The Core Insight

For grounded physical understanding rather than simulation-specific dynamics. A child doesn't learn Newtonian mechanics and then apply it, they learn invariants through interaction. Gravity always pulls down. Heavier things are harder to move. Objects don't pass through each other. These are discovered truths that hold regardless of the specific environment.

The key distinction is between:

- Parametric physics knowledge: knowing that F=ma with specific values
- Structural physics knowledge: knowing that forces cause acceleration, that there's always *some* relationship between effort and motion

A child learning to catch doesn't know the gravitational constant, but they learn that balls fall in predictable arcs and that their arm has certain reach and speed limits. This transfers across balls of different sizes, different lighting conditions, indoors and outdoors because they've learned the structure, not the parameters. Current approaches train on simulation with specific parameters and hope that domain randomization covers reality. I am proposing something slightly different where we learn the underlying structure of physics in a way that's invariant to specific parameters, then adapt the parameters online through interaction.

Essentially a child already knows:

- How far they can reach
- How fast they can move
- How much force they can exert
- What happens when they lose balance
- Their own reaction time and perceptual limits

This self-knowledge is learned through exploration and occasionally failing. a child learns their new body after a growth spurt. As a child they are constantly testing the limits of their movements and their abilities as a person. Current robots typically have this hardcoded or calibrated once. The robot "knows" its kinematics from a URDF file, not from self-exploration. This creates an issue where if something changes (wear, damage, payload), the self-model is wrong.

What Would This Actually Look Like?

1. Learning Physical Invariants

Instead of learning a specific dynamics model, learn constraints and relationships that hold universally:

- Conservation laws (momentum, energy)
- Causality (actions precede effects)
- Contact constraints (objects don't interpenetrate)
- Continuity (state doesn't jump discontinuously)

These could potentially be learned from diverse experience and would transfer across sim and real because they're true in both. The question is how to represent and learn them—perhaps as learned constraints in a structured world model, or as inductive biases in the architecture itself.

One concrete approach: train on data from multiple physics simulators with different engines, parameters, and even different approximations. What the model learns to predict consistently across all of them might be closer to "true" physics than what any single simulator teaches.

2. Self-Model Through Exploration

Have the robot actively explore its own capabilities:

- Move joints to find limits
- Apply forces and observe accelerations to estimate inertias
- Attempt tasks at varying speeds to find dynamic limits
- Deliberately explore failure modes in safe conditions

This gives us an empirical self-model that's grounded in the actual embodiment, not a specification. Importantly, this process should be continuous—the robot should keep updating its self-model as it accumulates experience.

There's precedent for this in some legged locomotion work where robots learn their own dynamics through random motion and system identification. Extending this to manipulation and integrating it with a VLA is less explored.

3. Adaptation Through Interaction

When encountering a new situation, don't just apply the learned model—use interaction to refine it:

- Probe uncertain properties (is this heavy? is this slippery?)
- Start with conservative actions and increase as uncertainty decreases
- Explicitly represent uncertainty about physical parameters and reduce it through observation

This is essentially online system identification, but integrated into the policy rather than as a separate calibration step.

4. Abstracted Action Representations

Rather than learning specific joint trajectories, learn action *concepts* that can be instantiated on different bodies:

- "Reach toward" rather than a specific joint configuration
- "Apply force until contact" rather than a position command
- "Move quickly" relative to the robot's own capabilities

This requires the self-model—you need to know your own capabilities to translate an abstract intention into concrete motor commands.

Technical Challenges

This is hard for several reasons:

Learning true physics from finite experience is epistemologically tricky. How do you know you've learned an invariant rather than a pattern that happens to hold in your training data? Adversarial environments could exploit this.

Representing physical knowledge in a form that's both learnable and generalizable is an open problem. Is it symbolic constraints? Learned embeddings? Architectural inductive biases? Graph networks with physical structure?

The exploration problem is significant. Safe exploration of one's own limits requires careful curriculum design. A robot can't learn about falling without falling, but falling can be destructive.

Grounding abstract knowledge in specific actions requires solving the symbol grounding problem—connecting high-level physical concepts to low-level motor commands through the robot's specific embodiment.

Related Work to Build On

A few threads that connect to this:

Intuitive physics in cognitive science—there's substantial work on how humans represent physical knowledge, what biases we have, and how we learn. Josh Tenenbaum's group at MIT has done a lot here. Their "physics engine in the head" hypothesis is relevant—humans may have something like an approximate simulator, but it's learned and adapted through experience.

Meta-learning for dynamics attempts to learn a prior over dynamics models that can be quickly adapted to new situations with little data. This is related to your adaptation idea.

Morphology-agnostic policies try to learn control that transfers across different robot bodies. This connects to your abstract action representation idea.

Curiosity-driven exploration for learning world models—intrinsic motivation to explore states where the model is uncertain or wrong.

Differentiable physics with learnable parameters could provide a framework where the structure is fixed (Newton's laws) but parameters are inferred from interaction.

Possible Research Framing

"Embodied Physical Intelligence Through Grounded Self-Modeling"

Core questions:

1. What physical knowledge is truly invariant across embodiments and environments, and how can it be learned?
2. How can a robot build and maintain an accurate model of its own capabilities through experience?
3. How should abstract physical reasoning be grounded in specific motor actions given a self-model?
4. What exploration strategies efficiently discover physical invariants and self-capabilities while remaining safe?

The unifying hypothesis would be something like: a robot that learns invariant physical structure and maintains a grounded self-model will exhibit robust transfer across simulation and reality, across different tasks, and across changes to its own body—because it relies on truths that hold universally rather than patterns that hold locally.

Things to note

- This is ambitious and there's a reason the field has mostly gone the other direction (bigger models, more data, domain randomization). The current paradigm is working reasonably well for many tasks, and the approach you're describing requires solving several hard problems simultaneously.
- it's more aligned with how biological intelligence works, and it might be necessary for robust general-purpose robotics. The current approach may hit a ceiling where no amount of data or domain randomization can cover the true diversity of physical situations.