 2023.
[24] NVIDIA. GR00T N1: An Open Foundation Model for Generalist Humanoid Robots. *arXiv:2503.14734*, 2025.
[25] E. J. Hu et al. LoRA: Low-Rank Adaptation of Large Language Models. *ICLR*, 2022.