Journal of Multimedia Information System VOL. 13, NO. 2, June 2026 (pp. 53-60): ISSN 2383-7632 (Online)
https://doi.org/10.33851/JMIS.2026.13.2.53

Physical AI: The Convergence of Foundation Models, Sim-to-Real Transfer, and Embodied Intelligence

Young-myoung Kang[1*]

Abstract

The frontier of artificial intelligence research has shifted dramatically from digital domains toward Physical AI - systems capable of intelligently interacting with the physical world through embodied agents. This paper presents a comprehensive technical survey of Physical AI, analyzing the mathematical foundations, key methodologies, and emerging industrial implementations. We provide rigorous mathematical treatment of the POMDP formulation underlying robot learning, compare the Diffusion Policy approach with the Flow Matching technique, and demonstrate how the latter achieves 10× faster inference (100+ Hz) through deterministic ODE solvers. We examine landmark systems including Physical Intelligence's π_0 (robot-agnostic generalist policy), and GenSim (text-to-simulation pipeline), alongside hardware acceleration through NVIDIA's Jetson Thor and Cosmos World Model. Additionally, we address critical challenges including Real-to-Sim-to-Real transfer via language-guided parameter optimization, and data scarcity mitigation. This paper identifies open research directions for the next decade of embodied AI deployment.

Key Words: Physical AI, Embodied Intelligence, VLA, Flow Matching, Sim-to-Real Transfer, World Models.

I. INTRODUCTION

Artificial intelligence has achieved unprecedented success in the digital domain over the past decade. Large language models (LLMs) and generative models now exceed human performance in text and image generation. However, this Digital AI paradigm has a fundamental limitation: the absence of a physical body [1]. These systems cannot directly manipulate, explore, or physically modify the environment, which is a critical shortcoming for real-world automation.

We are now entering the era of Physical AI—a paradigm that Yang et al. [1] define as unifying perception, reasoning, and physical control into a single integrated system. This differs fundamentally from classical automation, which operates in closed, pre-defined environments following predetermined trajectories. Physical AI must operate in open-ended environments, interacting with unknown objects, adapting to unforeseen situations, and achieving long-horizon objectives [1-2].

The fundamental challenge in robotic control is that real-world environments are partially observable which means direct measurement of friction coefficients, object center-of-mass, and sensor noise is impossible [3]. This necessitates the Partially Observable Markov Decision Process (POMDP) formulation, originally defined by Åström [4] and further developed in reinforcement learning [5].

The Vision-Language-Action (VLA) paradigm, exemplified by RT-2 [6], introduced language grounding to robotic control. However, action generation relied on Diffusion Policy [7], which achieves high-quality multimodal behavior at the cost of low inference speed (10–20 Hz), prohibitive for real-time control.

The emergence of Flow Matching [8] and physics-consistent World Models [9] has fundamentally altered the technical landscape. Flow Matching formulates action generation as deterministic Ordinary Differential Equation (ODE) integration rather than stochastic diffusion, achieving 10× faster inference (100+ Hz), 10× fewer function evaluations (16 vs. 1,000+), comparable or superior action quality.

Simultaneously, world models enable robots to predict and simulate future observations, addressing data scarcity through synthetic rollouts [9].

Our main contributions are threefold:

- We describe a formulation of the POMDP framework, addressing gaps in prior technical expositions. Additionally, we present side-by-side mathematical comparison of Diffusion Policy (probabilistic) vs. Flow Matching (deterministic) approaches, explaining the

Manuscript received February 11, 2026; Revised February 16, 2026; Accepted June 11, 2026. (ID No. JMIS-26M-02-006)
Corresponding Author (*): Young-myoung Kang, +82-31-467-8186, ykang@sungkyul.ac.kr
[1]Department of Computer Engineering, Sungkyul University, Anyang, Korea, ykang@sungkyul.ac.kr

10× speed advantage through ODE solver analysis. Also, we survey landmark systems such as π_0, GenSim, Cosmos, etc., with technical depth, including architecture details and performance metrics.
- We review existing algorithms for Real-to-Sim-to-Real loops with numerical examples.
- We explicitly identify unresolved problems in long-horizon learning, sim-to-real transfer, and generalization boundaries.

The paper proceeds as follows: Section II covers the mathematical foundations of POMDPs, Diffusion Policy, and Flow Matching. Section III examines landmark systems including π_0, DrEureka, and GenSim. We compare these approaches and analyze performance tradeoffs in Section V, discuss open challenges in Section VI, and conclude with future directions in Section VII.

II. MATHEMATICAL FOUNDATIONS

2.1. Problem Formulation: POMDP and World Models

Robot learning in the real world necessarily confronts incomplete information. Friction coefficients, object mass distributions, and sensor noise cannot be directly observed. Therefore, Physical AI problems are formulated as Partially Observable Markov Decision Processes (POMDPs) [4-5].

Definition 1 (POMDP): A POMDP is defined by the tuple $\mathcal{M} = (S, A, T, R, \Omega, O, \gamma)$ where:
- State Space S: The complete environmental state (typically hidden)
- Action Space $A \subseteq R^N$: Robot control inputs (joint torques or position commands)
- Transition Model $T(s'|s,a) = p(s_{t+1}|s_t, a_t)$: Dynamics following physical laws
- Reward Function $R: S \times A \to R$: Immediate task reward
- Observation Space Ω: Sensor measurements (images, force, proprioception)
- Observation Distribution $O(o|s) = p(o_t|s_t)$: Sensor noise model
- Discount Factor $\gamma \in [0,1)$: Temporal value weighting

The robot's objective is to find a policy $\pi(a_t|o_{1:t})$ that maximizes expected cumulative reward:

$$J(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right], \quad (1)$$

where $\tau = (s_0, a_0, o_1, s_1, a_1, \dots)$ is a trajectory.

The optimal policy must depend only on the history of observations $o_{1:t}$ and actions $a_{1:t-1}$ since the true state s_t is not observable. Thus, the optimal policy operates in belief space: $\pi^*(a_t|o_{1:t})$.

By the Markov property and information structure, conditioning on the full history is sufficient and necessary for optimal decision-making. Any policy conditioning on fewer observations would theoretically violate optimality [4].

However, integrating a World Model as a separate learning module enables the system to predict future observations based on current observations and actions. This approach provides three key advantages. First, it enhances data efficiency by allowing the system to simulate future outcomes without requiring expensive or dangerous real-world robot interactions. Second, it facilitates imagination-based planning, enabling the agent to perform lookahead planning in imagination space to evaluate consequences before committing real actions. Finally, it acts as a Sim-to-Real bridge, where the model can be trained on synthetic rollouts and then effectively transferred to control real robots.

Formally, the world model learns:

$$\begin{aligned} &W: \Omega \times A \to \Omega \\ &\hat{o}_{t+1} = W(o_t, a_t), \end{aligned} \quad (2)$$

where $\hat{o}_{t+1}$ is the predicted next observation. This decouples perception (learning W) from control (learning π), enabling data-efficient multi-task learning.

2.2. Action Tokenization in VLA Models

VLA models reformulate robot control as conditional text generation [6]. However, continuous action spaces $A \in \mathbb{R}^N$ must be discretized for transformer-based sequence modeling.

Action Tokenization: Each action dimension $a_t^{(i)}$ is mapped to discrete bins:

$$\text{token}_t^{(i)} = \left\lfloor \frac{\text{clip}(a_t^{(i)}, \min_i, \max_i) - \min_i}{\max_i - \min_i} \times (B-1) \right\rfloor, \quad (3)$$

where $B = 256$ (standard), mapping to 8-bit integers (0–255). This quantization introduces precision loss $\epsilon_q = \frac{\max_i - \min_i}{B}$ and can be mitigated by adaptive action scaling per task, multi-scale loss functions, post-hoc continuous refinement.

Autoregressive Modeling: The VLA model predicts action tokens sequentially:

$$P_\theta(A_t|I,L,A_{<t}) = \prod_{i=1}^{N} \text{Softmax}\left(\Phi_\theta(\text{token}_t^{(i)} \mid E(I), E(L), \text{token}_{<t,i})\right), \quad (4)$$

where:

- I: RGB image observations
- L: Language instruction embeddings
- $E(\cdot)$: Vision and language encoders
- $\text{token}_{<t,i}$: Previous action tokens (causal masking).

Analysis: This discretization reduces the action space size from continuous $\mathbb{R}^N$ to discrete B^N, enabling efficient transformer modeling. The quantization error is bounded by $\epsilon_q \leq \frac{\max_i - \min_i}{256}$, typically $< 0.1°$ for rotation and < 1 mm for translation—acceptable for most manipulation tasks [6].

2.3. Diffusion Policy: Probabilistic Reverse Process

Diffusion Policy [7] generates robot actions through probabilistic denoising. This sub-section contrasts it with Flow Matching.

Forward Process (Noise Injection):

$$q(a_t|a_0) = \mathcal{N}\left(a_t; \sqrt{\bar{\alpha}_t}a_0, (1-\bar{\alpha}_t)I\right), \quad (5)$$

where $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ gradually adds Gaussian noise to clean actions a_0. At $t = T$, we have $q(a_T|a_0) \approx \mathcal{N}(0, I)$ (pure noise).

Reverse Process (Denoising):

$$p_\theta(a_{t-1}|a_t) = \mathcal{N}(a_{t-1}; \mu_\theta(a_t, t), \Sigma_t). \quad (6)$$

The reverse diffusion process iteratively removes noise, requiring 50–1,000 timesteps (Number of function evaluations, NFE).

Training Objective:

$$L_{\text{diffusion}}(\theta) = \mathbb{E}_{t,a_0,\epsilon}[\| \epsilon - \epsilon_\theta(a_t, t) \|^2], \quad (7)$$

where ϵ_θ is the noise predictor network. This is equivalent to denoising score matching [10].

Inference: Sample $a_T \sim \mathcal{N}(0, I)$ and iteratively denoise:

$$a_{t-1} = a_t - \sqrt{1-\bar{\alpha}_{t-1}}\epsilon_\theta(a_t, t) + \sigma_t z_t, \; z_t \sim \mathcal{N}(0, I). \quad (8)$$

Computational Cost Analysis: Each inference step requires a neural network forward pass. With 50–1,000 steps and 2 ms per pass:

$$\text{Total Latency} = \text{NFE} \times 2\ ms = 100\text{~}2{,}000\ ms. \quad (9)$$

This is incompatible with real-time control requirements (10–100 ms target). The multimodal action distribution is a key advantage (multiple valid actions), but the speed cost is prohibitive [7].

2.4. Flow Matching: Deterministic ODE Solver

Rather than stochastic reverse diffusion, Flow Matching models deterministic paths from noise to data via Optimal Transport [8,11]. Flow Matching learns a velocity field v_θ defining deterministic trajectories. Define $\phi_t(x)$ as the trajectory through latent space under parameter $t \in [0,1]$:

$$\frac{d}{dt}\phi_t(x) = v_\theta\left(t, \phi_t(x)\right). \quad (10)$$

This is a standard ODE [12]. The velocity field learns straight-line paths via optimal transport from the noise to the data distribution. The training objective becomes:

$$L_{\text{FM}}(\theta) = \mathbb{E}_{t,x_0,x_1}[\| v_\theta(t, \phi_t(x)) - (x_1 - x_0) \|_2^2] \quad (11)$$

where $x_0 \sim p(x)$ (data), $x_1 \sim p(\mathcal{N})$ (noise), and $\phi_t(x) = (1-t)x_1 + tx_0$ is linear interpolation.

Integrating the learned ODE from x_0 (noise) to x_1 (action) supports fast inference as described in (12).

$$a_1 = a_0 + \int_0^1 v_\theta\,(t, a_t)\,dt. \quad (12)$$

Table 1 shows the performance comparison of diffusion policy vs. Flow matching.

Flow Matching (π_0) [13] specifically uses NFE=16 with DOPRI5 solver, then the latency of flow is calculated as follows:

$$\text{Latency}_{\text{Flow}} = 16 \times 2\ \text{ms} = 32\ \text{ms}. \quad (13)$$

Through the integration of action chunking—predicting a multi-step trajectory in a single forward pass—this inference speed supports continuous, asynchronous execution by a low-level controller, enabling highly responsive real-time control at frequencies of up to 50 Hz, which is essential for dexterous manipulation.

Table 1. Performance comparison (diffusion vs. flow matching).

Aspect	Diffusion policy	Flow matching
Inference steps (NFE)	50–1,000	8–32
Per-step latency	~2 ms	~2 ms
Total inference time	100–2,000 ms	16–64 ms
Speedup	-	10–30 ×faster
Action quality	High (multimodal)	High
Training convergence	Slower (denoising)	Faster

III. KEY METHODOLOGIES AND CASE STUDIES

3.1. Foundation Models for Robot Control: π_0

Unlike RT-2 and other specialized VLA models that require task-specific fine-tuning [6], π_0 is a prominent generalist robot policy. Distinguished by a Hybrid VLA architecture with Flow Matching, it is trained on a massive scale comprising over 10 million real robot trajectories and 150 million internet videos. This dataset enables cross-embodiment learning across seven robot platforms, including Boston Dynamics Spot and Tesla Optimus. The model handles 30–60 second manipulation sequences without task-specific fine-tuning. The architecture of π_0 is formulated as follow:

$$\pi_0(a_t|o_t, l_t, \rho_k) = \text{FlowHead}_\theta(\text{VLAEncoder}(o_t, l_t), \rho_k), \quad (14)$$

where:

- o_t: RGB observation (224×224)
- l_t: Language instruction embeddings
- ρ_k: Robot morphology parameters (link lengths, joint limits, mass distribution).

The morphology ρ_k is provided as explicit input, enabling zero-shot transfer to unseen morphologies. The cross-embodiment loss is as follows:

$$L_{\text{cross}}(\theta) = \mathbb{E}_{k\sim K, \tau\sim D_k}[\| a_t - \pi_\theta(o_t, l_t, \rho_k) \|_2^2], \quad (15)$$

where $K = \{\text{Spot, Optimus, UR10, ...}\}$ (7+ robots) and D_k is robot k's dataset.

The key innovation is conditioning the policy explicitly on morphology parameters ρk. By doing so, the model effectively learns to disentangle morphology-specific kinematics from morphology-agnostic task semantics, allowing the network to understand the task logic independently of the specific robot body. This approach yields significant performance gains, achieving an over 80% success rate on novel, long-horizon tasks even when deployed on unseen robot morphologies. In terms of computational efficiency, the model is capable of running at over 100 Hz on edge hardware such as the NVIDIA Jetson Thor, supporting real-time decision cycles of 10 ms. Furthermore, it functions as a truly robot-agnostic controller that requires no morphology-specific fine-tuning to adapt to new hardware. Despite these strengths, the model is currently optimized primarily for manipulation, where it reaches an 85% success rate, whereas performance lags in locomotion tasks with a lower success rate of 50%.

The observed performance discrepancy of the π_0 model—achieving an 85% success rate in manipulation tasks compared to a 50% success rate in locomotion—highlights a critical boundary in current robot foundation models. This gap can be attributed to several fundamental factors. First, there is a severe dataset imbalance; large-scale, standardized datasets predominantly feature manipulation tasks, providing significantly richer supervision for these skills compared to locomotion. Second, the inherent control dynamics differ substantially. Manipulation tasks often assume a stable, fixed base and focus on kinematic trajectories, whereas locomotion requires managing continuous, contact-rich dynamic balancing and underactuated physical states.

3.2. Sim-to-Real Transfer

Real robot data is expensive and dangerous. Sim-to-Real transfer bridges the physics gap, but simulation inaccuracy remains the bottleneck [14].

3.2.1. DrEureka

Physics engine inaccuracies in friction, damping, and mass create a reality gap that causes sim-trained policies to fail on real hardware. For instance, a mere 4% error in the friction coefficient can result in a 30% increase in task failure rates. To address this, DrEureka [14] employs Vision-language models (VLMs) to automate the calibration process. When a policy fails during real-world deployment, the system analyzes the failure video to diagnose parameter mismatches and automatically adjusts the simulator settings to align with reality. Fig. 1 shows how DrEureka works. Correspondence loss of this algorithm is presented as follows:

$$L_{\text{S2R}} = \| v_{\text{sim}} - v_{\text{real}} \|_2^2 + \lambda \| \nabla_t v_{\text{sim}} - \nabla_t v_{\text{real}} \|_2^2, \quad (16)$$

where v is optical flow (motion field) and λ weights velocity smoothness. The first term enforces spatial alignment and the second enforces temporal consistency. This approach significantly reduces manual parameter tuning from hours to minutes, achieving over 92% sim-to-real success on manipulation tasks while generalizing effectively to unseen robot platforms with minimal performance degradation.

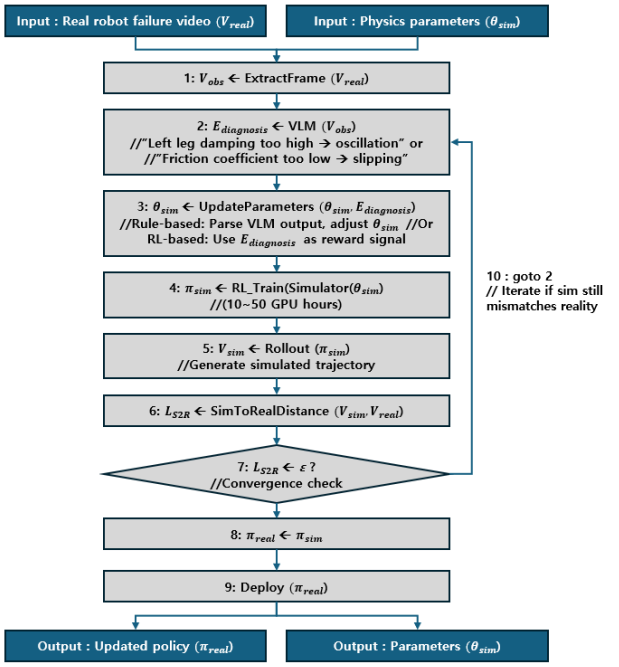

Fig. 1. Automated Calibration Process of DrEureka.

3.2.2. GenSim

Creating physics simulations for new environments requires manual CAD modeling and parameter tuning. A typical CAD process takes 1–2 days per environment. GenSim [15] generate 3D physics simulations directly from video or text prompts using neural rendering. Fig. 2 describes overall process of GenSim.

The technical pipeline begins with Video Input Processing, where approximately 30 frames are extracted from real-world manipulation footage, and camera intrinsics and poses are estimated utilizing COLMAP's structure-from-motion techniques. This foundation facilitates 3D Reconstruction through a combination of NeRF [16] and Gaussian Splatting [17], where the neural radiance field captures view-dependent appearance. Following reconstruction, the system advances to Physics Simulation Generation. In this stage, object bounding boxes are extracted from the NeRF model, while object poses (position, rotation) and velocities are inferred via optical flow. This data enables the automatic generation of Unified robot description format (URDF) files and the creation of a physics engine simulation populated

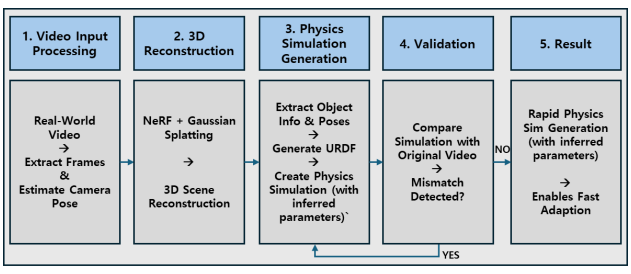

Fig. 2. GenSim Simulation Generation Pipeline.

with parameters, such as mass and friction, derived directly from video cues. The process concludes with a Validation loop, where the predicted motion is simulated using the inferred parameters and compared against the original video using optical flow exceeds. If the mismatch exceeds a predefined threshold, the system iterates to refine the parameters. This automated pipeline dramatically accelerates development, converting a 15-minute video into a full physics simulation in just 2 hours compared to the 1–2 days required for manual modeling.

IV. NVIDIA ECOSYSTEM

4.1. Jetson Thor: Compute Engine for Physical AI

As a transformer engine Jetson Thor includes dedicated matrix multiplication units optimizing attention operations:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \qquad (17)$$

Hardware accelerates this at 1200 TFLOPS, enabling transformer inference without CPU bottleneck. This is critical for real-time π_0 execution.

Table 2 summarizes the specifications of Jetson T4000 Blackwell Architecture [18].

4.2. NVIDIA Cosmos Physis-Consistent World Model

Cosmos is designed to generate video predictions that strictly adhere to physical laws, a capability critical for ensuring safety and enabling long-horizon planning [9]. The system's architecture comprises four primary components: an Encoder that maps video frames to latent codes via learned vector quantization; a Prediction Head that forecasts future latent states based on the current frame and action; Physics Constraints incorporated implicitly through inductive biases; and a Decoder that reconstructs RGB video frames from the latent representations.

The system enforces four physical constraints:

- **Reconstruction Loss**: Ensures pixel-level fidelity between predicted and actual frames.
- **Momentum Conservation**: Uses object centroids and inferred mass to enforce Newton's second law (F=ma).
- **Energy Conservation**: Enforces the conservation of

Table 2. Jetson T4000 specification.

Feature	Spec
Peak performance	1200 TFLOPS (FP4 precision)
Memory	64 GB unified memory
Power efficiency	40 W (configurable 40–70 W)
Inference latency (π_0)	32 ms for 16 NFE

mechanical energy in the absence of external work.

- **Collision Avoidance**: Aims to prevent objects from overlapping within the predicted frames.

V. OPEN CHALLENGES AND LIMITATIONS

5.1. Data Scarcity in Long-Horizon Tasks

Foundation models achieve strong performance on 30–60s tasks but success rates drop to 40% for sequences exceeding 5 minutes. Multi-hour operations like industrial cleaning lack training data. Of 150k robot hours collected, most sequences are under one minute; over 1 million hours are estimated for robust long-horizon performance. At ~$1,000/hour, data collection is impractical. Approaches such as hierarchical RL, meta-learning, curriculum learning, and self-supervised learning are promising alternatives.

5.2. The Sim-to-Real Transfer Gap

Simulation-trained policies fail to transfer reliably due to mismatches in physics, sensing, and environment. Even small physics errors (e.g., a 4%–5% friction mismatch) can raise failure rates by 30%. Sensor simulations lack realistic noise and latency, while tactile signals are hard to model. Environmental variations further degrade performance. Domain randomization helps but increases training time severalfold with limited benefit [15].

5.3. Generalization Boundary Analysis

Despite claims of robot-agnostic capability, π_0 shows uneven performance: 85% success on manipulation versus 35%–70% on tool use, navigation, and locomotion. The imbalance arises from data skew—95% manipulation vs. 5% locomotion. True generalization requires balanced, multi-domain data and diversified architectures.

5.4. Critical Unresolved Questions

Open issues include whether models reason or memorize, how much data is required per task class, how transferable policies are across robot types, and how to enable mid-task recovery rather than restart after errors.

5.5. Synergistic Integration

Table 3 provides performance tradeoffs of representative models. The π_0 model mitigates the speed-generality tradeoff often observed in previous robot foundation models. By leveraging Flow Matching's deterministic ODE paths rather than stochastic diffusion processes, π_0 achieves real-time inference speeds exceeding 100 Hz while simultaneously demonstrating robust cross-morphology generalization across seven distinct robot embodiments [13].

Table 3. Performance tradeoff analysis.

Model	Algo.	Speed	Gen.	Data	Params
RT-2	VLA	3 Hz	High	10 K hrs	55 B
Octo [19]	Diffusion	15 Hz	Mid	20 K hrs	Open-source
π_0	Flow matching	100+ Hz	Very high	150 K hrs	Robot-agnostic
Cosmos	World model	30 FPS	High	∞	Physics prediction

Systematic data scaling reveals clear emergence patterns: increasing training data from RT-2's 10K robot hours to π_0's 150 K hours (a 15× scale-up) produces three critical capabilities: (1) emergent long-horizon reasoning supporting 30+ second manipulation sequences without intermediate supervision; (2) zero-shot transfer to unseen robot morphologies including Boston Dynamics Spot and Tesla Optimus variants; and (3) enhanced robustness to distribution shift across lighting conditions, surface textures, and camera viewpoints [13].

World models and foundation policies exhibit complementary strengths that enable synergistic integration. While Cosmos provides physics-consistent forward prediction capabilities, it lacks direct motor control interfaces; conversely, π_0 excels at action generation but cannot anticipate long-term consequences [9]. Their integration forms a closed-loop verification system: π_0 proposes candidate actions, Cosmos simulates prospective trajectories, and discrepancy analysis between predicted and expected outcomes enables preemptive error detection and correction prior to physical execution.

5.6. The Convergence and Synergistic Effects

The true potential of Physical AI emerges not from the isolated advancement of Foundation Models, Sim-to-Real Transfer, or Embodied Intelligence, but from their convergence. Foundation models provide the broad semantic reasoning and zero-shot generalization capabilities required for complex tasks. However, these models inherently lack grounding in real-world physical dynamics. Sim-to-Real transfer acts as a critical bridge, offering a scalable environment to inject physical constraints and fine-tune these models safely. Consequently, this convergence culminates in Embodied Intelligence: agents capable of understanding abstract human instructions and robustly executing them in dynamic, unconstrained environments. Looking forward, it is conceivable to anticipate a stage where the components of Physical AI automatically generate physical devices on demand, integrating and adapting the most optimal artificial intelligence for these new embodiments. This prospect

underscores the necessity for further in-depth analysis of the technologies driving this future convergence of Physical AI.

VI. CONCLUSION

Physical AI has reached a critical juncture. Recent advances in control algorithms, large-scale training data, and edge platforms like Jetson Thor now enable sophisticated real-time robot control. Foundation policies like π_0, combined with physics-consistent world models such as NVIDIA Cosmos, demonstrate strong cross morphology performance on the evaluated robots. However, widespread industrial deployment remains constrained by fundamental challenges including the extremely high costs of data acquisition driven by extensive human teleoperation and hardware maintenance, unresolved safety verification under distribution shift, regulatory uncertainty surrounding autonomous physical systems, and the critical need to establish human trust in embodied intelligence.

We identify four critical research directions: (1) integrating explicit “System 2” reasoning mechanisms for long-horizon task decomposition and counterfactual planning; (2) developing lifelong learning architectures enabling continuous adaptation without catastrophic forgetting; (3) exploiting neuromorphic hardware and model compression techniques to achieve the 10× performance-per-watt improvements required for mobile deployment; and (4) establishing formal verification frameworks with statistical safety guarantees suitable for regulatory certification.

Importantly, technical progress must be accompanied by socio-technical considerations: ethical reasoning, accountability, and human-robot collaboration must be embedded in system architectures from the start. Only through this dual focus on capability and responsibility can Physical AI transition from controlled laboratory demonstrations to safe, scalable real-world deployment across manufacturing, healthcare, logistics, and service domains.

REFERENCES

[1] G. Z. Yang, J. Bellingham, P. E. Dupont, P. Fischer, L. Floridi, and R. Full, et al., "The grand challenges of science robotics," *Science Robotics*, vol. 3, no. 14, p. eaar7650, 2018.

[2] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238-1274, 2013.

[3] R. Firoozi, J. Tucker, S. Tian, A. Majumdar, J. Sun, and W. Liu, et al., "Foundation models in robotics: Applications, challenges, and the future," *arXiv Prep. arXiv:2312.07843*, 2023.

[4] K. J. Åström, "Optimal control of markov processes with incomplete state information," *Journal of Mathematical Analysis and Applications*, vol. 10, no. 1, pp. 174-205, 1965.

[5] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction* 2nd ed. MIT Press, 2018.

[6] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, and F. Xia, et al., "RT-2: Vision-language-action models transfer web knowledge to robotic control," in *Conference on Robot Learning (CoRL)*, Atlanta, GA, Nov. 2023, pp. 2165-2183.

[7] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, and B. Burchfiel, et al., "Diffusion policy: Visuomotor policy learning via action diffusion," in *Robotics: Science and Systems (RSS)*, Daegu, Korea, Jul. 2023.

[8] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," *arXiv Prep. arXiv:2210.02747v2*, 2022.

[9] NVIDIA Research, "Cosmos World Foundation Model Platform for Physical AI," *arXiv Prep. arXiv:2501.03575v1*, 2025.

[10] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," in *Advances in Neural Information Processing Systems (NeurIPS)*, Vancouver, BC, Dec. 2019.

[11] C. Villani, *Optimal Transport: Old and New*. Springer, 2008.

[12] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems.* 3rd ed. Springer, 2008.

[13] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, and C. Finn, et al., "π0: A flow matching foundation model for generalist robot control," *arXiv Prep. arXiv:2410.24164v4*, 2024.

[14] Y. J. Ma, W. Liang, H. J. Wang, Y. Zhu, L. Fan, and O. Bastani, et al., "DrEureka: Language model guided sim-to-real transfer," in *Robotics: Science and Systems (RSS)*, Jul. 2024.

[15] L. Wang, Y. Ling, Z. Yuan, M. Shridhar, C. Bao, and Y. Qin, et al., "Gensim: Generating robotic simulation tasks via large language models," *International Conference on Learning Representations*, 2024.

[16] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99-106, 2021.

[17] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3D Gaussian Splatting for Real-Time Radiance Field Rendering," *ACM Transactions on Graphics*, vol. 42,

no. 4, Jul. 2023.

[18] NVIDIA, "Jetson Thor: The compute engine for physical AI—Performance benchmarks," https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-thor/

[19] Team Octo, "Octo: An open-source generalist robot policy," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

AUTHOR

Young-myoung Kang is an Assistant Professor in the Department of Computer Engineering at Sungkyul University, Korea, where he also leads the Sungkyul Artificial Intelligence Convergence Institute. Before joining the faculty in 2021, he spent nine years as a Principal Engineer at Samsung Electronics, bringing extensive industrial expertise to his academic work.

He earned his B.S. degree from Gyeongsang National University in 2000, followed by an M.S. and Ph.D. in Computer Science and Engineering from Seoul National University in 2003 and 2013, respectively. His early career included three years (2003–2006) as a Software Engineer at LG Electronics. His research portfolio spans wireless and wired networks, Wi-Fi, Bluetooth, IoT, and the integration of Artificial Intelligence and Deep Learning.

Beyond his research, he actively contributes to the global academic community as a reviewer for numerous high-impact international journals and conferences, including various IEEE Transactions journals.