VERDI: VERIFIED CONTRADICTIONS FOR CONTINUAL WORLD-MODEL OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

World models serve different domains and are tuned to different metrics, from visual fidelity to physical consistency and efficiency, so automating their optimization is valuable. Their optimization operators—reusable training, representation, sampling, or inference changes—look portable, yet models agreeing on architecture class, size, and baseline score can respond to the same operator in *opposite* directions, so what repairs one degrades its nearest neighbor. Existing methods supply the search but not the answer: they rank candidates by observational descriptors, assume these predict the same intervention response, and discard failed transfers as noise. We argue that an operator's effect sign is fixed by the model's response along the direction the operator modifies, not by any descriptor, so similarity is not a sufficient statistic and no fixed description can repair itself. The repair signal is a *verified contradiction*—nearby models whose opposing outcomes for the same operator are established by probing and by paired experiments under a verifier the search loop cannot edit. VERDI turns this into recursive self-optimization through shared-probe Optimization Fingerprints, IRG-guided retrieval, frozen-verifier settlement, and probe admission gated by held-out replay. Across nine backbones, 27 leave-one-backbone-out campaigns, and 216 transfer decisions, VERDI predicts effect signs with 82.9% accuracy, reaches the first verified positive in 3.83 trials and 105 GPU-h (66.5% and approximately 67% below cold start), and lowers negative-transfer risk from 33.8% to 5.9% at 47.2% coverage. Ablations confirm that the geometry, not permissive acceptance, does the work, and a small-scale Franka Panda study gains 17.5 percentage points on two tasks. **Project page:** https://verdiwm-anno.github.io/

1 INTRODUCTION

World models predict how environments evolve and are used for planning, simulation, and control (Ha & Schmidhuber, 2018; Hafner et al., 2020; 2025). They serve different domains—robotics, driving, interactive simulation—so different backbones are tuned to different metrics, from visual fidelity to physical consistency or efficiency (NVIDIA et al., 2025; Guo et al., 2025; Google DeepMind, 2025; Han & Yilmaz, 2026), and each still needs its own improvement campaign. Automated, continual world-model optimization is therefore valuable: it turns a per-model effort into experience that compounds. We call a reusable optimization operation an *operator*: an executable change to training, representation, sampling, or inference. Operators look portable, yet models that agree on architecture class, parameter count, and baseline scores can respond to the same operator in *opposite* directions, so what repairs one degrades its nearest neighbor. We therefore ask: *how can optimization experience accumulate when the models that look most alike can call for opposite interventions?*

Existing methods supply the search but not the answer. Backbone-specific methods improve one model at a time (Guo et al., 2025; NVIDIA et al., 2025; Zhu et al., 2025); research agents automate proposing, implementing, and evaluating such improvements (Lu et al., 2024; Huang et al., 2024; Novikov et al., 2025; Zhang et al., 2025); warm-started transfer reuses priors across related tasks (Feurer et al., 2015; Bai et al., 2023; Achille et al., 2019; Chen et al., 2022). All of them rank candidates by observational descriptors—architecture, metadata, embeddings, scores—and assume these predict the same intervention response, so a failed transfer is discarded as noise rather than as evidence about the description that failed.

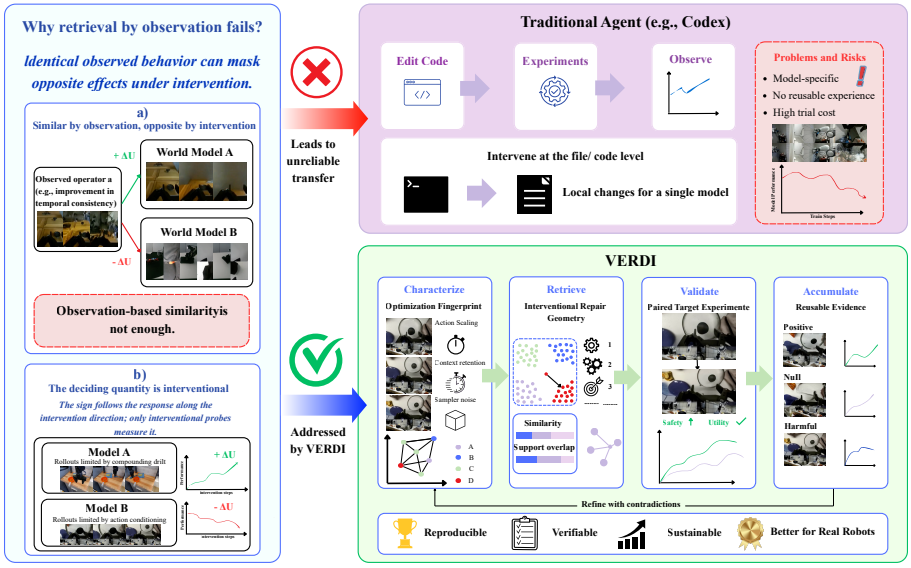

Figure 1: **From observational similarity to reusable repair evidence.** Left: (a) observationally similar models can respond oppositely to the same operator; (b) its effect depends on the interventional response along the direction it modifies. Upper right: a conventional agent edits code and evaluates changes within one model. Lower right: VERDI characterizes models with shared probes, retrieves operators by response geometry, validates paired target experiments under a frozen verifier, and accumulates positive, null, and harmful outcomes for reuse. Verified contradictions guide probe refinement, with new probes admitted only when held-out replay improves selection and calibration.

We argue that the bottleneck is that these descriptors are observational: whether an operator helps is decided by the model's *response along the direction that the operator modifies*, relative to the error that dominates its rollouts. Two models can match on every descriptor yet differ in what limits them, so the same operator repairs one and degrades the other. Similarity is thus not a sufficient statistic for intervention effects, and no fixed description can repair itself, because it carries no signal about the distinction it is missing. That signal is observable in a *verified contradiction*: two nearby models whose opposing outcomes for the same operator survive cheap inference-time probing and paired experiments under a verifier that the search loop cannot edit.

Continual optimization should therefore co-evolve two objects in one loop: the operator library, and the behavioral description that decides where each operator applies. Models should be characterized by interventional responses, not metadata; candidates should be settled by experiment, not prediction, so that null and harmful outcomes become evidence; and a verified contradiction should trigger the probe that best explains it, admitted only if held-out replay improves selection. Each batch of campaigns then consumes settled outcomes and returns a better geometry, instead of starting cold.

We instantiate this view in VERDI, a four-stage loop (Figure 1). *Characterize* forms an *Optimization Fingerprint* from shared inference-time probes. *Retrieve* ranks executable operators with the Interventional Repair Geometry (IRG) over an Operator registry and Operator outcome archive. *Validate* runs paired target experiments that a frozen verifier—not the proposal or the search loop—settles. *Accumulate* records each outcome with its support, uncertainty, and lineage, and admits a new probe only when held-out replay improves selection and calibration.

Experiments on ACWM-Phys (Xue et al., 2026) and nine pretrained backbones test both claims. Selected operators raise Ctrl-World motion smoothness from 68.75 to 81.53 and COSMOS3 simulated control by 12–24 points, and a small-scale Franka Panda study gains 17.5 percentage points. Across 27 leave-one-backbone-out campaigns and 216 transfer decisions, VERDI predicts effect signs with 82.9% accuracy, reaches the first verified positive in 3.83 trials and 105 GPU-h, 66.5% and approximately 67% below cold start, and holds negative-transfer risk at 5.9% over 47.2% coverage against 33.8% under unconditional reuse. Ablations show that the geometry does the work: raw responses or learned embeddings reach 0.708 and 0.741 sign accuracy, and removing IRG gives 4.90

trials and 0.18 risk. Under a matched 200 GPU-h allowance, VERDI completes 24/27 campaigns in 3.83 trials, versus 20/27 for Codex (GPT-6). Our contributions are threefold:

- **A paradigm for continual world-model optimization.** Verified opposing outcomes between models that a similarity description places close together reveal a missing behavioral distinction, so operators and the geometry that predicts where they apply should co-evolve.
- **A response-aware framework.** VERDI realizes this as a four-stage loop of Optimization Fingerprints, IRG-guided retrieval, frozen-verifier validation, and an outcome archive whose probe refinement must improve held-out selection.
- **A benchmark and verification protocol.** Leave-one-backbone-out evaluation over nine backbones, 27 campaigns, and 216 transfer decisions, with archive replay and a planted-fault audit, verifies both claims.

2 Related Work

World-model evaluation. Evaluation now extends beyond visual fidelity to dynamics, action fidelity, and embodied utility. EWMBench assesses scene, motion, and semantics (Yue et al., 2025); WorldArena and WorldArena2.0 combine video metrics with embodied tasks (Shang et al., 2026a;b). GigaWorld analyzes relations among evaluation metrics (GigaAI, 2025). These benchmarks define outcomes; VERDI asks which operator changes them for a given model and whether its effect transfers across backbones.

World-model improvement. World models have advanced through latent imagination for control (Hafner et al., 2025), controllable manipulation dynamics (Guo et al., 2025; Zhu et al., 2025), and large video or world-action backbones (NVIDIA et al., 2025; Han & Yilmaz, 2026). These works improve individual models through architecture, training, or conditioning choices. VERDI studies how such interventions become reusable evidence across backbones by recording their measured effects, support conditions, and validity boundaries.

Research and self-improving agents. Research agents automate code changes and experimental search (Lu et al., 2024; Novikov et al., 2025), including robot-policy improvement loops (Xiao et al., 2026; Li et al., 2026). Warm-started optimization, task embeddings, and optimization-history models guide new decisions from prior tasks (Feurer et al., 2015; Achille et al., 2019; Chen et al., 2022). These lines of work automate search or transfer optimization priors, but do not directly model how an operator affects heterogeneous world-model backbones. VERDI uses validated source–target effects to select operators and refines its response geometry when nearby models with comparable support yield opposing effects.

3 Method

3.1 Problem Setting

We study repeated campaigns for improving pretrained world models. In campaign x, a model M_x produces future observations, and, when supported by its interface, actions or task events from a context and optional control inputs. The campaign $\mathcal{C}_x = (M_x, \mathcal{D}_x, \mathcal{G}_x, V_x, B_x)$ fixes the permitted data $\mathcal{D}_x$, optimization goal $\mathcal{G}_x$, verifier V_x, and nominal experimental budget B_x. The goal specifies a utility $U_{\mathcal{G}_x}$, protected outcomes that must remain within declared tolerances, and a minimum meaningful gain $\tau_{\mathcal{G}_x}$. This formulation covers both action-conditioned world models and world-action models; backbone-specific details are handled by adapters.

The object we seek to reuse is an operator a: an executable change to training, representation, sampling, or inference with a specified model hook, dose, schedule, and preconditions. Its value is an empirical target-side outcome, not the fact that it worked once. Let $M_x^{(a)}$ be the model after applying a, and $M_x^{(0)}$ its baseline control. The campaign utility and its paired gain are

$$U_x(a) = U_{\mathcal{G}_x}\left(M_x^{(a)}; \mathcal{D}_x, V_x\right), \qquad \Delta U_x(a) = U_x(a) - U_x(0). \tag{1}$$

Both arms use matched initialization, data, evaluation contexts, rollout seeds, and the same frozen verifier. Protected outcomes are checked separately against their declared tolerances. An operator

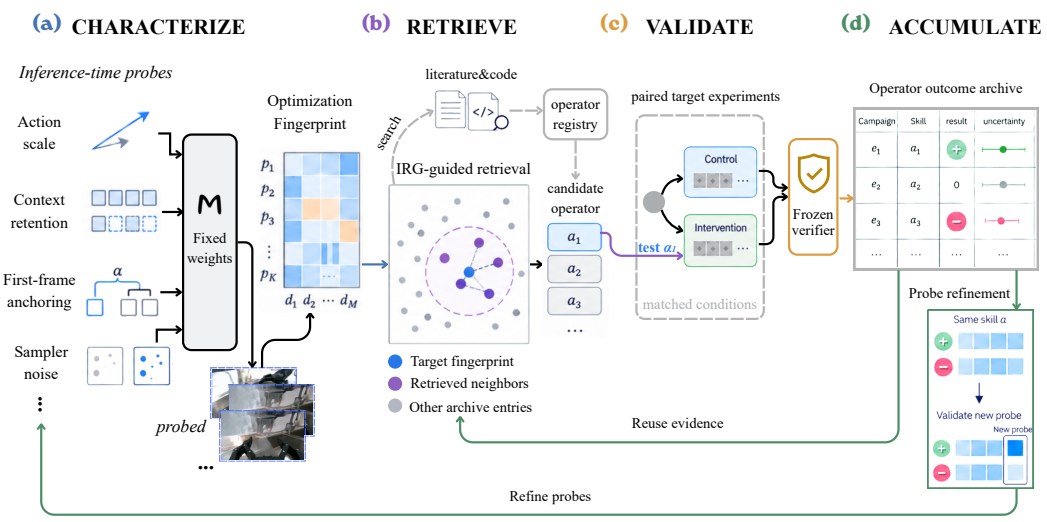

Figure 2: **VERDI's continual optimization loop.** *Characterize* measures shared inference-time responses to build an Optimization Fingerprint. *Retrieve* uses IRG-guided retrieval to compare the fingerprint with archived models and may search literature or code for a new operator. *Validate* runs paired target experiments under a frozen verifier, while *Accumulate* writes settled positive, null, harmful, and interaction outcomes to an Operator outcome archive. The feedback path admits a new probe only after held-out checks.

outcome is positive only when the lower confidence bound on $\Delta U_x(a)$ exceeds $\tau_{\mathcal{G}_x}$ and every protected check passes. A utility gain cannot compensate for a protected regression. Throughout, a *trial* is one candidate operator attempt; compilation, screening, confirmation, and additional rollout seeds are events within that attempt.

3.2 Method Overview

VERDI follows four stages (Figure 2). *Characterize* measures responses to a shared set of inference-time probes and forms an Optimization Fingerprint. *Retrieve* uses IRG-guided retrieval to rank operators using the Operator registry and Operator outcome archive and to guide external search when no suitable record exists. *Validate* tests candidates through paired screening and held-out confirmation on the target model. *Accumulate* records the settled operator outcome and uses reliable opposing outcomes among nearby models to decide whether an additional probe is useful.

For illustration, suppose models A and B respond similarly to existing probes, but a temporal operator helps A and harms B under paired validation. This disagreement suggests a missing distinction. VERDI proposes a history-sensitive probe, remeasures both models, and retains the probe only if held-out replay improves selection.

A fixed LLM executor searches external sources and implements operators through backbone adapters with implementation receipts. It cannot change the verifier or decide which outcomes enter the archive. In transfer evaluation, the Operator outcome archive and Operator registry are frozen before target evaluation; newly discovered operators enter later versions.

3.3 Characterize: Building the Optimization Fingerprint

We probe a model at inference time, before choosing a training or architecture change. Each shared probe $q \in \mathcal{Q}$ makes a small, reversible change of dose α to a behavioral input or inference setting: for example, action magnitude, retained temporal context, first-frame anchoring, or sampler noise. Adapters translate the same probe contract to each backbone without updating weights or changing the optimization goal. For diagnostic and validity measurements z_k on matched contexts ξ and seeds

ω, we estimate

$$J_x[k, q] = \mathbb{E}_{\xi,\omega}\left[\frac{z_k(M_x^{q,\alpha}; \xi, \omega) - z_k(M_x; \xi, \omega)}{\alpha}\right]. \tag{2}$$

Each column of J_x records one probe's response. Pairing reduces sampling noise, and bootstrap estimates give uncertainty Σ_x. Action scaling probes action sensitivity; context retention probes history dependence. These diagnostics guide candidate operator selection; actual outcomes still require paired target-side experiments.

We call $\phi_x = (J_x, G_x, \Sigma_x, c_x)$ the *Optimization Fingerprint*. Here $G_x = J_x^\top W_x J_x + \lambda I$ weights standardized diagnostic outcomes according to the goal and regularizes weak probe directions; c_x records which probe and operator hooks the adapter supports. The fingerprint provides a common behavioral description across backbones while keeping measurement uncertainty and implementation compatibility explicit. Probes measure the model's responses; operators are the optimization operations subsequently selected and tested.

3.4 Retrieve: IRG-Guided Operator Selection

The Interventional Repair Geometry (IRG) organizes models and their historical operator outcomes using responses to shared inference-time probes. It compares compatible Optimization Fingerprints, with goal-specific weighting and a penalty for mismatched evaluation support. Two models are close when the same controlled perturbations change relevant outcomes in similar ways, even if their architectures differ. Fingerprint uncertainty enters confidence estimates and abstention.

The Operator registry stores executable operator definitions and their implementation requirements; the Operator outcome archive records what happened when those operators were applied. IRG-guided retrieval ranks candidate operators using outcomes from compatible neighbors. The target fingerprint, failure coordinates, supported hooks, and goal also guide external search for additional candidates. A discovered candidate is compiled and evaluated by the same target-side screen and verifier; retrieval alone never assigns a target outcome. The archive and registries are frozen before held-out outcomes are observed, and newly discovered operators belong to a later version.

The Operator outcome archive links each operator and source fingerprint to its settled gain, protected outcomes, support, uncertainty, cost, verifier result, and artifact lineage. It retains positive, null, harmful, and jointly tested interaction outcomes. For target t and operator a, only records compatible with its goal, support, and adapter capabilities contribute to the gain estimate:

$$\widehat{\Delta U}_t(a) = \frac{\sum_{s\in\mathcal{N}_t(a)} w_{ts}(a)\, r_s^U(a)}{\sum_{s\in\mathcal{N}_t(a)} w_{ts}(a)}, \qquad w_{ts}(a) \propto \frac{\rho_s(a)\kappa_{ts}}{\epsilon + \mathrm{dist}_{\mathrm{IRG}}(\phi_t, \phi_s)}. \tag{3}$$

Here $r_s^U(a) = \Delta U_s(a)$ is the source gain, $\rho_s(a)$ measures evidence quality, and κ_{ts} measures support overlap. The selector allocates trials by an upper-confidence gain-per-cost score; separate compatibility checks determine transfer priority. These estimates rank hypotheses, while target trials establish effects.

3.5 Validate: Paired Target Experiments

Before a retrieved operator receives transfer priority, VERDI checks that it compiles for the target, source and target supports overlap, enough credible source evidence exists, probe responses align, nearby effect signs agree, and a calibrated lower bound on predicted gain clears the goal threshold. This check prevents weak or conflicting evidence from being treated as a transferable result. A candidate that fails it can still be explored from a cold start if it compiles locally. Protected outcomes are measured on the operator-applied target model.

Each selected operator first receives a paired, low-cost screen against the matched control. Promising candidates then undergo held-out confirmation with longer rollouts, more optimization steps, and additional seeds. The frozen verifier settles the result using the campaign's predeclared utility threshold and protected tolerances. A positive record requires a lower confidence bound on target gain above $\tau_{\mathcal{G}_t}$ and all protected checks to pass. Valid confirmed null and harmful operator outcomes are also recorded with their uncertainty; compilation failures, invalid runs, and unconfirmed screens do not become settled outcome evidence. These paired target experiments establish whether the operator improves the target model.

3.6 ACCUMULATE: RECORDING OUTCOMES AND REFINING PROBES

After validation, VERDI writes the settled operator outcome together with its support, uncertainty, verifier decision, cost, operator version, and implementation receipt to the Operator outcome archive. Positive records provide future candidates, while null and harmful records mark limits of applicability. Jointly measured combinations receive interaction records rather than being attributed to either operator alone. Source–target tests update the evidence available for future retrieval without rewriting the original source records.

The archive also reveals when its behavioral description is insufficient. In the example above, models A and B are close in IRG but show reliable opposing outcomes for the same temporal operator. The current fingerprint may therefore omit a distinction relevant to operator selection; an isolated failure without a comparable nearby source does not establish this. VERDI uses the disagreement context and available adapter hooks to propose a history-sensitive probe, and retains it only if held-out replay improves operator selection and calibration checks pass. Probe refinement updates IRG for later campaigns, while every claimed operator outcome remains grounded in a verifier-controlled target experiment.

4 EXPERIMENTS

We evaluate model-specific improvements, compare transfer and operator-search efficiency, and then isolate the contributions of selection, probe evolution, and verification through ablations.

4.1 EXPERIMENTAL PROTOCOL

Campaigns freeze checkpoints, splits, contexts, seeds, metrics, validity gates, registry snapshots, and nominal budgets. In LOBO, baseline and operator-applied arms share initialization, data, schedules, and evaluators. The LOBO study covers nine backbones with three seeds each (27 campaigns, 216 transfer decisions; App. 6), excluding each target's records from its source archive. Transfer and controlled selector/scaffold comparisons use this study, as do archive-prefix replay and external-system comparisons. Confidence intervals resample the nine backbone clusters with their seeds retained; separate case studies state their own evaluation scope. Trials and compute are macro-averaged over backbones after averaging tasks and repeated searches within successful campaign–seed units; unsuccessful campaigns remain in the completion denominator (App. B). We distinguish cost to the first verified positive, $C_{\text{first}+}$, from full search cost, C_{search} (App. B.10). A 512-step screen orders candidates; the official 50-step evaluation and held-out confirmation settle effects. Protected metrics are checked separately from utility (App. C.1).

4.2 TARGETED OUTCOMES OF MODEL-SPECIFIC OPTIMIZATION

We evaluate visual quality, physical-interaction prediction, and real-robot control to distinguish improvements in generated observations from downstream action outcomes.

Visual quality. On Ctrl-World, VERDI selects a latent-motion prior that reduces rollout drift (Figure 3a). Four primary coordinates improve, including motion smoothness from 68.75 to 81.53; subject consistency falls by 0.96 within tolerance (Table 1). A protected dynamic-degree regression on open-drawer limits these gains to descriptive coordinate improvements, not a verified gain over the full scope (App. C.3). On COSMOS3, a combined reinforcement-learning (RL) and self-forcing operator reduces subject and background drift in long rollouts, while the complementary WAM and real-robot results below test whether the operator reaches downstream control (App. C.4).

Physical consistency. Table 2 reports operator-applied minus baseline changes on the eight ACWM-Phys environments under the frozen evaluator. PSNR and SSIM increase, while masked MSE decreases, in every environment. The largest PSNR gain occurs on `robot_arm` (+1.08 dB); `cloth_move` shows the largest SSIM gain (+0.0099) and masked-MSE reduction (-9.040×10^{-3}).

Real-world robot control. We evaluate the operator-applied COSMOS3 world-action model (WAM), using the combined RL + self-forcing operator, separately on a Franka Panda with eight-step

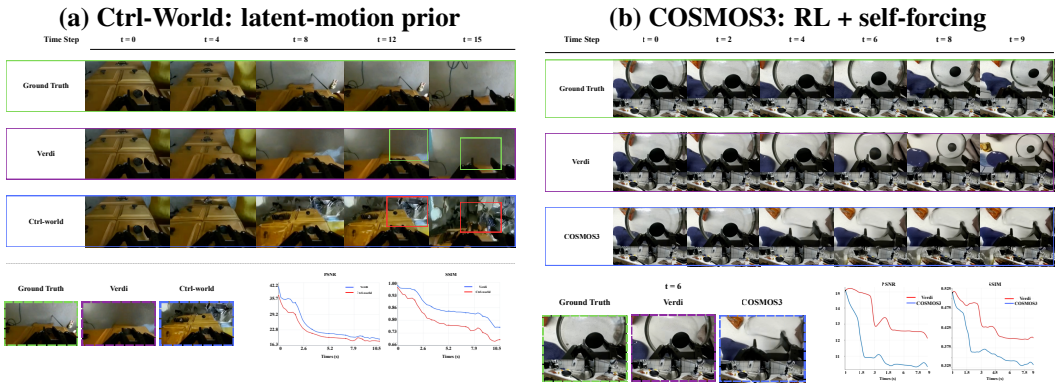

Figure 3: **Visual quality across backbones.** Ground truth, baseline, and operator-applied rollouts, with paired PSNR/SSIM curves. (a) Ctrl-World's latent-motion operator. (b) COSMOS3 with RL + self-forcing on "loosen the pot handle" (DROID (Khazatsky et al., 2024)).

Table 1: **Model-specific optimization results.** Backbones match Figure 3: (a) Ctrl-World visual metrics; (b) COSMOS3 simulation success and action diagnostics under the combined RL + self-forcing operator. Success decay is the 8-step to 14-step task gap in percentage points.

(a) Ctrl-World: visual quality

Metric ↑	Base	VERDI
Semantic alignment	90.70	**91.30**
Depth accuracy	93.28	**96.16**
Flow score	26.54	**30.35**
Motion smoothness	68.75	**81.53**
Subject consistency	**84.31**	83.35

(b) COSMOS3: simulated control

Diagnostic	COSMOS3	VERDI
Pick-and-place success ↑	62%	**74%**
Drawer opening success ↑	41%	**59%**
Peg insertion success ↑	28%	**52%**
Expert-action L2 error ↓	0.084	**0.061**
Success decay ↓	34 pts	**22 pts**

Table 2: **ACWM-Phys physical-consistency summary.** Entries are operator-applied minus baseline changes under the frozen evaluator. Higher PSNR and SSIM and lower masked MSE (mMSE) are better. ΔmMSE is reported in units of 10^{-3}.

Environment	ΔPSNR (dB) ↑	ΔSSIM ↑	ΔmMSE (10^{-3}) ↓
push_cube	+0.15	+0.0002	-0.284
stack_cube	+0.03	+0.0020	-0.531
push_rope	+0.02	+0.0008	-0.151
cloth_move	+0.94	+0.0099	-9.040
push_sand	+0.64	+0.0010	-0.783
pour_water	+0.85	+0.0019	-1.718
robot_arm	+1.08	+0.0098	-2.230
reacher	+0.06	+0.0002	-0.119

action chunks and re-observation (Figure 4). Success rises from 11/20 to 14/20 for pick-and-place and from 7/20 to 11/20 for drawer opening: +17.5 percentage points on average. Marginal 95% Wilson intervals are [34,74]% to [48,85]% and [18,57]% to [34,74]%, respectively. These small-sample results suggest downstream benefit.

Additional downstream evaluations. Appendix E.5 groups DreamerV3, V-JEPA 2, DINO-WM, and TD-MPC2 visualizations with task metrics and a case index. On RoboTwin, V-JEPA 2 success increases from 128/400 (32.0%) to 207/400 (51.75%). These standalone evaluations are excluded from the 216 LOBO decisions.

4.3 Cross-Backbone Transfer and Search Efficiency

Useful individual operators do not establish that experience transfers. Chronological-prefix replay tests this on the same held-out campaigns: as the archive grows, sign accuracy rises from 0.61 to 0.829, trials fall from 7.8 to 3.83, and negative-transfer risk falls from 0.18 to 0.059 while coverage rises from 0.22 to 0.472 (Figure 5).

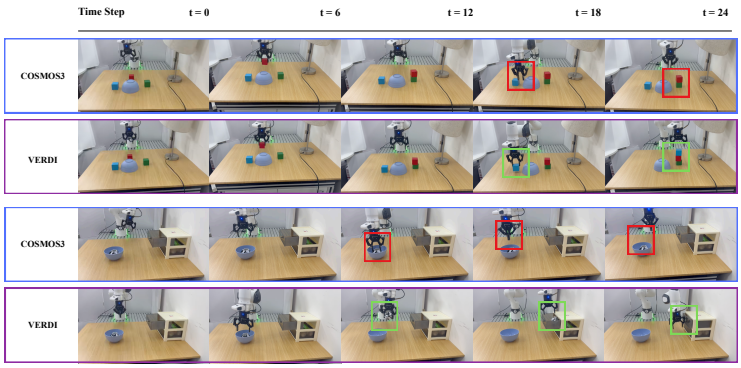

Figure 4: **Real-robot control.** Baseline COSMOS3 (blue) and COSMOS3 with RL + self-forcing (purple) on pick-and-place and drawer opening, with matched initial states and instructions. Boxes mark failed (red) and successful (green) actions. Additional qualitative rollouts on the AgileX Cobot Magic dual-arm robot are provided in Appendix C, Figure 12.

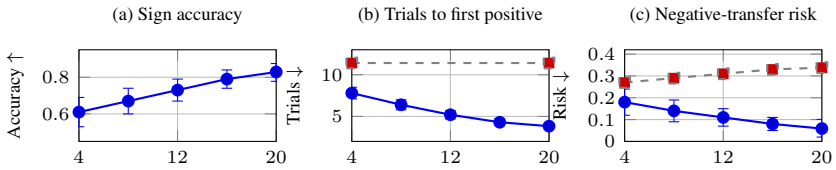

Figure 5: **Chronological archive replay.** Frozen prefixes evaluated on 27 campaign–seed units from nine backbones and 216 decisions per prefix; bars are descriptive 95% intervals from nine backbone clusters. Dashed lines: cold start (b), unconditional reuse (c).

With the full archive, VERDI identifies the first verifier-confirmed positive operator outcome after 3.83 trials and 105 GPU-h on average, compared with 11.41 trials and 318 GPU-h for cold-start search. Across 216 held-out transfer decisions, its gate accepts 102 (47.2% coverage), including 6 harmful outcomes (5.9% risk), versus 73/216 (33.8%) harmful outcomes under unconditional reuse. Coverage measures the gate's acceptance rate, while risk is conditional on accepted transfers. Per-backbone results appear in App. B.4. Cross-family transfer accepts 62/156 candidates, including 40 positive, 19 null, and 3 harmful outcomes (App. E.4).

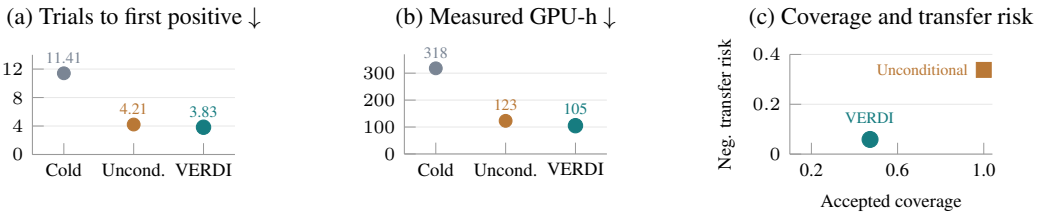

Figure 6: **Transfer versus cold start and unconditional reuse.** (a,b) Nine-backbone macro-means of trials and measured GPU-h over successful campaigns; 27 campaigns define the completion denominator. (c) Coverage and conditional risk across 216 decisions. Point estimates; intervals and aggregation are in App. E.1.

4.4 Comparison with Coding Agents and Human Experts

Table 4 compares coding agents and human experts across nine backbones and 27 campaigns, with a shared registry, verifier, and 200 GPU-h nominal allowance. VERDI completes 24 campaigns, versus 20 for Codex and 19 each for Claude Code and experts, and requires an average of 3.83 candidate trials to reach the first verifier-confirmed repair, compared with 4.6 candidate trials for Codex, 4.9 for Claude Code, and 7.4 for human experts. VERDI, Codex, and Claude Code attain comparable final utility, while human experts attain higher final utility. Native executors differ, so this comparison measures end-to-end performance rather than isolating the control scaffold. Matched-GPT-5.5 controls are in App. B.3.

Table 3: **Selection-signal ablation.** 216 held-out decisions; 27 LOBO campaign–seed units.

Selector	Done /27	Sign acc.↑	Trials ↓	Viol. ↓
Random	15	.500	8.90	.33
Metadata	18	.648	5.20	.13
Raw response	20	.708	4.30	.09
Learned emb.	22	.741	4.60	.10
Full fingerprint	**24**	**.829**	**3.83**	**.03**

Table 4: **External systems.** Completion and trials over 27 campaigns.

System	Done ↑	Trials ↓
VERDI (GPT-5.5)	**24/27**	**3.83**
Codex (GPT-6)	20/27	4.6
Claude Code (Fable-5)	19/27	4.9
Human experts	19/27	7.4

4.5 Ablation Studies

We ablate four decisions in VERDI: representation, control scaffold, probe admission, and protected-outcome checks. These tests separate candidate ranking from permissive acceptance.

Selection representation. Table 3 varies only the selector with the executor, archive, Operator registry, and verifier fixed. Full fingerprint reaches 0.829 effect-sign accuracy on 216 decisions, versus 0.708 for raw responses and 0.741 for a learned embedding. At matched coverage (102/216), risks are 5.9%, 11.8%, and 14.7% for IRG, response-informed RGPE, and the learned embedding (App. E.3).

Control scaffold under a fixed executor. The external comparison also changes the executor. With GPT-5.5 fixed, removing eligibility checks raises trials from 3.83 to 4.40 and risk from 0.059 to 0.21, while removing IRG gives 4.90 trials and 0.18 risk (Table 5). Bare and generic loops require 189–210 GPU-h versus 105 for the full system.

Table 5: **Control-scaffold ablation.** Nine backbones, 27 campaigns; shared registry, verifier, contexts, and nominal 200-GPU-h allowance. Done counts successful campaigns; trials and $C_{\text{first}+}$ use successful campaigns.

Configuration	Done/27	Trials	GPU-h	Risk	Prot. viol.
w/o eligibility checks	21	4.40	132	.21	.14
w/o IRG	20	4.90	147	.18	.11
Bare LLM agent	18	6.00	189	.31	.26
Generic research-agent loop	16	6.60	210	.36	.29
Full VERDI	**24**	**3.83**	**105**	**.059**	**.03**

Probe refinement. Starting from four probes, gated growth admits three more, reducing nested regret from 0.29 to 0.21 and calibration error from 0.14 to 0.08; unrestricted growth uses fourteen probes but worsens both to 0.31 and 0.19 (Figure 7). At a matched seven-probe budget, contradiction-driven selection achieves regret 0.21 versus 0.30 for random selection and 0.26 for replay-gain (App. E.2).

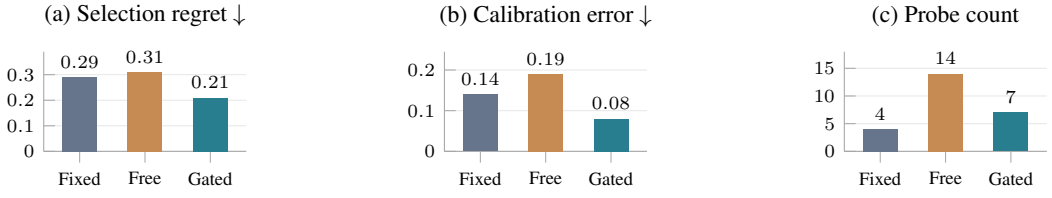

Figure 7: **Probe-growth ablation.** Fixed: four probes; Free: unrestricted growth; Gated: admission from opposing outcomes. Regret is relative to the best admissible operator, over nine LOBO targets (27 campaign–seed units). The matched-budget extension uses disjoint discovery/promotion/calibration/target sets of 4/2/2/1 backbones per fold (App. E.2).

Protected-outcome verification. Removing negative-correlation gates raises nominal gain from +12.78 to +14.92 but also raises protected violations from 0.03 to 0.41 (App. B.7); a larger scalar gain cannot substitute for a verified improvement.

5 Conclusion

We studied how optimization experience can accumulate across world models that look alike yet can require opposite interventions. VERDI treats verified disagreement as evidence: it probes each backbone, retrieves operators through the Interventional Repair Geometry, and settles paired target experiments under a frozen verifier; verified contradictions then repair the description used by later campaigns. Across nine backbones, this loop predicts effect signs with 82.9% accuracy. It reduces trials to a first verified improvement by approximately 66.5% relative to cold start and achieves 5.9% negative-transfer risk at 47.2% coverage. Future work should re-measure fingerprints and local effects after material changes to architecture, data, or hooks, and give each new objective its own verifier version.

AI Use Statement

We used generative AI tools in two roles. First, GPT-5.5 is the explicitly evaluated executor in VERDI: under a frozen prompt and tool schema, it proposes candidate operators, implements them through backbone adapters, and executes the declared search and evaluation workflow. Second, language models assisted with manuscript editing, LaTeX diagnostics, figure and table layout. The authors retain responsibility for validating AI-assisted text, code, figures, and scientific claims and for the final content of this work.

Ethics Statement

VERDI improves auditability but does not guarantee truth; results require independent replication, human review, and disclosure of generative-agent use. Training jobs, repository edits, and robot access require isolated credentials, least privilege, sandboxing, human approval, and an independent safety layer. Verify licenses and privacy for robot data/checkpoints, record provenance without secrets, assess dual-use risks, and use staged evaluation with explicit compute accounting.

Reproducibility Statement

Baseline and operator-applied arms share initialization, steps, data, evaluator, paired contexts/seeds, target hiding, archive visibility, Operator registry, and nominal allowance; measured costs are separate. The GPT-5.5 executor runs with a frozen prompt and tool schema. Protocols and controlled comparisons are detailed in Appendix B; base models and datasets are publicly available.

References

Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless C. Fowlkes, Stefano Soatto, and Pietro Perona. Task2vec: Task embedding for meta-learning. *arXiv preprint arXiv:1902.03545*, 2019.

Tianyi Bai, Yang Li, Yu Shen, Xinyi Zhang, Wentao Zhang, and Bin Cui. Transfer learning for bayesian optimization: A survey. *arXiv preprint arXiv:2302.05927*, 2023.

Yutian Chen, Xingyou Song, Chansoo Lee, Zi Wang, Qiuyi Zhang, David Dohan, Kazuya Kawakami, Greg Kochanski, Arnaud Doucet, Marc'aurelio Ranzato, Sagi Perel, and Nando de Freitas. Towards learning universal hyperparameter optimizers with transformers. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2205.13320.

Matthias Feurer, Jost Tobias Springenberg, and Frank Hutter. Initializing bayesian hyperparameter optimization via meta-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, pp. 1128–1135, 2015.

GigaAI. Gigaworld-0: A large-scale world model and analysis of metric correlation structure. *arXiv preprint*, 2025.

Google DeepMind. Genie 3: A new frontier for world models. Technical report, 2025.

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-world: A controllable generative world model for robot manipulation. *arXiv preprint arXiv:2510.10125*, 2025.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025. doi: 10.1038/s41586-025-08744-2.

Yuci Han and Alper Yilmaz. Enhancing policy learning with world-action model. *arXiv preprint arXiv:2603.28955*, 2026.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. Mlagentbench: Evaluating language agents on machine learning experimentation. *arXiv preprint arXiv:2310.03302*, 2024.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *Advances in Neural Information Processing Systems*, 2025.

Bo Jiang, Depu Meng, Yihan Hu, Yichen Xie, Tianshuo Xu, and Wei Zhan. Lamo: Self-supervised latent motion priors for physical realism in video generation. *arXiv preprint arXiv:2605.23878*, 2026.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Aditya Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. In *Robotics: Science and Systems*, 2024.

Ruiying Li, Yunlang Zhou, YuYao Zhu, Kylin Chen, Jingyuan Wang, Sukai Wang, Kongtao Hu, Minhui Yu, Bowen Jiang, Zhan Su, Jiayao Ma, Xin He, Yongjian Shen, Yang Yang, Guanghui Ren, Maoqing Yao, Wenhao Wang, and Yao Mu. Roboclaw: An agentic framework for scalable long-horizon robotic tasks. *arXiv preprint arXiv:2603.11558*, 2026.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The ai scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.

Alexander Novikov et al. Alphaevolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13139*, 2025.

NVIDIA, Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Yuqing Shang, Zonghao Li, Yuchen Ma, Weijie Su, Xiaobo Jin, Zan Wang, Li Jin, Xin Zhang, Yansong Tang, Hao Su, et al. Worldarena: A unified benchmark for evaluating perception and functional utility of embodied world models. *arXiv preprint arXiv:2602.08971*, 2026a.

Yuqing Shang et al. Worldarena 2.0: Extending embodied world model benchmarking on modality, functionality and platform. *arXiv preprint arXiv:2605.17912*, 2026b.

Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale. *arXiv preprint arXiv:2308.12952*, 2023.

Shihan Wu, Xuecheng Liu, Shaoxuan Xie, Pengwei Wang, Xinghang Li, et al. Robocoin: An open-sourced bimanual robotic data collection for integrated manipulation. *arXiv preprint arXiv:2511.17441*, 2025.

Wenli Xiao, Jia Xie, Tonghe Zhang, Haotian Lin, Letian Fu, Haoru Xue, Jalen Lu, Yi Yang, Cunxi Dai, Zi Wang, Jimmy Wu, Guanzhi Wang, S. Shankar Sastry, Ken Goldberg, Linxi Fan, Yuke Zhu, and Guanya Shi. Enpire: Agentic robot policy self-improvement in the real world. https://research.nvidia.com/labs/gear/enpire/, 2026.

Haotian Xue, Yipu Chen, Liqian Ma, Zelin Zhao, Lama Moukheiber, Yuchen Zhu, and Yongxin Chen. Acwm-phys: Investigating generalized physical interaction in action-conditioned video world models. *arXiv preprint arXiv:2605.08567*, 2026. URL https://arxiv.org/abs/2605.08567v2.

Shijie Yue et al. Ewmbench: Evaluating scene, motion, and semantic quality in embodied world models. *arXiv preprint arXiv:2505.09694*, 2025.

Jenny Zhang, Shengran Hu, Cong Lu, Jeff Clune, and Robert Tjarko Lange. Darwin-gödel machine: Open-ended self-improving agents. *arXiv preprint arXiv:2505.22954*, 2025.

Fangyuan Zhu et al. Irasim: Learning interactive real-robot action simulators. *arXiv preprint*, 2025.

A Formal Method Details

We study automated improvement of a pretrained world model under a user-specified goal and a finite verification allowance. Small, controlled inference-time probes provide paired response Jacobians, which describe how distinguishable behavioral changes appear along different probe directions. VERDI organizes these responses into an Interventional Repair Geometry (IRG) and uses compatible local charts to rank operator outcome evidence across backbones. When nearby models show opposite effects for the same operator, that disagreement supplies the next diagnostic direction and updates the geometry for later campaigns.

A.1 Goal-Conditioned World-Model Improvement

A campaign is a tuple $\mathcal{C} = (b, \theta_0, \mathcal{D}, G, V_G, \mathcal{B})$, where b is a backbone with pretrained parameters θ_0, $\mathcal{D}$ is the campaign data, G is a goal specification, V_G is an independent verifier, and $\mathcal{B}$ is a nominal resource allowance. The goal specifies a primary utility U_G, evaluation contexts $x \sim \mathcal{D}^{\mathrm{ver}}$, horizons $h \in \mathcal{H}_G$, and a set of validity constraints v_j. For a candidate model θ, we define

$$J_G(\theta) = \sum_{h \in \mathcal{H}_G} w_h \, \mathbb{E}_{x \sim \mathcal{D}^{\mathrm{ver}}}[U_G(\theta; x, h)], \quad \text{s.t.} \quad v_j(\theta) \le 0 \ \ \forall j. \tag{4}$$

The constraints make the utility contract explicit: an operator must improve the target while preserving action use, motion, and other declared behaviors. Our outcome schema follows the multi-dimensional world-model evaluation suite used in the experiments, with tolerances calibrated from the observed relationships among metrics. All comparisons use paired evaluation contexts and seeds. For an operator set S, we retain $\boldsymbol{y}(S) = [\Delta J_G(S), \{\Delta U_k(S,h)\}_{k,h}, \{v_j(S)\}_j, C(S)]$, where $C(S)$ is measured cost. Keeping this vector makes it possible to follow an operator across short-horizon fidelity, long-horizon dynamics, and behavioral validity instead of reducing the campaign to one score.

A.2 Interventional Repair Geometry

A static probe bank first extracts $\boldsymbol{z}_x = P_v(\theta, \mathcal{D}^{\mathrm{dev}}, G)$ for campaign instance x. These inexpensive observations condition effect prediction; the geometric coordinates come from paired intervention responses so that their meaning is shared across backbones.

The backbone adapter declares a capability vector $\boldsymbol{c}_x$ and a set of typed intervention paths $\mathcal{I}_{b,s}(d)$ with semantic dose d and $\mathcal{I}_{b,s}(0) = \mathrm{Id}$. A path must be inference-only, reversible, paired with the same seeds and trajectories as the baseline, and measurable at at least two nonzero doses. Examples include action scaling, controlled context retention, first-frame anchoring strength, and sampler-noise stress. Training losses and data reweighting are excluded because their optimization response is neither cheap nor dose-comparable across backbones.

Let $\boldsymbol{q}_x(\boldsymbol{d})$ denote the robustly standardized vector of goal, diagnostic, and validity outcomes under intervention dose vector $\boldsymbol{d}$. In chart q, paired central differences estimate the local response Jacobian

$$\boldsymbol{J}_x^q[:, s] = \left. \frac{\partial \mathbb{E}[\boldsymbol{q}_x(\boldsymbol{d})]}{\partial d_s} \right|_{\boldsymbol{d}=0} \approx \frac{\widehat{\boldsymbol{q}}_x(+\delta_s) - \widehat{\boldsymbol{q}}_x(-\delta_s)}{2\delta_s}. \tag{5}$$

For a one-sided path, the compiler records a normalized secant instead. A small dose grid checks that the response is locally stable; paths whose finite-difference slope changes beyond tolerance remain finite-dose probes rather than local directions.

The Jacobian induces a weighted IRG metric on the intervention frame,

$$\boldsymbol{G}_x^q = (\boldsymbol{J}_x^q)^\top \boldsymbol{W}_q \boldsymbol{J}_x^q + \lambda \boldsymbol{I}, \qquad \boldsymbol{r}_x^q = \mathrm{vec}(\boldsymbol{W}_q^{1/2} \boldsymbol{J}_x^q), \tag{6}$$

where $\boldsymbol{W}_q$ is fixed by the goal and outcome schema before the test campaign. Bootstrap repeats provide uncertainty $\boldsymbol{\Sigma}_x^q$. A chart records the goal and outcome schema, capability class, and supported intervention frame; compatible charts form the IRG. Static probes $\boldsymbol{z}_x$, capabilities $\boldsymbol{c}_x$, and missingness metadata condition the effect predictor. Starting from a small canonical frame, we retain an additional direction only when nested cross-domain replay shows that it improves operator selection enough to justify its probing cost.

A.3 Operator Descriptors and Compilation

An operator is represented by a descriptor $a = (\ell_a, \psi_a, \omega_a, d_a, s_a, \mathsf{pre}_a, \mathsf{inv}_a, \boldsymbol{\chi}_a)$, where ℓ_a is a required hook type, ψ_a is the transformation, ω_a is its scope, d_a is a behaviorally defined dose, s_a is a schedule, pre_a lists applicability conditions, inv_a lists semantic invariants, and $\boldsymbol{\chi}_a$ is a falsifiable prediction over operator response and horizon behavior. Examples include preserving the data split, applying a loss to the declared tensor, or restricting a sampler change to inference. The Operator registry stores these descriptors; the Operator outcome archive records their confirmed positive, null, or harmful outcomes by task type and capability class.

Each backbone exposes a capability vector $\boldsymbol{c}_b$ and a typed hook inventory. A compiler maps a to $\Gamma_b(a) = (\delta_a, r_a)$ when the preconditions and invariants are satisfied. The receipt records hook-contract tests, dose units and direction, outcome semantics, and invariant checks. For cross-backbone comparison, compatible receipts propose a partial intervention-axis map $\boldsymbol{T}_{i\to x}$ and outcome map $\boldsymbol{B}_{i\to x}$; the maps are then checked against paired response measurements.

A.4 Capability-Preserving Geometric Transfer

Compatible receipts define a shared chart and maps $\boldsymbol{T}_{i\to x}$ and $\boldsymbol{B}_{i\to x}$. We reject a map when paired anchor contexts do not preserve the measured response directions:

$$e_{\mathrm{align}}(i,x) = \frac{\|\boldsymbol{J}_x\boldsymbol{T}_{i\to x} - \boldsymbol{B}_{i\to x}\boldsymbol{J}_i\|_F}{\|\boldsymbol{J}_x\boldsymbol{T}_{i\to x}\|_F + \|\boldsymbol{B}_{i\to x}\boldsymbol{J}_i\|_F + \epsilon}. \tag{7}$$

Within a fixed shared chart, a simple fixed, uncertainty-normalized distance defines IRG neighbors:

$$d^2_{\mathrm{IRG}}(x,i) = (\boldsymbol{r}_x - \widetilde{\boldsymbol{r}}_i)^\top \boldsymbol{W}_q^{\mathrm{dist}}(\boldsymbol{r}_x - \widetilde{\boldsymbol{r}}_i) + \lambda_c d^2_{\mathrm{cap}}(\boldsymbol{c}_x, \boldsymbol{c}_i), \tag{8}$$

where tildes denote transported coordinates and $\boldsymbol{W}_q^{\mathrm{dist}}$ is fixed from the frozen, target-independent archive M_k^{full} before target evaluation, with noisy coordinates downweighted. The archive contains shared prior entries and all eligible pre-target source campaigns; target-specific records and post-evaluation updates are excluded. Each operator a defines a local effect function τ_a over this space, estimated only from campaigns satisfying the same operator contract. Thus transfer predicts a context-local effect, not the universal validity of a recipe.

A weighted local predictor gives mean effect $\widehat{\mu}_a(x)$ and uncertainty $\widehat{\sigma}_a(x)$. The geometry version, predictor, and operator set are frozen before an independent group-calibration split constructs a one-sided lower bound $L_a(x)$. Campaigns, not frames, are calibration units, and a changed IRG version invalidates the old bound.

Transfer prioritization uses an eligibility check, while effect acceptance comes from the subsequent target-side experiment. Protected violations $v_j \le 0$ are measured after execution and are not guessed from source records. Formally,

$$\begin{aligned}\mathsf{Cert}_{\mathrm{app}}(x,a) = \mathbb{I}[\mathsf{compile} = 1]\mathbb{I}[\mathsf{overlap} \ge \kappa_{\min}]\mathbb{I}[N_{\mathrm{eff}} \ge n_{\min}] \\ \cdot \mathbb{I}[e_{\mathrm{align}} \le \epsilon_{\mathrm{align}}]\mathbb{I}[\mathsf{sign_agree} \ge \gamma]\mathbb{I}[L_a(x) > \delta_G],\end{aligned} \tag{9}$$

The six terms cover compilation, support overlap, evidence, response alignment, sign agreement, and predicted gain. A failed term removes transfer priority for that candidate. The system can still reuse lower levels of infrastructure: L0 transfers the loop, L1 transfers compatible diagnostics, L2 transfers a compiled operator, and L3 transfers an effect prior under Eq. 9. In every case, the target experiment settles the effect.

Among valid candidates, the online selector is deliberately simple:

$$a_t = \arg\max_{a\in\mathcal{A}_t} \frac{\widehat{\mu}_a(x) + \beta_t\widehat{\sigma}_a(x)}{\widehat{C}(a)}. \tag{10}$$

Validity and capability constraints define $\mathcal{A}_t$ and are not traded against reward through tunable penalties. A transfer that passes these checks initializes the posterior; an abstention uses a cold prior.

We use progressive fidelity: a low-cost paired screen estimates effect direction, and promoted candidates receive paired multi-seed confirmation on held-out contexts. Screening updates search uncertainty; only confirmation creates a settled effect record.

A.5 Individual Operator Outcomes and Interactions

Operator combinations are useful for exploration but ambiguous for attribution. If a set S is promising, VERDI schedules matched contrasts using common seeds and examples. For operator $i \in S$, its local contribution is estimated from the available factorial contrasts

$$\widehat{\tau}_i = \sum_{A \in \mathcal{F}_i} w_A \left[\overline{Y}(A \cup \{i\}) - \overline{Y}(A) \right], \tag{11}$$

where $\mathcal{F}_i \subseteq 2^{S \setminus \{i\}}$ contains matched contexts. With all subsets, Eq. 11 recovers Shapley-style factorial credit; under a smaller allowance, leave-one-out effects are retained explicitly as conditional rather than universal effects. Pairwise interaction is stored separately as $\widehat{\tau}_{ij}^{\text{int}} = \overline{Y}(\{i,j\}) - \overline{Y}(\{i\}) - \overline{Y}(\{j\}) + \overline{Y}(\varnothing)$.

An entry becomes a *confirmed positive operator outcome* when its lower confidence bound exceeds the campaign threshold, all validity gates pass, protected metrics remain within tolerance, and the result replicates over the required seeds. Its context records the IRG chart, backbone capabilities, data regime, goal, and horizon range. The Operator outcome archive stores validated positive, null, harmful, and interaction outcomes, so later campaigns can use both successful and unsuccessful evidence.

A.6 Probe Refinement

A fixed probe basis can miss distinctions relevant to operator selection. We identify pairs of campaigns with the same goal schema and shared operator support that are close in IRG but have reliable, opposing outcomes for the same operator. Let

$$\begin{aligned} \mathcal{K}_\tau = \Big\{ (i,j) : d_{\text{IRG}}(i,j) \le \epsilon_{\text{IRG}}, \\ \exists a \in \mathcal{A}_i \cap \mathcal{A}_j : \\ \text{LCB}(-\tau_{i,a}\tau_{j,a}) > 0, \\ \min(|\tau_{i,a}|, |\tau_{j,a}|) > \epsilon_c \Big\}. \end{aligned} \tag{12}$$

Here $\tau_{i,a}$ denotes the utility effect $r_i^U(a)$ in the main text. The set above formalizes the opposing outcomes described in the main text, using a confidence bound on the negative effect product and a minimum-effect filter. An operational sufficient check is that the individual effect intervals lie on opposite sides of zero, each beyond ϵ_c, together with the sign-error screening criterion. A generic confidence bound is not assumed to be a posterior probability. Such a disagreement marks a distinction relevant to operator selection that the current chart has missed. In our audits of opposing outcomes, this can arise when a feature-conditioning operator improves soft-body scenes but degrades several metrics on rigid-body scenes whose static descriptions look similar. The system requests a new diagnostic direction when the disagreement survives uncertainty and schema checks. From candidate probes, it selects

$$s^\star = \arg\max_s \frac{\text{LCB}\Big[\widehat{\mathcal{R}}_{\text{nested}}(\mathcal{S}) - \widehat{\mathcal{R}}_{\text{nested}}(\mathcal{S} \cup \{s\}) \mid \mathcal{K}_\tau\Big]}{C_{\text{probe}}(s)}, \tag{13}$$

which directly targets future operator-selection errors rather than raw feature separation. Because disagreements are screened over many campaign pairs, we control the false discovery rate over $\mathcal{K}_\tau$ and shrink small-sample effects toward the chart mean before staging a direction. The inner replay learns the frame and effect predictor; disjoint promotion domains measure regret and calibration. Repeated search failure without such a disagreement instead points to a missing operator and prompts search for additional operators.

The executor searches network-accessible literature, documentation, and repositories for cross-domain ideas and synthesizes target-adapter code. A new candidate first runs in shadow mode under nested leave-one-domain/backbone-out replay. It enters the next registry version when it lowers selection regret and trials-to-first-positive without worsening calibration or the frozen regression battery. Registry snapshots, verdict-facing probes, and V_G stay fixed within a campaign, so discovery can expand the library without changing an ongoing evaluation.

Algorithm 1 IRG Learning and Selective Operator Transfer in VERDI

Require: Campaign $\mathcal{C}$, probes P_v, intervention frame $\mathcal{S}_v$, Operator registry $\mathcal{R}_v$, Operator outcome archive $\mathcal{E}$
Ensure: Verified operator $A^\star$, updated $\mathcal{E}$, staged capabilities

```
Instantiate and freeze verifier V_G; audit backbone capabilities c_b
Measure static covariates z and compile the supported intervention frame S_b ⊆ S_v
Run paired dose paths; estimate (J_x, G_x, Σ_x) and locality residuals
Query literature, documentation, and repositories with the frozen Optimization Fingerprint; type-check and shadow-stage candidates for R_{v+1}
for each operator a ∈ R_v do
    Compile a and chart maps; reject failed contracts or excessive map distortion
    Evaluate Cert(x, a); initialize a transported-field or cold posterior
end for
Let A be the remaining compiled candidates
while remaining nominal allowance is sufficient for a valid trial do
    Select a from A using Eq. 10
    Run the cheapest admissible paired screen; log response y(a) and receipts
    Update search uncertainty; keep screen evidence provisional
    if a passes the low-cost promotion rule then
        Run held-out multi-seed confirmation under frozen V_G
        if a is composite and remains positive then
            Schedule matched factorial or leave-one-out contrasts
        end if
        Mark only replicated, gate-valid positive operator outcomes as confirmed
        Archive the verifier-settled operator outcome, including null or harmful outcomes, in E
    end if
    if nearby models have reliable opposing operator outcomes K_τ then
        Stage the direction maximizing nested-replay regret reduction in Eq. 13
    else if search stagnates without such a disagreement then
        Stage a new operator for later campaigns
    end if
end while
Shadow-replay staged capabilities on disjoint promotion domains; queue only regret- and calibration-valid passes for version v + 1
return best verified operator A*, E, and the staged queue
```

What is learned. The result is an IRG together with verified operators, local outcomes and interactions, transfer eligibility checks, negative results, and replay-tested capabilities. Each settled experiment either improves the local effect model, marks a transfer boundary, or exposes opposing outcomes that guide the next IRG revision.

B Ablation and Audit Details

The complete LOBO design covers nine held-out backbones with three seeds each (27 campaign–seed units) and 216 transfer decisions (24 per target). Archive-prefix replay, main selector comparisons, and fixed-executor controls use all nine backbones. Separate ACWM-Phys diagnostics and the historical three-backbone retrieval controls in §B.12 retain their stated scopes.

Trials and GPU-h are averaged hierarchically: task and repeated-search values are first averaged within each campaign–seed unit, then over successful units within each backbone, and finally with equal weight across the nine backbones. Campaigns without a verified positive within budget remain in the completion denominator but are excluded from trial and compute means. Tables 3, 5, and 8 report completion counts. Descriptive 95% percentile-bootstrap intervals resample nine complete backbone clusters, retaining all within-backbone observations; the reported endpoints are the 2.5th and 97.5th percentiles (App. E.1).

Effect-sign accuracy compares the signs of predicted and observed utility changes under the target contract. Positive, null, and harmful are the verifier's final outcomes after checking the gain confidence

Table 6: **Backbone coverage of the LOBO study.** All nine named backbones serve as held-out targets, totaling 27 campaigns and 216 transfer decisions; source records exclude the target in each fold. All nine also enter archive replay and external-system comparisons. ACWM-Phys supplies a separate seed and diagnostic testbed.

Backbone / testbed	Evaluation role
Ctrl-World	LOBO, archive replay, and external comparison
Cosmos-Predict 2	LOBO, archive replay, and external comparison
Cosmos3	LOBO, archive replay, and external comparison
IRASim	LOBO, archive replay, and external comparison
OccWorld	LOBO, archive replay, and external comparison
DreamerV3	LOBO, archive replay, and external comparison
TD-MPC2	LOBO, archive replay, and external comparison
DINO-WM	LOBO, archive replay, and external comparison
V-JEPA 2	LOBO, archive replay, and external comparison
ACWM-Phys	Eight-environment seed and diagnostic testbed; outside the nine-target count

bound, goal threshold, and protected metrics; they are not identical to effect-direction labels. Coverage uses all 216 decisions. Negative-transfer risk counts harmful outcomes among accepted, executed transfers; accepted null outcomes remain in its denominator. Protected violations are measured over executed transfers.

Archive ablation convention. For held-out target t in fold k, the complete archive is denoted M_k^{full} and is frozen before target-side evaluation. It contains the shared prior archive and all eligible pre-target source campaigns; target-specific records, target-side calibration, tuning decisions, and post-evaluation updates are excluded. The archive ablation compares M_k^{full} against $M_k^{\text{empty}} = \varnothing$ (the cold-start arm). All other ablations keep M_k^{full} fixed and vary only the named component; source and target protocols, budgets, and verifiers remain unchanged.

Eight-environment testbed (ACWM-Phys). The fingerprint and distance analyses use an eight-environment testbed built on ACWM-Phys, a compact action-conditioned world model, spanning rigid-body contact (push/stack cube), deformable and granular media (cloth, rope, sand), fluid transport (pour water), and articulated dynamics (robot arm, reacher). Its size makes paired multi-seed confirmation, the exhaustive reference evaluation of §B.9, and the planted-fault audits of §B.8 feasible at full fidelity. The shared prior archive also contains settled campaigns from other backbones; together with eligible source campaigns, these entries constitute M_k^{full}. Table 6 lists all nine LOBO targets separately from the ACWM-Phys seed testbed. Each target can contribute source evidence in other folds, but its own records are excluded when it is held out. For held-out transfer, the archive snapshot and selector calibration are frozen before target-side evaluation, and target results are added only after the fold is settled. Cross-backbone behavior of the geometry is evaluated separately in the main campaigns (§4).

B.1 Backbone identities and evaluation contracts

The nine LOBO targets are distinct model interfaces. Every target uses the same contract layers: a frozen goal and verifier, paired baseline/operator-applied contexts, shared probe semantics translated by its adapter, and a target-specific set of protected outcomes. Ctrl-World, Cosmos-Predict 2, and Cosmos3 additionally have standalone case studies; IRASim, OccWorld, DreamerV3, TD-MPC2, DINO-WM, and V-JEPA 2 contribute to the full LOBO transfer analysis. Additional standalone evaluations for DreamerV3, TD-MPC2, DINO-WM, and V-JEPA 2 are grouped in App. E.5.

For the family-level analysis (App. E.4), we group Ctrl-World, Cosmos-Predict 2, and Cosmos3 as visual prediction; IRASim and OccWorld as action–observation/occupancy; and DreamerV3, TD-MPC2, DINO-WM, and V-JEPA 2 as latent dynamics/features. Operationally, their interfaces differ as follows. Ctrl-World, Cosmos-Predict 2, and Cosmos3 are the detailed visual or world-action studies: the first conditions visual prediction on robot actions, the second predicts future video observations with representation guidance, and the third couples visual prediction with action chunks for simulation and control. IRASim is an interactive action-conditioned simulator whose contract

Table 7: **Identity and role of each LOBO target.** The "role" column describes the model interface used by VERDI; the final column gives the reported evaluation scope.

Backbone	Role in the evaluation	Paper evidence and contract scope
Ctrl-World	Action-conditioned visual world model for robot manipulation	LOBO target, archive replay, and model-specific case study. The case study uses action/context/sampling probes, visual-quality coordinates, and protected dynamic-behavior checks.
Cosmos-Predict 2	Video/world-model predictor with representation and temporal prediction hooks	LOBO target, archive replay, and model-specific representation-guidance case study. The case study reports visual and structural coordinates together with the declared protected checks.
Cosmos3	World-action model for long-horizon simulation and control	LOBO target, archive replay, model-specific simulation study, and real-robot study. The reported contract includes rollout success, action-level diagnostics, and protected horizon/control checks.
IRASim	Action-conditioned manipulation world model	LOBO target; action–observation rollout contract and aggregate transfer analysis.
OccWorld	Occupancy-oriented world model interface	LOBO target; occupancy/free-space contract and aggregate transfer analysis.
DreamerV3	Latent-imagination control model	LOBO target; latent rollout/control contract and aggregate transfer analysis.
TD-MPC2	Latent-dynamics model for model-predictive control	LOBO target; latent dynamics and MPC-rollout contract and aggregate transfer analysis.
DINO-WM	Visual-representation world model interface	LOBO target; feature-space prediction/planning contract and aggregate transfer analysis.
V-JEPA 2	Video-prediction and representation model interface	LOBO target; temporal representation/prediction contract and aggregate transfer analysis.

is centered on action–observation rollouts. OccWorld predicts future scene occupancy from scene context and motion, so its contract uses occupancy and free-space consistency. DreamerV3 and TD-MPC2 are latent-dynamics control interfaces: their adapters expose observation/action histories, latent prediction, and control rollout evaluation. DINO-WM predicts and plans in pretrained visual feature space, while V-JEPA 2 provides a video representation/prediction interface whose shared measurements focus on temporal and action-conditioned response. The goal, utility scale, and protected thresholds are instantiated per target contract; the shared layer specifies the measurements used for cross-target transfer.

For each row, the adapter exposes the supported subset of action scaling, context retention, first-frame anchoring, and sampler-noise probes. A probe is retained only when its paired response has the required dose semantics and measurement support. Operator hooks are recorded separately from probe hooks: training, representation, sampling, and inference changes are compiled only when the target adapter satisfies the declared preconditions. Thus the shared contract defines what is compared, while the adapter defines how the same measurement is executed on a particular backbone. Per-target trial counts, protected outcomes, and fold-level decisions are reported as the audit units for the nine-target aggregate; the three detailed case studies add model-specific metrics and implementation traces.

Questions and evaluation quantities across targets. The nine-target design asks whether response-informed selection remains useful across different model interfaces, whether it reduces target-side trials and compute, and whether it controls negative transfer under each frozen goal contract. Effect-sign accuracy compares predicted and observed signs of utility changes under the target contract. Trials and $C_{\text{first}+}$ measure search to a verifier-confirmed positive; coverage is the accepted fraction of candidate transfers, and risk is the negative fraction among executed accepted transfers. The 216 decisions are repeated observations within the 27 campaigns. Raw utilities from different target tasks are reported within their own contracts. ACWM-Phys diagnostics and RoboCoin adaptation are separate studies from the nine-target LOBO count.



Figure 8: **Dose-response slopes across eight environments and seven probe directions (ACWM-Phys testbed).** All axes and the color scale are described here rather than in the panel. Each cell is the paired finite-difference slope of the primary utility U_G per unit dose—same seeds and trajectories, with and without a small dose—robustly standardized (§A.2); rows (top to bottom) are the eight environments (push cube, stack cube, push rope, cloth move, push sand, pour water, robot arm, reacher), columns (left to right) the probe directions (action amplify, action attenuate, self-rollout mix, motion scale, action-time mix, event align, action align). Blue marks a positive slope, orange a negative one; saturation encodes magnitude ($|z| \approx 2$: strong response; $|z| \approx 0$: the path does not move U_G). Paths outside the supported intervention frame are marked $\times$. Columns 1–2 are the two one-sided directions of the canonical action-scaling path, recorded as normalized secants (§A.2); columns 3–4 complete the canonical frame; columns 5–7 are diagnostic directions admitted in IRG v2 through probe refinement based on opposing operator outcomes, whose ablation is reported in main-text Figure 7.

B.2 Additional Fingerprint Diagnostics

The measured probe-response matrix and pairwise geometry below characterize the underlying representation; selector comparisons are reported in Table 3.

B.3 Agent-Level Controls Under Matched Budget

Matched agent controls. The component ablations are complemented by matched agent controls. All arms share the same frozen LLM, Operator registry snapshot, paired evaluation protocol, seeds, and nominal per-campaign allowance ($\mathcal{B} = 200$ GPU-h; measured overruns are retained). They differ only in the machinery used for proposal, licensing, and settlement. The bare agent proposes and executes operators from the registry, while the generic research-agent loop follows a propose–implement–run–observe cycle. The frequency prior and random selector provide simple references for historical frequency and uninformed choice. End-to-end metrics follow main-text Figure 6.

Table 8 complements the native-stack comparison in Table 4: it holds the executor fixed and changes only the search scaffold. Removing either IRG or eligibility checks increases failed transfer and protected violations. The frequency prior tests whether a single historically successful operator can substitute for target-specific selection.

Separate expert transfer case study. The human-expert row in Table 4 aggregates actual runs on the same 27 backbone–seed campaign units as the system comparison. The small case study below is reported separately. As an external reference, two engineers experienced with video world models were given the same registry, nominal allowance, and verifier on Ctrl-World. They reached

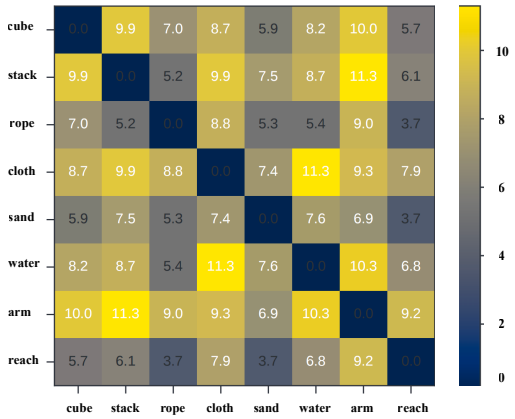

Figure 9: **Pairwise IRG distances between environments (ACWM-Phys testbed).** Cells show d_{IRG} from the full Optimization Fingerprint (Eq. 8); row/column order matches Fig. 8, and the neighbor threshold is 7.5. The geometry places sand and rope within range ($d = 5.3$) while separating the superficially similar cube and stack tasks ($d = 9.9$), defining the neighbors used for local operator-effect estimation (§A.4).

Table 8: Agent-level controls over nine backbones and 27 campaign–seed units, under a matched nominal allowance (200 GPU-h; same LLM and registry). Shared configurations match Table 5; Full VERDI is reported there.

Agent configuration	Done/27	Trials	GPU-h	Risk	Prot. viol.
w/o eligibility checks	21	4.40	132	.21	.14
w/o IRG	20	4.90	147	.18	.11
Bare LLM agent	18	6.00	189	.31	.26
Generic research-agent loop	16	6.60	210	.36	.29
Frequency prior	19	5.30	165	.29	.22

first-positive in 3.9 trials with protected violation 0.05— comparable to VERDI—but required approximately 14 hours of expert wall-clock per campaign and did not produce transferable records. Repeating the exercise on Cosmos-Predict 2 reset their search nearly from scratch (10.8 trials), whereas VERDI's certified transfer required 3.9 trials. The comparison points to the value of accumulating transferable evidence across campaigns, rather than a difference in single-campaign intuition.

B.4 Per-Backbone Transfer Breakdown

Table 9 reports all nine held-out targets under the same frozen-archive protocol. Trials and compute follow the hierarchical averaging rule above. Of 102 accepted transfers, 68 are positive, 28 null, and 6 harmful. Aggregate uncertainty is reported in App. E.1.

B.5 Auditable Cross-Backbone Transfer Trace

Table 10 follows one complete decision alongside the aggregate score. The Cosmos3 target and all of its records are hidden until the source archive, IRG version, thresholds, and operator ranking are frozen.

Table 9: **Nine-backbone LOBO results.** Each target has three campaign–seed units and 24 decisions. (a) Correct effect signs and trials/GPU-h to the first verified positive. (b) Accepted verifier outcomes, coverage, and conditional risk. Totals use all decisions; search means give equal weight to the nine backbones.

(a) Prediction and search efficiency

Target	Signs/24	Cold	Unconditional	Gated
Ctrl-World	17	10.1/286	3.8/104	3.3/84
Cosmos-Predict 2	21	12.4/344	4.5/132	3.9/108
Cosmos3	22	11.1/306	4.0/118	3.6/96
IRASim	21	10.7/298	3.6/102	3.5/91
OccWorld	17	13.2/361	4.8/142	4.4/126
DreamerV3	22	9.6/267	3.3/94	3.0/76
TD-MPC2	19	13.8/385	5.2/153	4.5/128
DINO-WM	20	11.6/320	4.0/117	3.7/101
V-JEPA 2	20	10.17/294	4.7/146	4.54/137
Total / macro-mean	179/216	11.41/318	4.21/123	3.83/105

(b) Gated outcomes and coverage

Target	Accepted (pos./null/harm.)	Coverage	Risk
Ctrl-World	11 (8/2/1)	11/24 = 45.8%	1/11 = 9.1%
Cosmos-Predict 2	10 (7/3/0)	10/24 = 41.7%	0/10 = 0.0%
Cosmos3	11 (8/2/1)	11/24 = 45.8%	1/11 = 9.1%
IRASim	12 (9/3/0)	12/24 = 50.0%	0/12 = 0.0%
OccWorld	11 (7/4/0)	11/24 = 45.8%	0/11 = 0.0%
DreamerV3	12 (6/5/1)	12/24 = 50.0%	1/12 = 8.3%
TD-MPC2	12 (8/3/1)	12/24 = 50.0%	1/12 = 8.3%
DINO-WM	13 (9/4/0)	13/24 = 54.2%	0/13 = 0.0%
V-JEPA 2	10 (6/2/2)	10/24 = 41.7%	2/10 = 20.0%
Total	102 (68/28/6)	102/216 = 47.2%	6/102 = 5.9%

Table 10: **One cross-backbone transfer trace.** The eligibility gate prioritizes a target experiment; the target-side verifier alone settles the effect and authorizes the archive update. The target outcome concerns the combined recipe and does not isolate the contribution of self-forcing.

Stage	Recorded evidence and decision
Held-out target	Cosmos3 long-horizon rollout campaign; target records hidden during retrieval.
Retrieved source	ACWM-Phys robot-arm campaign, IRG v5; confirmed the self-forcing component under the shared rollout-drift contract.
Fingerprint support	$d_{\mathrm{IRG}} = 0.43$; context-support overlap $= 0.74$; common horizon support through 24 rollout steps.
Ranked hypothesis	Self-forcing component, retrieved rank 1; the Cosmos3 target recipe combines it with RL, while imported evidence changes search order but supplies no target claim.
Six pre-trial checks	compile $= 1$; overlap $.74 \geq .60$; $N_{\mathrm{eff}} = 6.1 \geq 5$; alignment error $.11 \leq .15$; sign agreement $.83 \geq .75$; predicted-effect LCB $+2.3 > \delta_G = 1.5$.
Target screen	Paired low-cost screen: $\Delta U_G = +5.1$; promoted to confirmation, not written to the archive.
Held-out confirmation	Combined RL + self-forcing target recipe: paired three-seed effect $\Delta U_G = +6.8$ with 95% CI $[3.2, 10.4]$; all horizon and protected-coordinate gates pass.
Frozen-verifier verdict	Confirmed positive local effect for the declared context and horizon range.
Continual update	Write effect, uncertainty, support, cost, receipt, and artifact lineage to the Operator outcome archive; update the source–target outcome evidence for later campaigns.

Table 11: **Verifier ablation.** Error rates and verification cost (GPU-h) across verification policies.

Policy	False pos.	Harmful admit.	Protected viol.	Cost
Absolute score	.22	.19	.17	84
Unpaired	.18	.16	.14	91
Paired, no gates	.11	.09	.08	97
Paired + frozen verifier	.05	.03	.02	108
Full eligibility check	**.03**	**.01**	**.01**	114

Table 11 shows why settlement is separated from proposal. Unpaired or ungated policies admit more false positives and harmful effects; pairing reduces variance, while the frozen verifier and full eligibility check control protected violations. The extra compute pays for these checks.

B.6 Eligibility Check Leave-One-Term-Out

Eligibility checks. The six eligibility check terms guard distinct failure modes. We disable one at a time in the LOBO replay of Figure 6: failed compilation, insufficient support overlap, too little effective source evidence, response misalignment, sign disagreement, or an inadequate predicted gain. Post-trial protected-validity checks remain active throughout.

Table 12: Eligibility Check leave-one-term-out. Each row disables exactly one term of Eq. 9; the reported rate is the failure mode that term is designed to guard. The full-system reference is reported only in main-text Figure 6.

Disabled term	Exposed failure mode	Failure rate ↑
$\mathbb{I}[\mathsf{compile} = 1]$	compilation failure	0.19
$\mathbb{I}[\mathsf{overlap} \geq \kappa_{\min}]$	insufficient support overlap	0.17
$\mathbb{I}[N_{\text{eff}} \geq n_{\min}]$	insufficient evidence	0.15
$\mathbb{I}[e_{\text{align}} \leq \epsilon_{\text{align}}]$	response misalignment	0.28
$\mathbb{I}[\mathsf{sign_agree} \geq \gamma]$	sign disagreement	0.18
$\mathbb{I}[L_a(x) > \delta_G]$	insufficient predicted gain	0.24

Table 12 reports the failure rate exposed by each omitted check. Response alignment and the predicted-effect bound have the largest rates in this audit. Compilation and support checks also prevent wasted trials; a failed eligibility check falls back to a cold prior and the campaign can continue.

B.7 Validity-Gate Ablation and Threshold Sensitivity

Validity-gate sensitivity. The validity gates protect correlated partner metrics while the primary metric improves. We compare the full gate set with a version that removes the negative-correlation checks, and re-calibrate every tolerance τ at $0.5\times$ and $2.0\times$ its default ($1.5\times$ the baseline seed-level standard deviation).

Table 13: Validity-gate ablation and threshold sensitivity. Primary Δ is the mean primary-metric gain over gated campaign contexts; protected violation is the fraction of contexts where any protected metric exceeds tolerance; decision changed reports whether any campaign-level verdict (confirmed/null/rejected) flips relative to the default.

Configuration	Primary Δ ↑	Protected viol. ↓	Decision changed
Full gates ($\tau = 1.5\times$, default)	+12.78	**0.03**	–
No negative-correlation gates	+14.92	0.41	–
All tolerances at $0.5\times$	+11.86	0.04	no
All tolerances at $2.0\times$	+13.05	**0.03**	no

Table 13 shows the trade-off directly: removing the gates raises the primary gain from $+12.78$ to $+14.92$, while protected violations rise $13\times$. Re-calibrating all tolerances by a factor of two in either direction changes no campaign verdict, so the result is stable across this range.

B.8 Control-Plane Ablation Under Planted Faults

H/I/E denotes hypothesis proposal, implementation, and evaluation. Their separation keeps proposal and implementation authority distinct from the frozen evaluation and settlement authority.

Control-plane audit. The control-plane audit complements the geometric ablations. We construct 30 trap episodes (six per type) spanning five failure modes: evaluator leakage, within-noise gain, protected-metric sacrifice, confounded attribution, and a single-seed fluke. Ten additional episodes contain genuine, gate-valid improvements. Episodes are interleaved and presented without labels; each arm settles them as confirmed, null, or rejected. We report the false-conclusion rate and the missed-genuine rate, averaged over three seeds on Ctrl-World and Cosmos-Predict 2.

Table 14: Control-plane ablation on 30 planted-fault and 10 genuine-improvement episodes. False conclusion = trap settled as *confirmed*; missed genuine = valid improvement settled as *null/rejected*. Overhead is the mean extra compute per episode relative to direct settlement.

Control configuration	False conclusion ↓	Missed genuine ↓	Overhead (GPU-h/ep.)
Full control plane	**0.07**	0.10	+6.2
w/o H/I/E separation	0.28	0.09	+4.8
w/o validity gates	0.41	0.08	+5.1
w/o progressive fidelity	0.33	0.07	−3.4
No control plane (direct settle)	0.52	**0.04**	0.0

Table 14 shows the trade. Direct settlement accepts nearly every episode, but labels over half of the planted faults as confirmed effects. The full control plane reduces false conclusions by $7.4\times$ at a cost of 6.2 GPU-h per episode, with a modest increase in missed genuine effects. Removing H/I/E separation roughly quadruples false conclusions, especially on confounded-attribution traps. Removing progressive fidelity saves overhead but lets low-cost screens create confirmed records, which increases the single-seed failure mode.

B.9 Progressive-Fidelity Screening and Selector Ablation

Progressive-fidelity screening. The low-cost screen should retain effective operators while saving evaluation budget. We measure screen recall against exhaustive evaluation of all compiled candidates, then swap the selector for simpler alternatives while holding the screen fixed.

Table 15: Screening recall and selector ablation. Recall is the fraction of operators confirmed by exhaustive evaluation that the screen retains for confirmation; cost is the mean measured C_{search} per campaign. The exhaustive row is the recall reference, not a deployable policy.

Strategy	Screen recall ↑	C_{search} (GPU-h) ↓
Exhaustive evaluation (reference)	1.00	480
Screen + UCB selector (default)	**0.94**	**205**
Screen + kNN selector	**0.94**	214
Screen + random selector	0.91	268

Table 15 shows that the screen retains 94% of exhaustively confirmed operators at 43% of the exhaustive evaluation cost. The kNN and UCB rows have the same reported recall, with 214 versus 205 GPU-h. The comparison supports screening as a source of savings and separates that saving from the choice of acquisition rule.

B.10 Cost Ledger

Cost accounting. Fingerprinting and screening incur upfront cost; this ledger accounts for it per campaign. Avoided spend is estimated from the cold-start trajectory of the same campaign net of measured C_{overhead}; these case-specific costs are distinct from the nine-backbone means in Figure 6. This ledger is separate from the main-text measured $C_{\text{first}+}$.

Table 16: Per-campaign measured cost ledger (GPU-h). C_{overhead} = fingerprinting + screening + confirmation; it is reported separately from main-text $C_{\text{first}+}$. Net saved = avoided spend − overhead.

Campaign	Fingerprinting	Screening	Confirmation	Overhead	Net saved
Ctrl-World	9	51	36	96	+96
Cosmos-Predict 2	11	74	88	173	+139
Cosmos3	8	60	62	130	+132

Table 16 shows that fingerprinting costs only 8–11 GPU-h per campaign and is repaid within the first certified transfer; each listed campaign ends net-positive. Screening is the largest component for Ctrl-World, whereas confirmation is largest for the two Cosmos cases.

The three measured overheads sum to 399 GPU-h. Estimating the remaining 24 campaigns at 140 GPU-h each gives $399 + 24 \times 140 = 3{,}759$ GPU-h for the main LOBO study. Assumptions of 115–165 GPU-h per remaining campaign give 3,159–4,359 GPU-h. This is a sensitivity range for an extrapolation, not a measured total or confidence interval. Standalone downstream evaluations are excluded. This overhead estimate is reported separately from both $C_{\text{first}+}$ and C_{search}; the quantities are not added to produce a total.

B.11 Budget Alignment

Budget alignment. For the three detailed comparisons in Table 17, baseline and improved variants use identical training steps and data, with matched nominal hyperparameter-search allowance. Measured training compute and search overhead are reported separately, so differences in outcomes cannot be attributed to different schedules or search allowances.

Table 17: Training-schedule alignment and measured compute for the three detailed backbone–operator comparisons.

Campaign	Train steps	Data	Compute (GPU-h, base vs. improved)	HP trials
Ctrl-World	200k	identical	420 vs. 425	12
Cosmos-Predict 2	50k	identical	380 vs. 386	10
Cosmos3	50k	identical	390 vs. 394	10

B.12 Selector Source Ablation

Operator-text retrieval and the frequency prior provide two additional controls for the selector study. These historical controls use the detailed three-backbone subset: nine campaigns and 72 decisions, separate from the revised nine-backbone results in Table 3.

Complementary controls. These selectors test whether operator descriptions or historical success frequency suffice to prioritize operators. Table 3 compares metadata, raw responses, learned embeddings, and full fingerprints; Table 5 separately removes IRG while fixing the executor. The two ablations address the informativeness of the representation and the contribution of IRG to the complete search procedure, respectively.

Uncertainty across the detailed subset. For the three-backbone detailed subset, full-fingerprint 95% intervals are [0.73, 0.90] for sign accuracy, [3.2, 4.1] for trials, [0.02, 0.14] for negative transfer, and [0.01, 0.09] for protected violation. These backbone-cluster bootstrap intervals describe the original three-backbone subset (nine campaigns and 72 decisions), including its sign-accuracy point estimate of 0.83. They do not quantify uncertainty for the revised nine-backbone Table 3. Full-study intervals are reported in App. E.1.

Table 18: **Additional retrieval baselines on three backbones (9 campaigns, 72 decisions).** Only the selection signal changes across rows; the executor, Operator registry, nominal allowance, and verifier are held fixed.

Selection source	Sign accuracy ↑	Trials to positive ↓	Neg. transfer ↓	Protected violation ↓
Operator-text embedding	0.62	5.8	0.24	0.16
Frequency prior	0.57	5.2	0.29	0.22

C Additional Backbone-Specific Experiments

This appendix reports experiments and diagnostics not displayed in the main text: the eight-environment ACWM-Phys matrix, additional Ctrl-World metrics and views, Cosmos-family operators, downstream action generation, RoboCoin adaptation, and a probe-locality case study.

C.1 Extended Protocol

We treat VERDI as a research agent for a world model and evaluate its ability to propose operators targeted to the diagnosed failure of each backbone. To isolate the selected operator's effect, we preserve the original observation interface, tokenizer or VAE, decoder or renderer, data partition, and evaluation implementation unless a selected operator explicitly targets that component. The baseline is the original backbone before applying the operator, compared under the same data, matched training schedule, nominal search allowance, and evaluation protocol.

The instantiated operators differ by diagnosed mechanism. Ctrl-World (Guo et al., 2025) receives a LaMo (Jiang et al., 2026)-style latent-motion prior for insufficient physical consistency; Cosmos-Predict 2 (NVIDIA et al., 2025) receives DINOv2 (Oquab et al., 2024) representation guidance (REP) for structural drift; Cosmos3 receives a combined reinforcement-learning (RL) and self-forcing operator (Huang et al., 2025), evaluated jointly on long-horizon rollout and downstream control; and RoboCoin (Wu et al., 2025) uses an automatically constructed data and action-conditioning adapter. We report standard video metrics (PSNR, SSIM, LPIPS, and FID), WorldArena-style semantic, depth, flow, smoothness, and consistency metrics, and rollout diagnostics appropriate to each claim. Action-conditioned adaptation additionally includes a no-action control. Each subsection reports protected-coordinate checks alongside the gains.

C.2 Complete ACWM-Phys Optimization Results

Table 19 provides the complete eight-environment result behind the main-text summary. Deltas are computed as operator-applied minus baseline under the frozen ACWM-Phys evaluator. Positive PSNR and SSIM and negative masked MSE are favorable. The evidence column records whether a result comes from a single checkpoint or repeated receipts.

C.3 Ctrl-World Implementation and Additional Diagnostics

The original Ctrl-World produces plausible short-term predictions but exhibits motion drift, object-state jumps, and unstable long-horizon transitions. Its diagnostic response suggests that the failure is concentrated in latent state evolution rather than the visual encoder or renderer. VERDI therefore adds a LaMo-style branch containing a motion prior, delta-latent predictor, and drift readout. The objective combines reconstruction and drift consistency,

$$L_{\text{total}} = L_{\text{MSE}} + \lambda_{\text{drift}} L_{\text{drift}}, \tag{14}$$

where L_{drift} aligns the predicted latent displacement with the future motion direction.

These additional diagnostics show the trade-off of the motion operator: background consistency drops by 1.26 and dynamic degree by 1.54, while PSNR is approximately flat. The background regression remains within tolerance; dynamic degree exceeds tolerance on the open-drawer context group. That violation prevents a verified positive for a contract covering that group. The coordinate improvements therefore remain a descriptive trade-off for the current contract. A narrower claim would require a

Table 19: **Failure-specific optimization on all eight ACWM-Phys environments.** The selected operator differs across diagnosed failure families; status records the available confirmation boundary.

Environment	Failure family	Selected operator	Checkpoint/evidence
push_cube	rigid contact/action	cfg schedule $1 \rightarrow 1.1$	2 receipts
stack_cube	support/contact	cfg schedule $1 \rightarrow 1.5$	2 receipts
push_rope	topology/contact	self-forcing finetune	1 checkpoint
cloth_move	deformable memory	next forcing	800 steps
push_sand	granular transport	mixture reweight + scale 0.5	3 receipts
pour_water	fluid transport	event window + scale 0.25	metric pass
robot_arm	articulated dynamics	self-forcing finetune	800 steps
reacher	target-conditioned control	next forcing + scale 0.02	2 receipts

Environment	ΔPSNR ↑	ΔSSIM ↑	ΔmMSE ↓	Status
push_cube	+0.15	+0.0002	-0.000284	formal
stack_cube	+0.03	+0.0020	-0.000531	formal
push_rope	+0.02	+0.0008	-0.000151	formal
cloth_move	+0.94	+0.0099	-0.009040	formal
push_sand	+0.64	+0.0010	-0.000783	formal
pour_water	+0.85	+0.0019	-0.001718	formal
robot_arm	+1.08	+0.0098	-0.002230	formal
reacher	+0.06	+0.0002	-0.000119	formal

Table 20: **Additional Ctrl-World WorldArena metrics.** These three coordinates complement the primary metrics in Table 1. Baseline and operator use matched data and training schedules.

Metric	Ctrl-World	VERDI
Aesthetic Quality ↑	32.75	**36.12**
Background Consistency ↑	**86.37**	85.11
Dynamic Degree ↑	**45.32**	43.78

scope declared before evaluation or a fresh confirmation under a new frozen contract. The multi-view example below tests a different task from Figure 3.

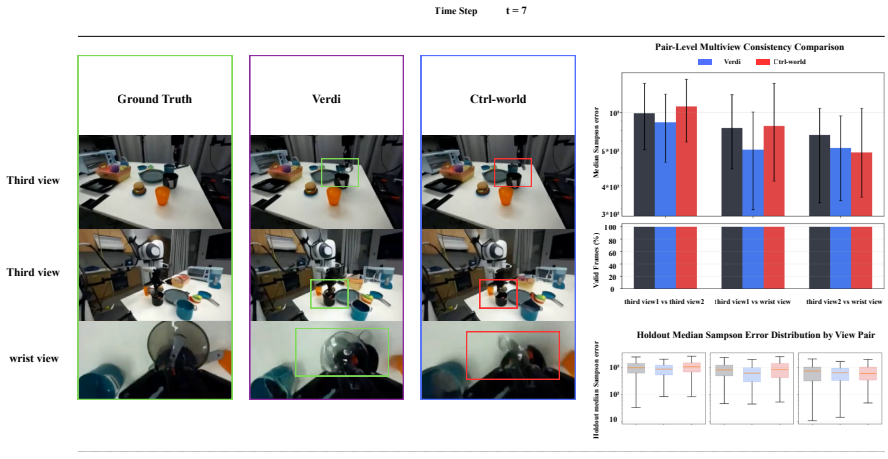

Figure 10: Consistency comparison across three camera views on the "put the lid on" task (DROID (Khazatsky et al., 2024)). Two views improve significantly over the Ctrl-World baseline; third-person view 2 and the wrist view degrade, consistent with the validity-gate analysis above.

Table 21: **Ctrl-World video quality.** Lower is better for LPIPS and FID.

Metric	Ctrl-World	VERDI
PSNR ↑	**20.5099**	20.4921
SSIM ↑	0.7799	**0.7899**
LPIPS ↓	0.1528	**0.1432**
FID ↓	54.0076	**51.7057**

Table 22: **Cosmos-Predict 2 results on WorldArena.** Baseline versus REP-guided operator under matched training schedules; measured compute is reported separately.

Metric	Cosmos-Predict 2	VERDI
Semantic Alignment ↑	90.38	**90.82**
Depth Accuracy ↑	95.35	**97.05**
Aesthetic Quality ↑	36.05	**37.30**
Background Consistency ↑	**85.35**	82.73
Dynamic Degree ↑	42.18	**43.15**
Flow Score ↑	29.55	**30.70**
Motion Smoothness ↑	78.93	**81.31**
Subject Consistency ↑	**83.07**	81.79

C.4 Structure and Long-Horizon Generation in the Cosmos Family

The Cosmos family exposes two different failure modes. Cosmos-Predict 2 shows structural drift under complex motion and occlusion, whereas Cosmos3 shows train–inference mismatch and compounding error during long rollouts. VERDI therefore selects representation guidance for the former and a combined RL + self-forcing operator for the latter rather than applying one global recipe.

Cosmos-Predict 2: representation-guided prediction. REP injects DINO features as an additional constraint so that prediction is aligned not only in pixel space but also in a higher-level structural feature space. The complete eight-coordinate WorldArena result is retained here in Table 22, while the main text uses Cosmos3 for the complementary long-horizon and control study.

Background consistency drops by 2.62 and subject consistency by 1.28. The background regression exceeds its tolerance on the BridgeData V2 (Walke et al., 2023) context group, so the current contract records the REP result as a mixed outcome. The archive keeps the regression together with the improved coordinates; a narrower claim would require a separately declared contract.

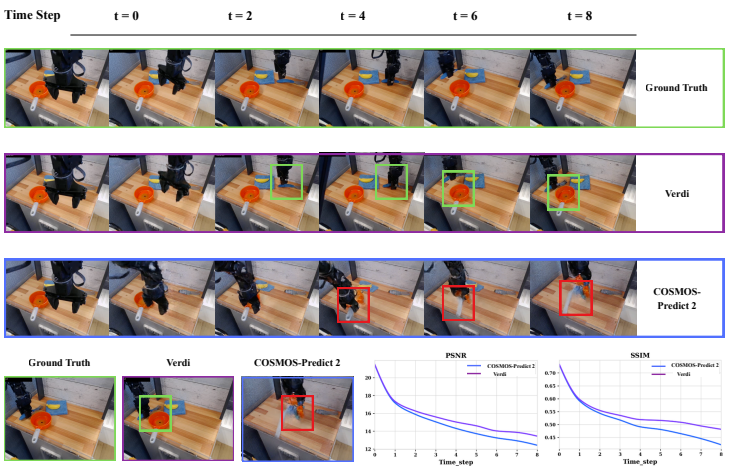

Figure 11: Qualitative comparison of Cosmos-Predict 2 and REP on "place the fork in the pot" (BridgeData): ground truth, baseline, and REP-improved prediction.

Table 23: **Simulated closed-loop manipulation success rate.** Vanilla versus Cosmos3 with the combined RL + self-forcing operator on paired simulation initial states. Each seed has 50 episodes per arm and task; entries are means across seeds and Δ is in percentage points.

Environment	Task	Avg. steps	Vanilla Cosmos3	With operator	Δ
Sim (50/seed)	Pick-and-place	8	62%	**74%**	+12.0
Sim (50/seed)	Drawer opening	11	41%	**59%**	+18.0
Sim (50/seed)	Peg insertion	14	28%	**52%**	+24.0

Table 24: **Action-level metrics.** Vanilla versus RL + self-forcing Cosmos3 on expert-action L2 error and the descriptive success gap between the 8-step and 14-step tasks. The gap mixes task and step-count differences.

Metric	Vanilla Cosmos3	With operator
Expert-action L2 error ↓	0.084	**0.061**
8-to-14-step task gap ↓	−34 pts	**−22** pts

Cosmos3: RL + self-forcing. Cosmos3 performs well in short rollouts but shows subject and background drift as generated states become its own context. The RL component targets downstream action outcomes, while self-forcing exposes the model to generated histories during training while preserving its architecture and input–output interface. Together they target action quality and the diagnosed train–inference mismatch.

Figure 3b shows the resulting long-horizon generation on "loosen the pot handle" (DROID (Khazatsky et al., 2024)). Action-level diagnostics are reported in App. C.5.

C.5 Simulation and Additional Action Diagnostics

The physical-robot evaluation is reported in §4.2 and Figure 4. Here we examine the complementary Franka simulation setting: pick-and-place, drawer opening, and peg insertion. Each seed uses 50 episodes per arm and task; the table reports the mean success percentage across the evaluated seeds, so entries need not be multiples of two percentage points. Vanilla and Cosmos3 with the selected operator use eight-step action chunks with re-observation after each chunk.

Applying the selected operator improves average simulation success by 18.0 percentage points in this three-task comparison. Expert-action L2 error decreases from 0.084 to 0.061, while the descriptive 8-to-14-step task gap narrows from 34 to 22 points. Because the two rows use different tasks as well as different step counts, this gap is not interpreted as a standalone estimate of error accumulation with horizon.

C.6 Bimanual AgileX Cobot Magic Qualitative Evidence

We additionally show representative paired rollouts on an AgileX Cobot Magic dual-arm robot. We conducted 40 validations for 3 task types, including "use of left and right hands", "pick and place", and "cooperation of left and right hands". As a result, the task success rate of VERDI increased by 19.6 percentage points. Red boxes mark the baseline's failed or misaligned interaction states, while green boxes mark the corresponding successful interaction states produced with the VERDI operator.

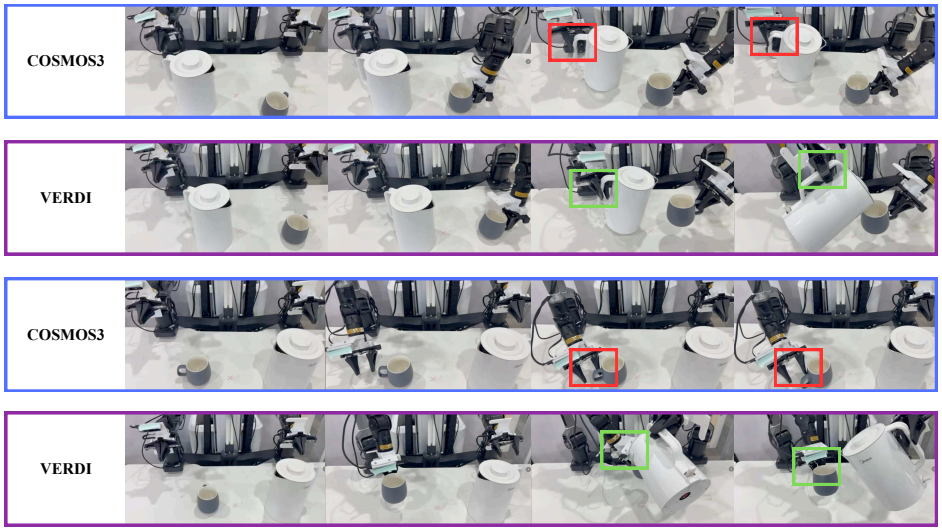

Figure 12: **Qualitative AgileX Cobot Magic dual-arm evaluation.** Representative paired frames for two manipulation sequences. The top and third rows show Cosmos3 baseline rollouts; the second and fourth rows show VERDI rollouts. Colored boxes identify the interaction state at the corresponding stage of each sequence.

C.7 Automatic Adaptation to RoboCoin

RoboCoin tests a different capability: converting a dataset with a new robot embodiment, action interface, camera layout, and task distribution into an executable world-model training task. VERDI parses episode boundaries, observations, action sequences, camera identities, temporal alignment, and embodiment metadata, then constructs a unified training and replay contract. It selects the DreamDojo/Cosmos-Predict 2.5 action-conditioned model initialized from AgiBot post-training weights and materializes the field mapping, action-conditioning, temporal sampling, and camera organization.

For single-view adaptation, the video stream is aligned with robot actions to form an action-conditioned future-video prediction task. The no-action control tests whether the prediction actually depends on the action input. Action conditioning improves held-out LPIPS by 0.029 and SSIM by 0.039 relative to this control. For multi-view adaptation, synchronized camera identities are retained rather than treating views as interchangeable; the reported cross-view subject-consistency variance is 0.029 versus 0.083 for independent per-view adaptation.

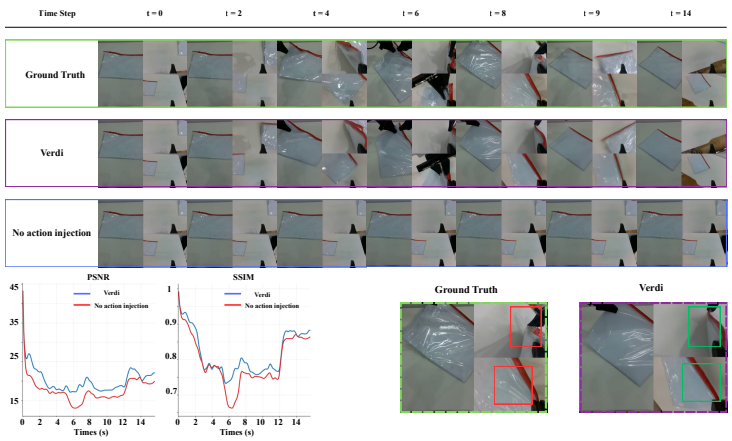

Figure 13: Action-conditioned adaptation on RoboCoin ("opening the red folder"): ground-truth future, action-conditioned prediction, and no-action control, organized in the same camera order and time frame across all three synchronized views. The gap between the action-conditioned and no-action rows evidences that the model uses the action input rather than unconditional visual continuation.



Figure 14: **Ground-truth vs. predicted frames on two RoboCoin tasks** (box relocation, top pair; cup pouring, bottom pair; two synchronized views per time step). VERDI-adapted Cosmos-Predict 2.5 (purple) tracks the ground truth (green) closely on both tasks, including object identity and multi-view correspondence.

This experiment gives one working adaptation contract for a single dataset family and base model. Broader embodiments and datasets remain future tests.

C.8 Probe Locality: A Counterexample Case Study

This case tests local chart validity, separately from the nested operator-selection regret replay in §4.5. The initial probe is local in five of eight environments. A cloth-move counterexample triggers an action-embedding temporal-mix probe, reducing the locality residual from 3.88 to 0.0196 and increasing held-out chart coverage from 5/8 to 6/8 (Figure 15). The result is a conditional chart-coverage improvement; downstream operator gains are evaluated separately in the frozen selector replay.

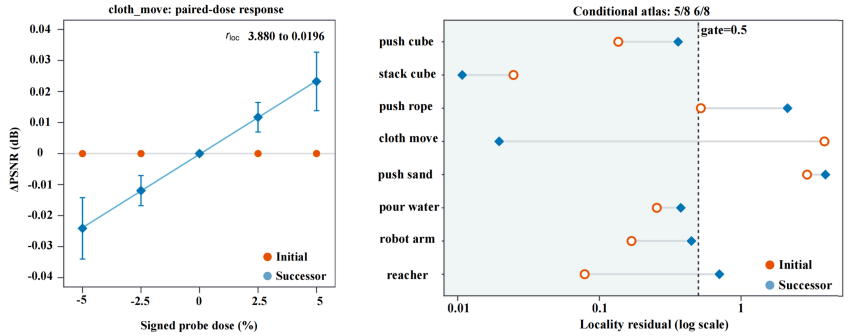

Figure 15: **Counterexample-driven probe evolution.** A non-local response triggers a temporal-mix probe. The library keeps both charts and admits the successor only where its paired-dose locality gate passes.

D LIMITATIONS

Although we validate VERDI on nine backbones with multiple random seeds, and further confirm its effectiveness in both real-world and simulated environments, our evaluation is constrained by limited human and computational resources. The nine backbones were chosen as representative, mainstream architectures rather than through an exhaustive enumeration of all possible backbones, and thus do not cover the majority of existing world models. In future work, we plan to collaborate with the community to extend our experiments and analysis to a broader range of backbone architectures.

E ADDITIONAL TRANSFER CONTROLS AND DOWNSTREAM EVALUATIONS

This appendix provides the full-study uncertainty estimates, matched-budget probe controls, response-informed baseline configuration, family-level outcomes, and standalone task visualizations. The main-study statistics use nine backbones, 27 campaign–seed units, and 216 decisions; standalone task evaluations are separate.

E.1 FULL-STUDY AGGREGATION AND UNCERTAINTY

Effect-sign accuracy is $179/216 = 82.9\%$, with a descriptive 95% backbone-cluster percentile-bootstrap interval of $[77.8\%, 87.5\%]$. Table 25 reports the search, coverage, and risk estimates. The resampling unit is one of nine complete backbone clusters. Trials and compute use the hierarchical mean specified in the introduction to Appendix B, rather than a mean of three integer candidate counts. Completion counts retain all 27 campaign–seed units. Unsuccessful campaigns are excluded only from trials/GPU-h means. Using unrounded macro-means, gated reuse reduces trials by approximately 66.5% and GPU-h by approximately 67% relative to cold start.

Figure 5 evaluates each chronological prefix on 27 campaign–seed units and 216 decisions, using nine backbone clusters for uncertainty. Figure 6 shows full-archive point estimates. The three-backbone intervals retained in §B.12 have a different scope and do not replace these full-study intervals.

E.2 MATCHED-BUDGET PROBE SELECTION

The original Fixed/Free/Gated comparison in Figure 7 changes both probe count and admission policy. To isolate selection, the three rules in Table 26 share seven probes, the candidate pool, replay/calibration admission thresholds, and the final evaluation set.

Table 25: **Nine-backbone aggregate results and descriptive 95% intervals.** Coverage and risk include explicit decision counts.

Strategy	Trials	GPU-h	Coverage	Risk
Cold start	11.41 [10.54, 12.33]	318 [295.4, 342.3]	—	—
Unconditional reuse	4.21 [3.83, 4.60]	123 [110.2, 136.1]	216/216 = 100%	73/216 = 33.8%
Gated VERDI	3.83 [3.49, 4.17]	105 [92.4, 118.4]	102/216 = 47.2% [44.9%, 50.0%]	6/102 = 5.9% [2.0%, 10.1%]

The protocol uses nine-fold outer LOBO. Each fold allocates four source backbones to discovery, two to promotion, two to calibration, and one held-out target to final evaluation. Three seeds per backbone give 12, 6, 6, and 3 campaign–seed units in the respective stages. Final evaluation covers nine backbone clusters.

Table 26: **Probe-selection controls at the same budget.**

Selection rule	Probe budget	Regret	Calibration error
Random candidate selection	7	0.30	0.15
Replay-gain	7	0.26	0.11
Contradiction-driven selection	7	0.21	0.08

Table 27: **Paired differences from contradiction-driven selection.** Positive values favor contradiction-driven selection; intervals resample nine backbone clusters.

Comparator	Metric	Difference	95% CI	Positive / 9
Random candidate selection	Regret	0.09	[0.078, 0.102]	9/9
Random candidate selection	Calibration error	0.07	[0.058, 0.087]	9/9
Replay-gain	Regret	0.05	[0.021, 0.070]	7/9
Replay-gain	Calibration error	0.03	[−0.002, 0.061]	6/9

These are unadjusted 95% intervals. The first three exclude zero. For calibration error relative to replay-gain, the interval includes zero: the improved point estimate alone does not establish a nonzero difference.

E.3 Response-informed baselines and risk–coverage

Table 28 compares selectors at the same accepted count, $102/216 = 47.2\%$. IRG has 6 harmful outcomes, compared with 12 for RGPE and 15 for the learned embedding. Sign accuracy remains agreement in predicted versus observed utility-change direction over all 216 decisions; it is distinct from the verifier's positive/null/harmful classification.

Table 28: **Selectors at matched coverage.** Protected violations are measured over executed transfers.

Selector	Sign accuracy	Trials	Coverage	Harmful / accepted	Prot. viol.
IRG	179/216 = 0.829	3.83	102/216 = 47.2%	6/102 = 5.9%	0.03
Response-informed RGPE	171/216 = 0.792	4.25	102/216 = 47.2%	12/102 = 11.8%	0.08
Learned embedding	160/216 = 0.741	4.60	102/216 = 47.2%	15/102 = 14.7%	0.10

At 180 accepted decisions, IRG risk is $45/180 = 25.0\%$. At full coverage, all methods accept the same 216 decisions and have the same 73 harmful outcomes. Risk generally increases with coverage, although it is not strictly monotonic at every reported point. The offline sweep uses prediction scores saved before verification outcomes are revealed; acceptance thresholds are not fitted to test outcomes. Accepted null outcomes remain in the risk denominator.

Table 29: **Risk–coverage counts on the same 216 outcomes.** Each method uses its own ranking, saved before outcome revelation. Divide harmful counts by accepted counts to obtain risk.

Accepted decisions	IRG harmful	RGPE harmful	Learned embedding harmful
43	2	4	5
72	5	7	11
102	6	12	15
144	18	29	39
180	45	53	61
216	73	73	73

Shared observation and budget rules. IRG and RGPE share the initial historical archive, compatible compiled-candidate pool, operator registry, probe set and query budget, executor, frozen verifier, and stopping rule. Sequential search may use only observations completed before the current decision. Additional target observations count against the same search budget. Source GPs use historical data within each held-out fold; the target GP is updated with available observations. GP fitting costs are recorded separately.

E.3.1 RGPE CONFIGURATION

Table 30 specifies the response-informed RGPE implementation. It predicts continuous utility changes, whereas the original frozen verifier determines the final positive/null/harmful outcome.

Table 30: **Response-informed RGPE configuration.**

RGPE component	Experimental setting
Source models and candidate space	One GP for each remaining source backbone with compatible historical observations, up to 8 per fold; evaluate only operator configurations within the same task-goal and response schema that pass the shared compiler's compatibility checks
Inputs and regression target	Budgeted response descriptors, semantically aligned operator types and parameter encodings, and missing-value masks; predict continuous utility changes under the original goal contract with consistent utility direction; scaling uses available training data only
GP and noise	Single-output exact GP, constant mean, ARD Matérn-5/2 kernel with an output-scale parameter, and float64; use measured repeat variance when available, otherwise learn Gaussian-likelihood noise
Fitting	Maximize the exact marginal log likelihood with L-BFGS-B, up to 200 iterations, and record convergence status
Weights	Estimate the probability of minimum ranking loss using 256 posterior samples; use leave-one-out predictions for the target model and split tied contributions equally
Cold start	With fewer than 2 valid target observations, combine compatible source models with equal weights without estimating weights from target ranking loss
Ranking	Rank by ensemble expected improvement relative to a predefined positive-utility threshold, selecting only within the fixed finite candidate pool
Verification and calibration boundary	The original verifier determines positive/null/harmful outcomes and protected metrics; posterior probability of exceeding the utility threshold does not itself establish confirmation, and any additional probability calibration is restricted to independent source calibration data
Resources and sweep	Observation queries count against the common budget and GP fitting costs are recorded separately; use scores saved before outcome revelation for an offline risk–coverage sweep at accepted counts of 43, 72, 102, 144, 180, and 216

E.3.2 LEARNED-EMBEDDING TRAINING AND CALIBRATION

Historical source-model fitting, sequential target-GP updates, and offline learned-embedding training are distinct stages. Table 31 describes the last of these. All 216 held-out decisions are excluded from

training and calibration; the RGPE target updates above instead follow the causal, prediction-before-outcome rule.

Table 31: **Learned-embedding features, labels, and calibration.**

Component	Protocol
Input features	Binned historical transfer cost in GPU-h, source-family encoding, campaign time intervals, cumulative transfer success/failure counts, and discrete backbone metadata; encoded features are concatenated into a fixed-length vector
Training target	Predict the effect direction of an individual transfer; labels are derived from the direction of observed utility changes under the target contract, and the logit sign gives the predicted direction
Loss	BCEWithLogitsLoss with inverse-frequency class weights computed from the training set
Calibration	Temperature scaling using an 8:2 training/validation split within the training data; only a global temperature parameter is fitted
Data separation	Historical time-window split; all 216 held-out decisions form a separate test set, and training and calibration exclude their samples, labels, and features

E.4 Cross-family outcomes

The visual-prediction family contains Ctrl-World, Cosmos-Predict 2, and Cosmos3; action–observation/occupancy contains IRASim and OccWorld; latent dynamics/features contains DreamerV3, TD-MPC2, DINO-WM, and V-JEPA 2. Table 32 reports eligible, accepted, positive, null, and harmful counts, in that order, for every source–target family pair.

Table 32: **Source–target family transfer.** Entries are eligible/accepted/positive/null/harmful. VP: visual prediction; AO: action–observation/occupancy; LF: latent dynamics/features.

Source/target	VP	AO	LF	Total
VP	18/9/6/2/1	18/8/6/2/0	36/11/6/4/1	72/28/18/8/2
AO	18/8/5/3/0	6/4/3/1/0	24/9/4/4/1	48/21/12/8/1
LF	36/15/12/2/1	24/11/7/4/0	36/27/19/6/2	96/53/38/12/3
Total	**72/32/23/7/2**	**48/23/16/7/0**	**96/47/29/14/4**	**216/102/68/28/6**

Within-family transfer accepts 40/60 candidates (66.7% coverage), with 28 positive, 9 null, and 3 harmful outcomes (7.5% risk). Cross-family transfer accepts 62/156 (39.7% coverage), with 40 positive, 19 null, and 3 harmful outcomes (4.8% risk). Positive outcomes constitute 40/62 (64.5%) of accepted cross-family transfers, or 40/156 (25.6%) of all cross-family candidates. Acceptance is not itself a successful outcome. The following four visualizations illustrate additional backbones within the latent dynamics/features family; the matrix provides the separate evidence for cross-family transfer.

E.5 Standalone downstream evaluations and case index

The original detailed studies cover Ctrl-World's latent-motion prior (§C.3, Tables 20–21, Figure 10), Cosmos-Predict 2's DINO-guided representation operator (§C.4, Table 22, Figure 11), and Cosmos3's joint RL/self-forcing intervention (§C.4–C.6, Tables 23–24, Figures 3(b), 4, and 12). Their protected-metric regressions and mixed outcomes remain part of the record.

We additionally group the DreamerV3, V-JEPA 2, DINO-WM, and TD-MPC2 task visualizations here. Table 33 links each intervention to its figure and evaluation unit; Table 34 reports the corresponding results. All are standalone evaluations, excluded from the 216 main-study decisions and the main-study overhead estimate.

Table 33: **Case-to-results index for the four additional visualizations.**

Backbone / task	Intervention	Figure panels	Evaluation unit
DreamerV3 / Walker	Teacher-guided policy distillation + DAgger	Fig. 16(a), Scene 1; (b), left	Cumulative return of displayed trajectory
DreamerV3 / Reacher	ReacherIK + PD-teacher supervision	Fig. 16(a), Scene 2; (b), right	Mean distance over displayed trajectory
V-JEPA 2 / RoboTwin	Action-MoE, refined by VERDI	Fig. 17(a–c): Pick and Place, Tool Use, Open and Close	100 episodes per task per method; Push and Pull quantified but not pictured
DINO-WM / Wall	Navigation-policy optimization	Fig. 18(a–b), Cases 1–2	Six paired start–goal conditions; shared checkpoint
TD-MPC2 / Nut	Trust-region-constrained MPPI	Fig. 19(a), Layouts 1–2	Two qualitative trajectory comparisons

Table 34: **Standalone downstream results.** RoboTwin values are episode success counts. DINO-WM uses six paired conditions. DreamerV3 metrics describe the displayed trajectories; TD-MPC2 is qualitative.

Backbone / task / metric	**Baseline**	**VERDI**	**Change**
V-JEPA 2 · Pick and Place	40/100 = 40%	75/100 = 75%	+35 pp
V-JEPA 2 · Tool Use	35/100 = 35%	42/100 = 42%	+7 pp
V-JEPA 2 · Push and Pull	21/100 = 21%	34/100 = 34%	+13 pp
V-JEPA 2 · Open and Close	32/100 = 32%	56/100 = 56%	+24 pp
V-JEPA 2 · RoboTwin overall	**128/400 = 32.0%**	**207/400 = 51.75%**	**+19.75 pp, approximately 61.7% relative improvement**
DINO-WM · Wall Navigation (Successful trajectories / evaluated trajectories)	1/6 = 16.7%	3/6 = 50.0%	+33.3 pp; 2 additional successful trajectories
DINO-WM · Wall Navigation (Mean final distance to goal)	13.289 grid units	8.135 grid units	38.8% reduction
DINO-WM · Wall Navigation (Mean distance to goal over the trajectory)	19.247 grid units	12.841 grid units	33.3% reduction
DreamerV3 · Walker Walk	Cumulative return 476.5	Cumulative return 655.4	+178.9 (approximately +37.5%)
DreamerV3 · Reacher Hard	Mean fingertip-target distance 7.4 cm	0.3 cm	95.9% reduction
TD-MPC2 · Nut Assembly	Before rollout	VERDI rollout	Qualitative trajectory comparison in Layouts 1 and 2

DreamerV3: Walker Walk and Reacher Hard. For DreamerV3, VERDI selects teacher-guided policy distillation with DAgger data aggregation for Walker, and ReacherIK with PD-teacher supervision for Reacher. The reported return and distance are measurements of the displayed trajectories, not means over a larger evaluation set. In the displayed Walker trajectory, cumulative return increases from 476.5 to 655.4 (+37.5%); the displayed Reacher trajectory's mean fingertip–target distance decreases from 7.4 to 0.3 cm. Figure 16 shows the corresponding behavior over time and both task-metric curves.

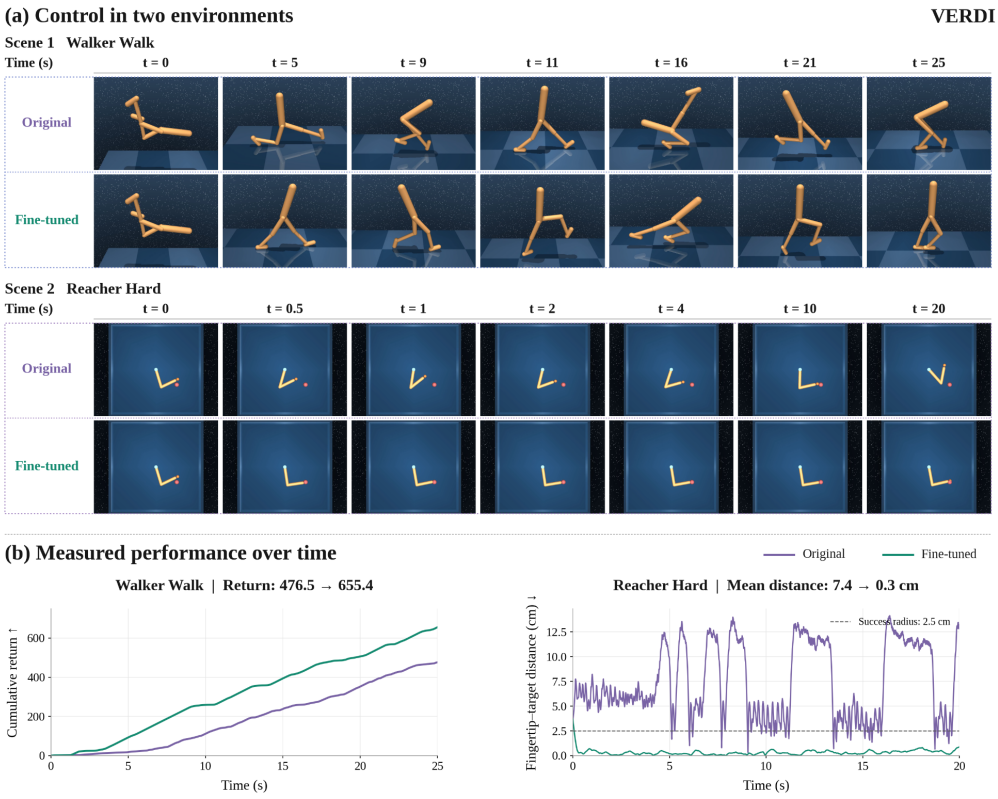

Figure 16: **DreamerV3 on Walker Walk and Reacher Hard.** Original and Fine-tuned denote the baseline and the VERDI-selected configuration. Walker uses teacher-guided distillation with DAgger; Reacher uses ReacherIK and PD-teacher supervision. Curves and reported return/distance describe the displayed trajectories.

V-JEPA 2: RoboTwin. For V-JEPA 2, task adaptation of an existing checkpoint on our RoboTwin data defines the baseline. VERDI selects Action-MoE and iteratively refines dense versus sparse routing, expert count, shared experts, and the router. The agent performs this refinement without manual hyperparameter tuning by the authors; the final configuration is frozen before evaluation. Four task categories use 100 paired evaluation episodes per method each. Overall success increases from 128/400 (32.0%) to 207/400 (51.75%), a gain of 19.75 percentage points. Tool Use improves by 7 points, less than the other three tasks.

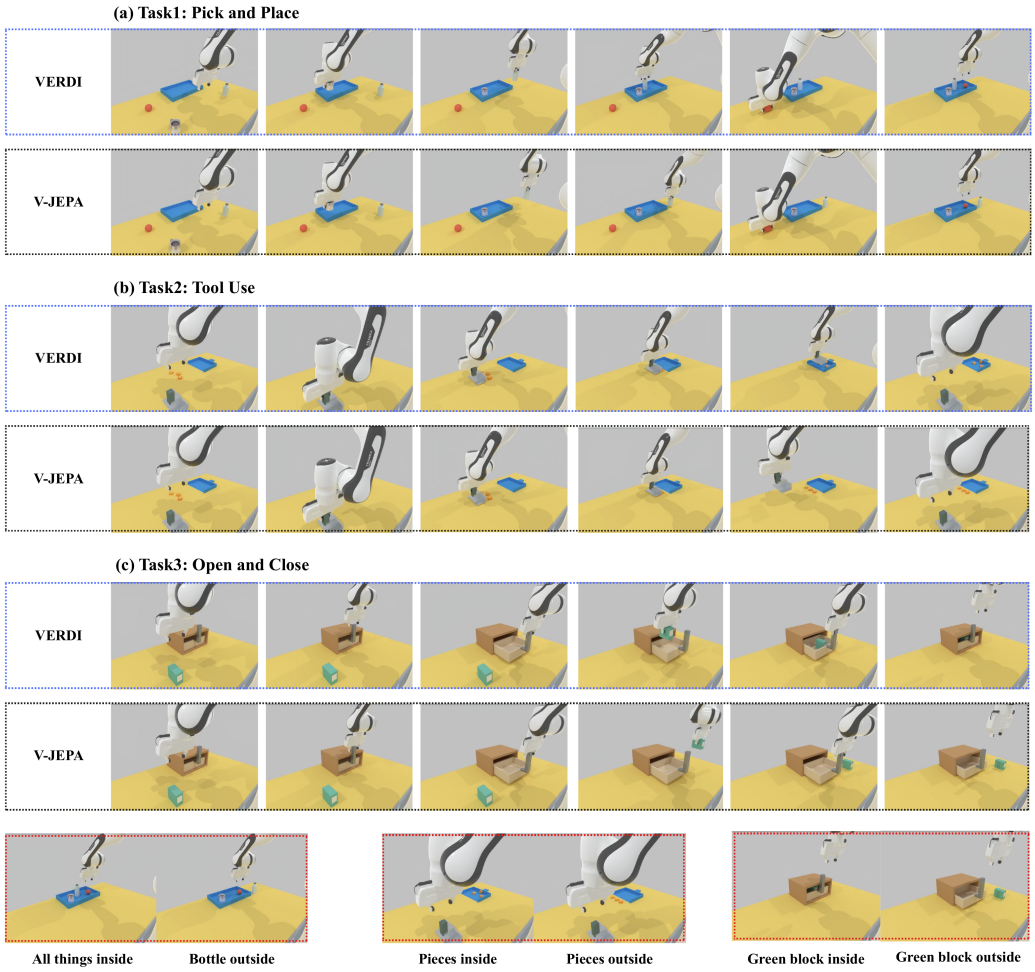

Figure 17: **V-JEPA 2 on RoboTwin.** Representative baseline and VERDI Action-MoE rollouts for Pick and Place, Tool Use, and Open and Close, with terminal-state comparisons. The full quantitative evaluation also includes Push and Pull (Table 34); that task is not pictured here.

DINO-WM evaluation. Each of six start–goal conditions uses the same checkpoint, initial and goal states, and planning budget for both methods, with one trajectory per method (12 trajectories in total). These are evaluation conditions, not six independent training runs. Success is determined by the simulator's true position at control step 100:

$$d_{100} = \sqrt{(x_{100} - x_g)^2 + (y_{100} - y_g)^2}, \qquad \text{Success} = \mathbf{1}[d_{100} < 4.5].$$

Distances use the environment's grid coordinates. Earlier entry into the goal region does not count if the final state leaves it. Neither collision count nor path length is an additional success condition.

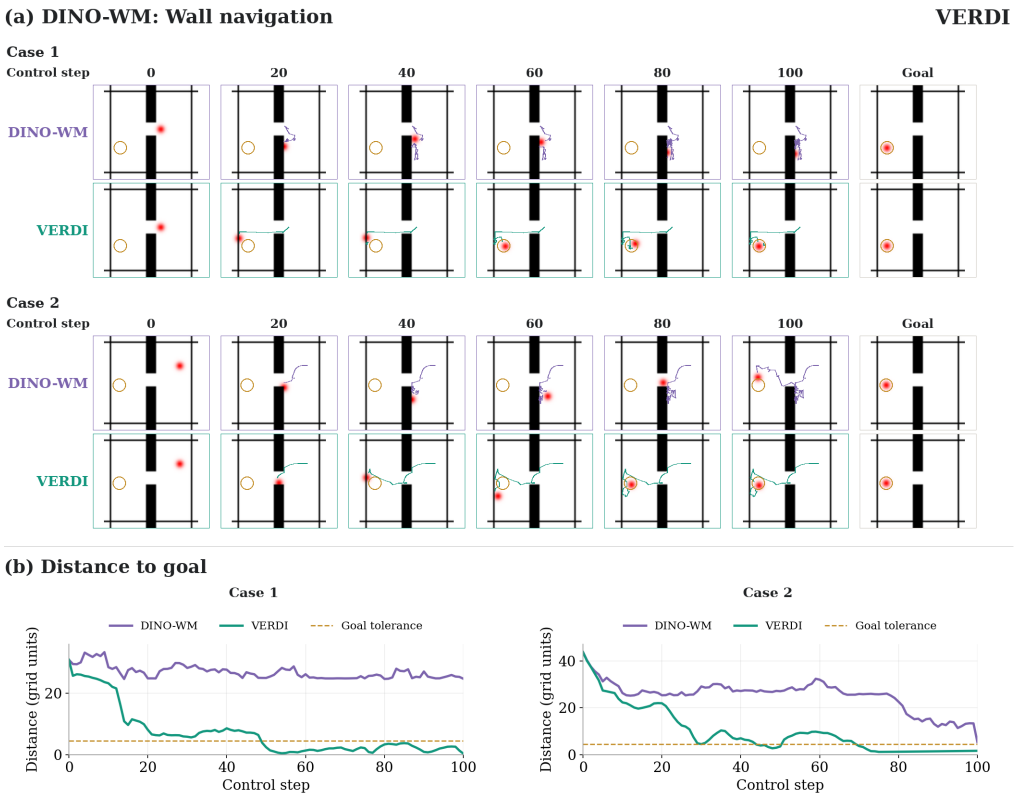

Figure 18: **DINO-WM Wall Navigation.** Two paired start–goal examples and their distance curves. The final-state success threshold is 4.5 grid units at control step 100. The complete six-condition results, including failures, appear in Table 35.

Table 35: **All six paired DINO-WM terminal-distance results.** Evaluation identifiers label start–goal conditions, not independent training runs.

Evaluation condition (seed)	Baseline final distance	VERDI final distance	Baseline success	VERDI success
10	14.707	15.831	No	No
11	24.727	0.337	No	Yes
12	5.306	1.640	No	Yes
13	14.439	15.037	No	No
14	18.858	14.990	No	No
15	1.695	0.973	Yes	Yes
Mean final distance / success count	13.289	8.135	1/6	3/6

The figure displays two improved cases. Case 1/2 identify the displayed examples; evaluation identifiers 10–15 label the complete paired set in Table 35. The mean final distance falls from 13.289 to 8.135 grid units, and mean distance over the trajectories from 19.247 to 12.841. Conditions 10 and 13 have slightly larger final distances after intervention; the aggregate improvement does not imply improvement on every trajectory.

The six-condition success fraction is descriptive and is not a 400-episode success-rate estimate. Likewise, the standalone outcomes are not added to the 24 main-study decisions or three LOBO campaign–seed units for any backbone. Their corresponding main-study counts remain in Table 9.

TD-MPC2: Nut Assembly. For TD-MPC2 Nut Assembly, the selected intervention is trust-region-constrained MPPI; the figure compares trajectories in two layouts. The before/after rows in Figure 19 follow each layout from the initial state through grasping and alignment. In Layout 1, the baseline remains misaligned over the displayed interval; in Layout 2, alignment occurs later than under the VERDI-selected configuration. The curves accompany these two trajectory comparisons and do not estimate an aggregate success rate.

This standalone illustration is separate from TD-MPC2's main-study results: 19/24 correct effect signs, 12/24 accepted transfers, and one harmful outcome among the 12 accepted transfers (Table 9). The main-study gated search mean is 4.5 trials and 128 GPU-h. These quantities describe the LOBO search procedure, whereas the figure shows behavior after the selected planning intervention.

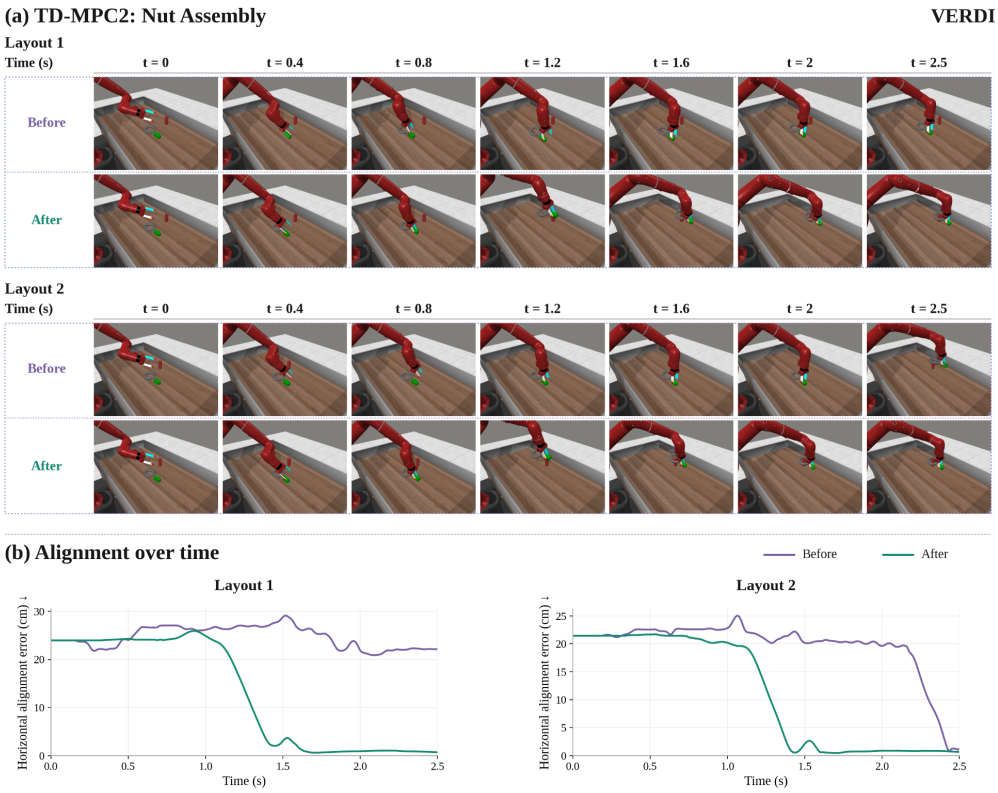

Figure 19: **TD-MPC2 Nut Assembly.** Baseline and VERDI-selected trust-region-constrained MPPI trajectories in two layouts, with the corresponding alignment curves. These are qualitative rollout comparisons; no aggregate task success rate is inferred from them.