VID2SID: Videos Can Help Close the Sim2Real Gap

Kevin Qiu[1,2], Yu Zhang[3], Marek Cygan[1,4], Josie Hughes[3]

[1]University of Warsaw [2]IDEAS NCBR [3]EPFL [4]Nomagic
kevinxqiu@gmail.com

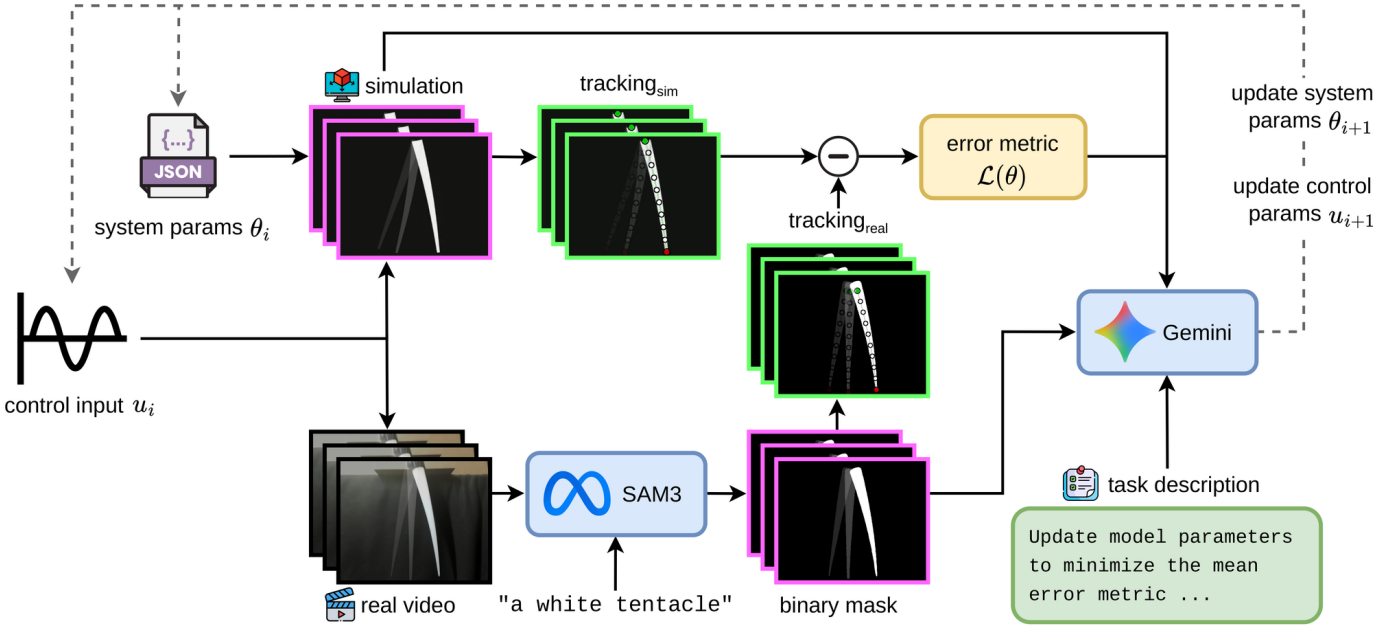

Fig. 1: Overview of VID2SID. Given paired sim-real videos, a perception layer extracts trajectories (SAM3 centerlines for soft robots, marker tracking for rigid robots). A VLM diagnoses discrepancies and proposes physics parameter updates with natural-language rationales. The process converges within 10 iterations, matching or exceeding black-box optimizers while providing interpretable reasoning at each step.

***Abstract*—Calibrating a robot simulator's physics parameters to real hardware is a prerequisite for generating useful synthetic training data. When sensing is limited to external cameras, perception noise and the absence of direct force measurements make the problem harder. We present VID2SID, a video-driven system identification pipeline that uses a vision-language model (VLM) to analyze paired sim–real videos, diagnose physical mismatches, and propose parameter updates with natural-language rationales. We evaluate on a tendon-actuated finger (4 MuJoCo parameters) and a deformable continuum tentacle (4–6 PyElastica parameters) in air and underwater. On holdout controls unseen during calibration, VID2SID achieves the best average rank across settings (1.7 vs. 2.7 for CMA-ES) and recovers ground-truth parameters with under 13% relative error (vs. 28–98% for baselines). An underwater stress test exposes model-class limits: when the simulator cannot represent real fluid dynamics, every optimizer converges to the same error floor, so the simulator architecture, not the parameters, bounds synthetic data quality. VID2SID's diagnostics identify this regime automatically, telling practitioners what to change and why.**

I. INTRODUCTION

Physics-based simulation generates synthetic training data at scale [1, 2], but policies trained on miscalibrated simulators fail on hardware. System identification (sysid) calibrates simulator parameters (friction, stiffness, damping) to close this gap [3]. Accurate calibration directly determines the fidelity of synthetic rollouts used for downstream policy training.

Black-box optimizers (Bayesian optimization [4], CMA-ES [5]) treat sysid as scalar minimization: reduce the error between simulated and real trajectories. They work, but they require hand-designed error metrics per platform and give no insight into which physical parameters are miscalibrated. When a policy fails after sim2real transfer, practitioners cannot tell whether the issue is parameter mismatch or a fundamental simulator limitation.

VID2SID replaces this pipeline with a VLM [6] that directly compares simulated and real videos. The VLM diagnoses

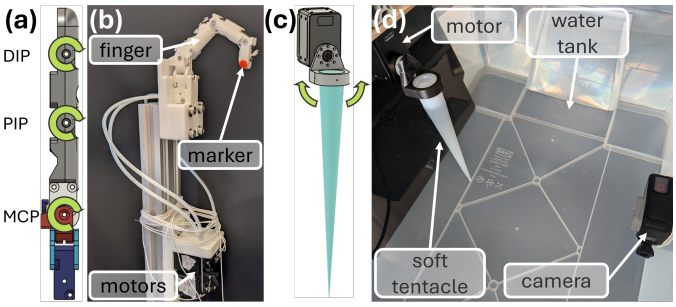

Fig. 2: Hardware platforms. Left: tendon-actuated finger with tracked marker (MuJoCo, 4 params). Right: silicone continuum tentacle in a water tank with overhead camera (PyElastica, 4–6 params).

concrete physical mismatches (e.g., "the simulated finger is too heavily damped; reduce friction and damping") and proposes targeted corrections. Each iteration produces a physical diagnosis as a byproduct.

II. Method

VID2SID operates in a closed loop. At each iteration: (1) the current simulator parameters generate a video of the robot executing a control sequence; (2) a perception layer extracts quantitative trajectories from both sim and real videos (marker tracking for rigid robots, SAM3 centerlines for soft robots); (3) a VLM receives the paired videos plus a structured prompt containing the current parameters with their bounds and an iteration history; (4) the VLM outputs updated parameter values with a natural-language diagnosis explaining the proposed changes.

Example diagnosis. A first-iteration VLM output on the finger (lightly paraphrased):

> *"The simulation finger is significantly slower and more heavily damped than the real hardware. The real tip oscillates through four cycles while the simulation barely completes two. Reduce friction from* $143 \rightarrow 25$ *and damping from* $190 \rightarrow 30$*. Leave stiffness unchanged until the damping correction lands."*

This single step reduces holdout error by 59%. The diagnosis identifies which *physical phenomenon* is being corrected and which parameters should be held fixed to avoid interaction effects.

We track the best parameters seen across all iterations, allowing the pipeline to benefit from early bold corrections without being degraded by later missteps as the remaining discrepancies grow subtler.

III. Experimental Setup

We evaluate on two platforms spanning rigid and deformable dynamics (Fig. 2):

Tendon-actuated finger (MuJoCo [7], 4 parameters: friction, damping, stiffness, tendon routing). A camera records the fingertip as the motor executes sinusoidal profiles. Markers are tracked at 30 fps.

TABLE I: Sim2real holdout error across platforms. Mean $\pm$ std over 3 seeds. VID2SID achieves the best average rank across all settings.

Method	Finger (px)	Tent.-Air (px)	Tent.-Water (px)	Rank
Random	27.8 ± 17.2	54.3 ± 1.9	76.1 ± 3.8	4.3
Golden-CD	12.1 ± 0.2	53.4 ± 4.8	77.8 ± 3.7	3.3
BO	16.1 ± 2.4	62.4 ± 10.4	$\mathbf{71.7 \pm 0.1}$	3.7
CMA-ES	15.3 ± 4.9	$\mathbf{52.1 \pm 2.9}$	77.6 ± 3.4	2.7
VID2SID	$\mathbf{10.9 \pm 6.3}$	53.0 ± 5.3	73.3 ± 3.1	**1.7**

Silicone continuum tentacle (PyElastica [8], 4–6 parameters: Young's modulus, density, damping, gravity). Tested in air (4 params) and underwater (6 params, adding fluid density and drag). SAM3 [9] extracts body centerlines from video.

Each method runs for 10 iterations. We evaluate on 4 holdout control profiles unseen during calibration, with 3 hardware repeats each (12 datapoints per seed, 3 seeds). We also run sim2sim validation where ground-truth parameters are known.

IV. Results

A. Sim2Real Holdout Generalization

Table I summarizes holdout generalization across all three settings. VID2SID achieves the best average rank (1.7 vs. 2.7 for CMA-ES). On the finger, VID2SID reduces holdout error by 10% compared to Golden-CD (10.9 vs. 12.1 px; Fig. 3). On the tentacle in air, performance is within 2% of CMA-ES (53.0 vs. 52.1 px).

B. Model-Class Limits

The underwater tentacle setting is a deliberate stress test. Under the shared-body protocol, all methods start from body parameters discovered by VID2SID during in-air calibration and tune only environment parameters. All methods cluster between 71.7–80.7 px with overlapping uncertainty, despite different search strategies. The narrow spread (range $<$10 px) indicates that optimizer choice matters less than the simulator's model class. The simplified drag model cannot capture fluid-structure interaction, so parameter tuning saturates regardless of method.

This finding is directly relevant to synthetic data quality: VID2SID's diagnostics can identify when the simulator itself, not the parameters, is the bottleneck. In such cases, generating more synthetic data or tuning parameters further will not improve downstream policies.

C. Sim2Sim Parameter Recovery

In a controlled sim2sim setting where ground-truth parameters are known (Table II), VID2SID recovers parameters with under 13% mean relative error (8.7% on finger, 12.4% on tentacle vs. 28–98% for baselines). Black-box methods often find compensating parameter combinations far from the true values. VID2SID's parameter-level accuracy means the calibrated simulator is physically faithful, not just error-minimizing.

TABLE II: Sim2sim holdout error. Finger error in mm (world coordinates); tentacle in px (centerline). VID2SID recovers physically correct parameters, not compensating combinations.

Method	Finger (mm) ↓	Tentacle (px) ↓
Random	15.67 ± 13.80	10.63 ± 9.59
Golden-CD	16.36 ± 17.79	11.44 ± 12.44
BO	9.70 ± 7.49	10.02 ± 11.03
CMA-ES	23.77 ± 19.23	10.02 ± 7.15
VID2SID	$\mathbf{7.13 \pm 2.56}$	$\mathbf{8.91 \pm 9.52}$

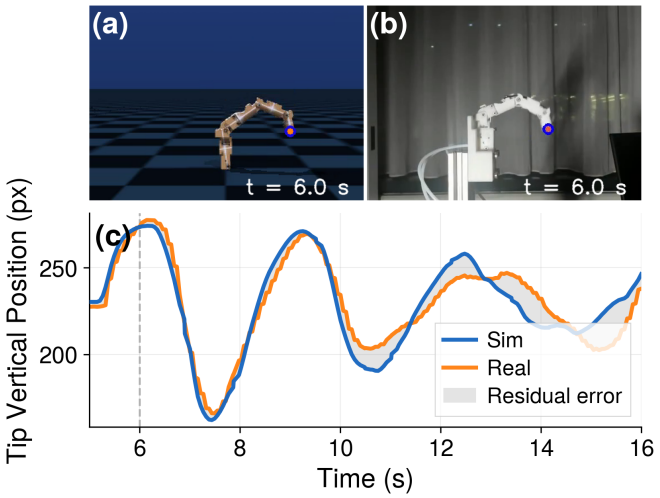

Fig. 3: Finger alignment after VID2SID calibration. (a) Simulated finger at t=6 s. (b) Real video frame with tracked marker. (c) Tip position over time: calibrated simulation (blue) tracks real motion (orange) with narrow residual (shaded).

D. Ablation Study

On the finger (Fig. 4), removing video input increases holdout error by 66%, showing that the VLM extracts physical information from raw frames that scalar metrics cannot provide. Disabling step-by-step reasoning in the prompt increases error by 13%, and removing iteration history actually *decreases* error by 10%, suggesting that history can anchor the VLM to early suboptimal iterations rather than encouraging fresh reasoning.

On the tentacle, the pattern reverses: removing video, history, or iterative refinement each *reduces* error by 38–40%. We attribute this to parameter observability. Finger parameters (damping, friction) produce visually distinctive signatures (oscillation speed, settling time) that the VLM reliably perceives. Tentacle material properties (Young's modulus, density, damping) produce subtler deformation effects obscured by SAM3 segmentation noise (∼3–5 px centerline jitter). The takeaway is that prompt design is platform-dependent: video helps when perception is clean, and hurts when segmentation is noisy.

E. Convergence and VLM Reasoning

Training error plateaus within 5 iterations on both platforms, matching or beating the convergence rate of black-box baselines. The VLM's self-reported confidence scores are well calibrated: recommendations succeed 89% of the time at confidence ≥ 0.9, giving practitioners a built-in reliability signal that scalar optimizers cannot provide.

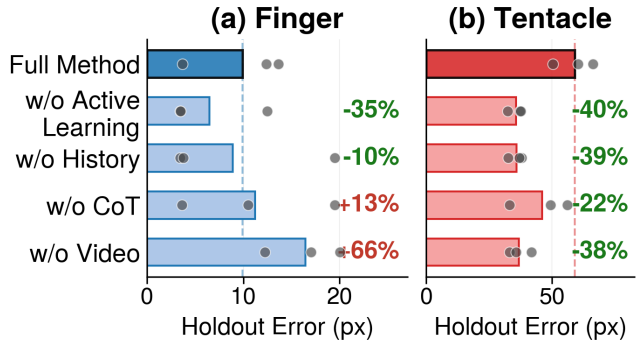

Fig. 4: Ablation study on (a) finger and (b) tentacle. Each bar removes one component. Dots show individual seeds. Removing video increases error by 66% on the finger but decreases it by 38% on the tentacle, showing that optimal prompt design is platform-dependent.

VLM consistency. Repeating identical inputs at temperature 1.0, the VLM shows high directional agreement: for each parameter, the majority of calls agree on whether to increase or decrease its value. Stochasticity primarily affects step size and diminishes as calibration progresses.

V. Related Work

Bayesian optimization [4, 10], CMA-ES [5], and random search minimize scalar error metrics. They require hand-designed objectives per platform and produce no diagnostic information. VID2SID requires only video rendering, not internal gradients, and the VLM's output at each iteration is a physics explanation, not just a parameter delta. Prior VLM applications in robotics have targeted planning, manipulation, and reward design; to our knowledge VID2SID is the first to use a VLM as a physics-reasoning optimizer that diagnoses sim–real discrepancies from video.

VI. Discussion

VID2SID improves simulator fidelity directly from video without hand-designed error metrics or per-platform engineering. The same pipeline handles rigid MuJoCo and deformable PyElastica without modification, and the written diagnoses help practitioners decide whether to keep tuning or invest in a better simulator model class.

Three calibration regimes. The experiments expose three regimes determined by perception quality and simulator expressiveness. (1) Clean perception + expressive simulator (finger): VLM video reasoning drives the strongest gains (10% holdout improvement, 66% ablation benefit from video). (2) Noisy perception (tentacle in air): text-only prompts outperform video because SAM3 jitter (∼3–5 px) obscures visual

signals. (3) Model misspecification (underwater): all optimizers land within a 10 px band, so the simulator architecture, not the parameters, bounds performance.

Computational cost. VLM inference adds $\sim$11 s per iteration ($\sim$28% overhead, \$0.14 per 10-iteration run), keeping total calibration under 10 minutes. This overhead is acceptable for offline simulator calibration but precludes real-time adaptation.

Implications for synthetic data. A calibration failure has two causes: the optimizer got stuck, or the simulator model class cannot represent the real system. Scalar sysid methods give the same error number in both cases. VID2SID's written diagnosis distinguishes the two, a routing signal that determines whether the next step is more compute or a new simulator. The sim2sim validation (Table II) shows that VID2SID recovers physically correct parameters (under 13% relative error), not compensating combinations far from the true values. This distinction matters for synthetic data: a simulator with the right physics produces rollouts where contact timing and dynamic responses match reality, so policies trained on this data generalize to held-out scenarios.

Limitations. VID2SID inherits VLM sampling stochasticity (std = 6.3 px on finger vs. 0.2 px for Golden-CD) and is evaluated with Gemini 2.5 Pro only. The pipeline assumes a single fixed camera with a static background. VLM recommendations are not guaranteed globally optimal; we observed occasional convergence to local minima that deterministic methods avoided. When the simulator model class is insufficient, no amount of calibration will close the gap, and the next step is a structural simulator upgrade rather than more parameter search.

Future directions. Distilling VLM reasoning into a lightweight model would enable real-time online calibration. Extending to multi-view and multi-robot settings with occlusions, and fine-tuning foundation models for physics reasoning to reduce prompt sensitivity, are open problems. The same video-conditioned reasoning loop could apply to reward design and policy debugging.

REFERENCES

[1] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 23–30.

[2] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray *et al.*, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.

[3] F. Muratore, F. Ramos, G. Turk, W. Yu, M. Gienger, and J. Peters, "Robot learning from randomized simulations: A review," *Frontiers in Robotics and AI*, vol. 9, p. 799893, 2022.

[4] R. Calandra, A. Seyfarth, J. Peters, and M. P. Deisenroth, "Bayesian optimization for learning gaits under uncertainty," *Annals of Mathematics and Artificial Intelligence*, vol. 76, pp. 5–23, 2016.

[5] N. Hansen, "The cma evolution strategy: A tutorial," *arXiv preprint arXiv:1604.00772*, 2016.

[6] G. Comanici, E. Bieber, M. Schaekermann, I. Pasupat, N. Sachdeva, I. Dhillon, M. Blistein, O. Ram, D. Zhang, E. Rosen *et al.*, "Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities," *arXiv preprint arXiv:2507.06261*, 2025.

[7] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 5026–5033.

[8] A. Tekinalp, S. H. Kim, Y. Bhosale, T. Parthasarathy, N. Naughton, A. Albazroun, R. Joon, S. Cui, I. Nasiriziba, M. Stölzle, C.-H. C. Shih, and M. Gazzola, "Gazzolalab/pyelastica," 2024. [Online]. Available: https://doi.org/10.5281/zenodo.7658871

[9] N. Carion, L. Gustafson, Y.-T. Hu, S. Debnath, R. Hu, D. Suris, C. Ryali, K. V. Alwala, H. Khedr, A. Huang, J. Lei, T. Ma, B. Guo, A. Kalla, M. Marks, J. Greer, M. Wang, P. Sun, R. Rädle, T. Afouras, E. Mavroudi, K. Xu, T.-H. Wu, Y. Zhou, L. Momeni, R. Hazra, S. Ding, S. Vaze, F. Porcher, F. Li, S. Li, A. Kamath, H. K. Cheng, P. Dollár, N. Ravi, K. Saenko, P. Zhang, and C. Feichtenhofer, "Sam 3: Segment anything with concepts," 2025. [Online]. Available: https://arxiv.org/abs/2511.16719

[10] F. Ramos, R. C. Possas, and D. Fox, "Bayessim: adaptive domain randomization via probabilistic inference for robotics simulators," *arXiv preprint arXiv:1906.01728*, 2019.