Swim2Real: VLM-Guided System Identification for Sim-to-Real Transfer

Kevin Qiu[1,2], Kyle Walker[3], Mike Y. Michelis[4], Marek Cygan[1,5], Josie Hughes[3]

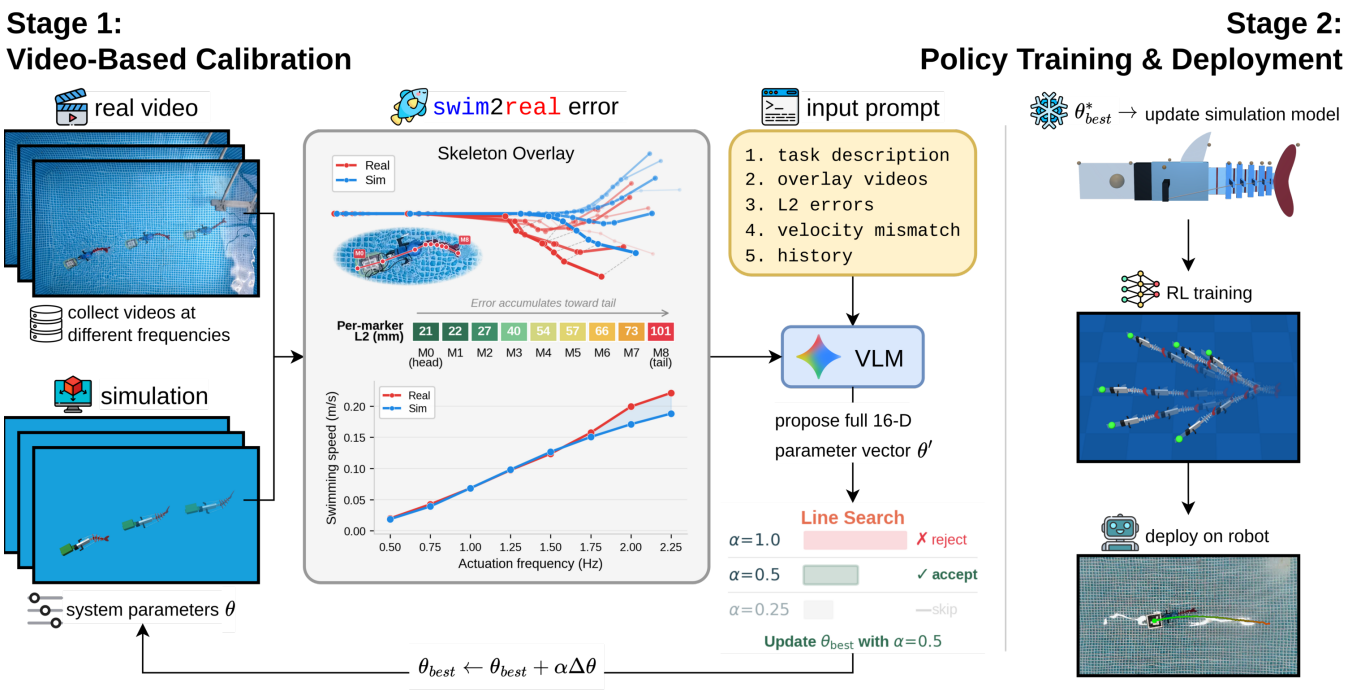

Fig. 1: Swim2Real calibrates a robotic fish simulator from video and deploys the resulting RL policy on hardware, with no hand-designed search stages. **Stage 1**: a VLM compares simulated and real swimming videos, proposes parameter adjustments, and a backtracking line search validates the step size, iterating for up to 40 evaluations. **Stage 2**: the calibrated simulator trains an RL policy that swims 12% farther than BayesOpt-calibrated policies. Motor commands from the trained policy are deployed on the physical fish at 50 Hz.

***Abstract*— We present Swim2Real, a pipeline that calibrates a 16-parameter robotic fish simulator from swimming videos using vision-language model (VLM) feedback, requiring no hand-designed search stages. Calibrating soft aquatic robots is particularly challenging because nonlinear fluid-structure coupling makes the parameter landscape chaotic, simplified fluid models introduce a persistent sim-to-real gap, and controlled aquatic experiments are difficult to reproduce. Prior work on this platform required three manually tailored stages to handle this complexity. The VLM compares simulated and real videos and proposes parameter updates. A backtracking line search then validates each step size, tripling the accept rate from 14% to 42% by recovering proposals where the direction is correct but the magnitude is too large. Swim2Real calibrates all 16 parameters simultaneously, most closely matching real fish velocities across all motor frequencies (MAE = 7.4 mm/s, 43% lower than the next-best method), with zero outlier seeds across five runs. Motor commands from the trained policy transfer to the physical fish at 50 Hz, completing the pipeline from swimming video to real-world deployment. Downstream RL policies swim 12% farther than those from BayesOpt-calibrated simulators and 90% farther than CMA-ES. These results demonstrate that VLM-guided calibration can close the sim-to-real gap for aquatic robots directly from video, enabling zero-shot RL transfer to physical swimmers without manual system identification, a step toward automated, general-purpose simulator tuning for underwater robotics.**

[1]University of Warsaw [2]IDEAS NCBR [3]EPFL [4]ETH Zurich [5]Nomagic
Correspondence: kevinxqiu@gmail.com

I. INTRODUCTION

Sim-to-real transfer requires simulators that accurately reproduce real-world dynamics [1]. Calibrating simulators for rigid-body systems benefits from analytical models and domain randomization [2], [3], but for soft and aquatic robots this remains challenging due to nonlinear material

responses, coupled parameter spaces and fluid-structure interactions that are difficult to tune manually or through standard optimization techniques [4], [5]. Ideally, calibration would be fully automatic, requiring only video of the real robot and producing calibrated simulator parameters with no manual intervention. This removes the need for the practitioner to decide which parameters to tune, which methods to use, or in what order. Any inaccurate calibration directly degrades downstream policy performance (Sec. V).

System identification fits simulator parameters to match observed behavior and is the classical approach to bridging this gap [6]. Recent methods frame system identification as a black-box process, applying Bayesian optimization [7] or evolutionary strategies [8] to search parameter spaces. These methods treat the simulator as an opaque function and rely solely on scalar error signals, ignoring the rich visual and physical structure of the comparison.

Recent work has demonstrated that video comparisons between simulated and real robots can drive system identification without hand-crafted cost functions, exploiting the visual structure that scalar metrics discard [9]. Vision-language models (VLMs) such as Gemini [10] provide a structured diagnostic signal. Rather than returning a scalar error, they reason about physical discrepancies from video, identifying when a simulated fish bends too sharply, moves too slowly, or exhibits incorrect tail dynamics. This VLM-based physical reasoning complements traditional optimizers, proposing *directions* informed by physical intuition, and a line search determines the correct *magnitude*.

To this end, we present Swim2Real, a VLM-guided system identification pipeline that takes raw swimming videos as input and automatically calibrates all 16 parameters of a tendon-driven fish simulator in MuJoCo [11]. Prior work on this platform [12] calibrated 9 parameters in three manually decomposed stages (stiffness via FFT analysis, motor geometry via grid search, and fluid coefficients via Bayesian optimization), requiring the practitioner to decide which parameters to group, which method to apply to each group, and in what order. Swim2Real calibrates all 16 parameters *simultaneously* (fluid coefficients, motor geometry, and per-joint stiffness and damping) with no hand-designed search stages, by iteratively comparing simulated and real swimming videos across eight actuation frequencies. A backtracking line search [13] validates each VLM proposal at geometrically decreasing step sizes, recovering updates where the VLM direction is correct but the magnitude is too large.

The contributions of this work are as follows:

1) Swim2Real, an end-to-end pipeline that calibrates all 16 simulator parameters directly from swimming videos, with no hand-designed search stages or per-parameter method selection.
2) Comprehensive sim-to-real validation: the calibrated simulator most closely matches real fish velocities across all motor frequencies (MAE = 7.4 mm/s, 43% lower than the next-best method), with zero outlier seeds across five runs.
3) Real-world deployment of RL policies trained in the calibrated simulator, which swim 12% farther than BayesOpt-calibrated policies, completing the pipeline from video input to physical robot execution.

II. RELATED WORK

Sim-to-Real Transfer. Domain randomization [2], [3] and its adaptive variants [14] train policies that are robust to simulator inaccuracies by sampling parameter distributions. While effective for rigid-body tasks [15], these methods do not reduce the sim-to-real gap itself but instead train around it. System identification offers a complementary approach by calibrating the simulator so that a single accurate model suffices for transfer [16].

System Identification for Robotics. Classical system identification fits parametric models to input-output data [6]. In the sim-to-real setting, BayesSim [7] estimates parameter posteriors via likelihood-free inference, while black-box methods apply Bayesian optimization [17] or CMA-ES [8] to minimize trajectory-matching errors [18]. These approaches rely on scalar cost functions and discard the visual structure of the comparison. Differentiable simulation [19] provides analytical gradients for system identification, including for soft robotic fish [20], [21], but the stateless ellipsoid fluid model used here lacks differentiable backend support. For soft robotic fish [5], [22], coupled fluid-structure dynamics make calibration especially difficult. Recent work on this platform [12] calibrated parameters by manually decomposing them into three stages, each with a tailored method (FFT matching, grid search, Bayesian optimization). Vid2Sid [9] demonstrated that iterative VLM feedback can replace hand-crafted metrics for system identification across multiple robotic platforms, but evaluated only up to 7 parameters with no downstream policy validation. Swim2Real scales VLM-guided calibration to 16 dimensions, introduces a backtracking line search that triples the accept rate to 42%, and validates the full pipeline through downstream RL training and real-world motor command transfer.

Foundation Models in Robotics. Large language models (LLMs) and VLMs have been applied to task planning [23], visuomotor control [24], and robot co-design [25], [26]. Eureka [27] uses LLMs to generate reward functions, and DrEureka [28] extends this to domain randomization distributions for sim-to-real transfer, automating the "train around the gap" approach rather than closing it. LLMs have also been used as iterative black-box optimizers [29] and to enhance Bayesian optimization surrogates [30], but operate on scalar or text feedback in low-dimensional spaces. In contrast, Swim2Real uses a VLM for *low-level parameter estimation*, observing simulation-reality discrepancies in video to propose quantitative parameter adjustments in a 16-dimensional continuous space.

III. THE SWIM2REAL PIPELINE

A. Problem Formulation

Consider a parameterized simulator $\mathcal{S}_\theta$ with parameters $\theta \in \mathbb{R}^d$ and a real robotic system producing reference tra-

jectories. The goal is to find θ^* that minimizes discrepancies between simulated and real marker trajectories:

$$\theta^* = \arg\min_{\theta\in\Theta} \frac{1}{|\mathcal{F}|} \sum_{f\in\mathcal{F}} \frac{1}{MT_f} \sum_{m=1}^{M} \sum_{t=1}^{T_f} \|\mathbf{p}_m^{\mathrm{s}}(t,f,\theta) - \mathbf{p}_m^{\mathrm{r}}(t,f)\|_2 \tag{1}$$

where $\Theta \subset \mathbb{R}^d$ is the bounded parameter space, $\mathcal{F}$ is a set of actuation frequencies, M is the number of body markers, T_f is the number of time steps at frequency f, and $\mathbf{p}_m^{\mathrm{s}}, \mathbf{p}_m^{\mathrm{r}} \in \mathbb{R}^2$ are simulated and real marker positions. We denote this objective $\mathcal{L}(\theta)$. Each evaluation of $\mathcal{L}(\theta)$ requires running $|\mathcal{F}|$ MuJoCo simulations and constitutes the dominant computational cost.

B. VLM-Guided Calibration

Rather than treating $\mathcal{L}$ as a black box, the proposed method uses a VLM (Gemini 2.5 Pro [10]) and provides it with two forms of information at each iteration (Fig. 1):

- **Video:** Side-by-side skeleton overlays of simulated (blue) and real (red) markers at each frequency, plus the real video at the worst-matching frequency.
- **Numerical data:** Per-marker, per-frequency L2 errors and velocities, the current parameter vector, parameter bounds, and the history of previous proposals and outcomes.

The prompt instructs the VLM to (1) identify physical discrepancies in the overlay videos (e.g., excess tail amplitude, insufficient thrust, incorrect body curvature), (2) reason about which simulator parameters are responsible, and (3) propose an updated parameter vector. The prompt also includes parameter semantics (e.g., "hingeStiffness_3 controls the restoring torque at joint 3") so the VLM can map visual observations to specific parameters. Unlike black-box optimizers that return only scalar improvements, this reasoning is interpretable and physically grounded (Sec. V).

C. Backtracking Line Search

VLM proposals often identify the correct update direction but overestimate the step size: the model can diagnose that, for example, the simulated tail bends too sharply, but quantifying the required stiffness reduction from video alone is imprecise. A backtracking line search addresses this by validating each proposal at decreasing step sizes, with the VLM providing the search direction in place of a gradient. Given the current best parameters θ_{best} and the VLM proposal θ', the search direction is $\Delta = \theta' - \theta_{\text{best}}$, and the method evaluates:

$$\theta_k = \theta_{\text{best}} + \beta^k \cdot \Delta, \quad k = 0, 1, \ldots, K-1 \tag{2}$$

where $\beta \in (0,1)$ is the decay factor and K is the maximum number of step sizes to try. The first k yielding $\mathcal{L}(\theta_k) < \mathcal{L}(\theta_{\text{best}})$ is accepted. If no step improves the objective, the round is rejected and the VLM receives this feedback in the next iteration. We use $\beta = 0.5$ and $K = 3$, so each round costs at most K simulation evaluations but triples the accept rate compared to evaluating only the full step (Sec. V). Algorithm 1 summarizes the complete procedure and Fig. 2 illustrates a single iteration. The budget B counts simulation evaluations only, as simulation is the computational bottleneck.

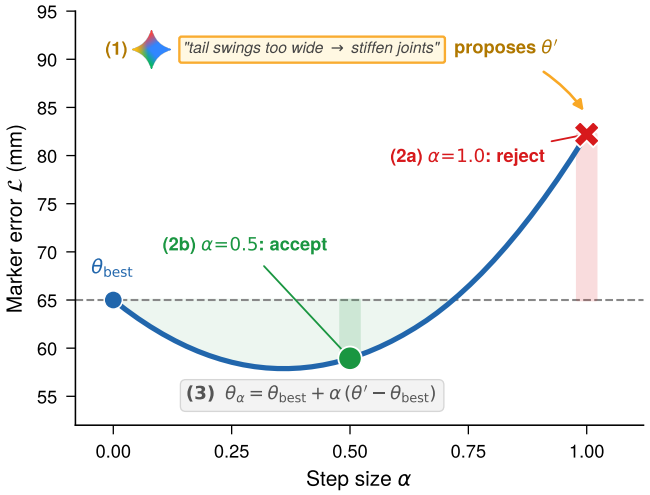

Fig. 2: One iteration of the backtracking line search. The VLM diagnoses a physical discrepancy and proposes updated parameters θ'. The full step ($\beta^0 = 1.0$) overshoots, but halving the step size ($\beta^1 = 0.5$) yields an improvement over $\mathcal{L}_{\text{best}}$ and is accepted. This triples the accept rate from 14% to 42% compared to evaluating only the full step.

TABLE I: The 16-dimensional parameter space covers fluid dynamics, motor geometry, and per-joint mechanical properties.

Parameter group	Count	Bounds	Unit
Fluid coefficients	5	$[0, 10]$	–
Motor arm length	1	$[0.01, 0.06]$	m
Hinge stiffness (per joint)	5	$[0.1, 5.0]$	N·m/rad
Hinge damping (per joint)	5	$[0.0, 2.0]$	N·m·s/rad

D. Parameter Space

The fish simulator [12] exposes $d = 16$ parameters spanning three physical subsystems (Table I), which Swim2Real searches jointly with no hand-designed search stages or domain-specific search strategy. The parameter bounds are inherited from the simulator design [12], and Swim2Real requires no additional domain knowledge beyond these bounds.

The five fluid coefficients parameterize blunt drag, slender drag, angular drag, Kutta lift, and Magnus lift in a stateless ellipsoid fluid model. Motor arm length controls the lever arm of the crank-slider tendon mechanism. Per-joint stiffness and damping govern the mechanical response of each of the five hinge joints in the discretized tail.

IV. EXPERIMENTAL SETUP

A. Hardware Platform

The experimental platform is a parametrically scalable tendon-driven robot fish [31], [12], 0.6 m in length with a total weight of 1.5 kg including electronics (Fig. 3). The

Algorithm 1 Swim2Real: VLM-Guided Calibration with Line Search

Require: Simulator $\mathcal{S}_\theta$, eval. budget B, param. bounds Θ, decay β, max steps K

$\theta_{\text{best}} \leftarrow \text{RandomInit}(\Theta)$; $\mathcal{L}_{\text{best}} \leftarrow \text{Evaluate}(\theta_{\text{best}})$; $B \leftarrow B-1$
$H \leftarrow \emptyset$ {Optimization history}
while $B > 0$ **do**
  $V \leftarrow \text{RenderOverlay}(\theta_{\text{best}})$ {Sim vs. real video}
  $\theta' \leftarrow \text{QueryVLM}(V,\ \theta_{\text{best}},\ H)$ {VLM proposes direction}
  $\Delta \leftarrow \theta' - \theta_{\text{best}}$; *accepted* $\leftarrow$ false
  {*Backtracking line search*}
  for $k = 0, 1, \ldots, K-1$ **do**
    $\theta_k \leftarrow \text{Clip}(\theta_{\text{best}} + \beta^k \Delta,\ \Theta)$
    $\mathcal{L} \leftarrow \text{Evaluate}(\theta_k)$; $B \leftarrow B-1$
    if $\mathcal{L} < \mathcal{L}_{\text{best}}$ **then**
      $\theta_{\text{best}} \leftarrow \theta_k$; $\mathcal{L}_{\text{best}} \leftarrow \mathcal{L}$; *accepted* $\leftarrow$ true; **break**
    end if
  end for
  Append $(\Delta, \beta^k, \textit{accepted})$ to H
end while
return $\theta_{\text{best}},\ \mathcal{L}_{\text{best}}$

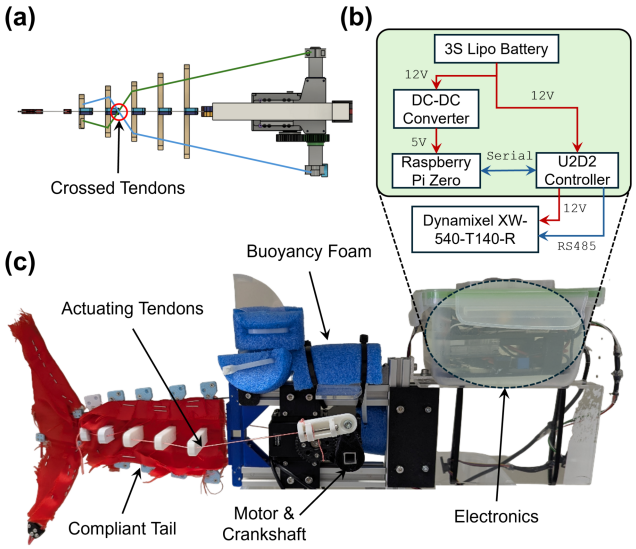

Fig. 3: (a) CAD cross-section showing the antagonistic tendon arrangement that crosses at the tail midpoint to produce an S-bend. (b) Block diagram of the onboard electronics. (c) The tendon-driven fish robot platform with annotated components. A single motor drives the full range of swimming gaits used for calibration and RL deployment.

design consists of a passive front-end and an active compliant tail, with pectoral and dorsal fins for stability. The passive front-end is constructed from acrylic sheet and 3D-printed mounts fixed to an aluminum frame. A Raspberry Pi Zero 2 W serves as the onboard processor, driving a waterproof Dynamixel XW-540-T140-R motor via a U2D2 controller, all powered by a 3S LiPo battery with a DC-DC converter (Fig. 3(b)).

The compliant tail uses super-elastic nitinol rods held by five 3D-printed spine segments, with antagonistic tendons that cross at the midpoint to produce a bio-inspired S-bend (Fig. 3(a)) driven by a dual-output crank-slider mechanism.

B. Data Collection and Simulation

Eleven markers on the body are tracked using a CSRT tracker from overhead camera footage (2160×3840, 60 fps) in a 2 m × 3 m pool. We use $M{=}9$ body markers (M0 at the head, M8 at the tail tip) for error computation, with trajectories rotated into the fish's local frame to remove global position and orientation [12]. The fish is actuated at eight frequencies ($\mathcal{F} = \{0.50, 0.75, \ldots, 2.25\}$ Hz) covering its operational range.

The simulator uses MuJoCo [11] with an implicit integrator at 1 kHz, discretizing the continuously bending tail as five rigid segments connected by hinge joints. Fluid forces use a stateless ellipsoid model [12] parameterized by five coefficients (Table I) that capture the dominant hydrodynamic forces without expensive CFD computation. This model was shown to outperform classical Elongated Body Theory [12]. Each evaluation runs all eight frequencies and computes the mean Euclidean distance between simulated and real marker positions across $M{=}9$ markers and $|\mathcal{F}|{=}8$ frequencies (Eq. 1). All reported marker errors are in-data: calibration and evaluation use the same eight frequencies. Forward swimming velocity (Sec. V) serves as an out-of-distribution metric, as it is not part of the calibration objective. A single 8-frequency simulation evaluation takes approximately 13 s on a single core of an AMD Ryzen 7 PRO 7840U, and each VLM call (video rendering, upload, inference, parsing) requires approximately 42 s. Mean wall-clock runtimes across 5 seeds are 19 min for Swim2Real (8 min simulation + 11 min VLM), compared to 6 min for BayesOpt, CMA-ES, and random search. The VLM overhead roughly triples wall-clock time, but remains minor relative to real-world video collection, which requires physical pool access, marker tracking, and multiple frequency sweeps.

C. Baselines

We compare Swim2Real (Sec. III) against three established black-box optimization approaches:

- *Bayesian optimization* (BayesOpt) uses a GP surrogate with Matérn 5/2 kernel and `gp_hedge` acquisition (portfolio of EI, LCB, PI) via scikit-optimize [17].
- *CMA-ES* [8] operates in the full 16D space with population size 12 ($4{+}\lfloor 3 \ln d \rfloor$) and initial step size $\sigma_0{=}0.2$ in normalized space, yielding $\sim$3 generations within the 40-eval budget.
- *Random search* samples uniformly within parameter bounds (performance reference).

Ablations isolate design choices in Swim2Real:

- *w/o line search.* Same VLM loop but evaluates only the full proposal ($\beta^0 = 1.0$), matching the iterative

protocol of Vid2Sid [9]. Each VLM call costs exactly 1 simulation evaluation.

- *w/ Gemini 3.1.* Replaces Gemini 2.5 Pro with Gemini 3.1 Pro Preview in the no-line-search configuration.
- *Warm start.* One VLM proposal followed by random search for the remaining budget. Tests whether iterative feedback is necessary.

D. Protocol

All experiments use a budget of $B = 40$ simulation evaluations across 5 random seeds (seeds 0–4). All methods except CMA-ES share identical random initializations per seed, ensuring a fair comparison. CMA-ES uses its own population-based initialization ($\leq$5 mm difference in initial error). We report all individual seed results transparently via strip plots alongside aggregate statistics.

E. Downstream RL Evaluation

To test whether calibration accuracy translates to usable policies, we train RL agents in simulators instantiated with each method's best-found parameters, following recent work showing that simulation fidelity is a prerequisite for aquatic policy transfer [32]. We use SAC [33] with identical hyperparameters across all conditions (learning rate 2×10^{-3}, batch size 256, γ=0.99, τ=0.1, 50K exploration steps), varying only the simulator parameters. Both tasks use a single continuous action $a \in \mathbb{R}$ controlling motor acceleration, and a $[256, 256]$ MLP trained for 5M steps (3 seeds per condition).

Forward swimming. The agent observes 11 marker positions and velocities, head pose, and motor state (52-dim). The reward is forward displacement along the swimming axis, $r = -x_{\text{head}}$ (the x-axis points backward, so negating the head position rewards forward progress), evaluated over 2 s episodes following [12].

Target reaching. The agent observes motor state, tail joint angles and velocities, and the goal vector in the body frame (22-dim). Training uses a three-stage curriculum that gradually expands the target region from 1×1 m to 3×4 m and tightens the success threshold from 10 cm to 5 cm. The reward combines goal distance, action cost, and a success bonus:

$$r = -d(\mathbf{p}_{\text{fish}}, \mathbf{p}_{\text{goal}}) - 0.5\|a\| + 300 \cdot \mathbf{1}[d < 0.05\,\text{m}] \quad (3)$$

This two-task evaluation tests locomotion quality and directional control, both required for real-world deployment.

V. RESULTS

A. Calibration Accuracy

Table II summarizes performance across all methods. Swim2Real achieves 51.3 ± 1.2 mm (mean $\pm$ std, N=5) with the tightest variance of any method. All five seeds fall within 50.2–53.2 mm, with zero outliers. BayesOpt reaches comparable final error (52.4 ± 2.1 mm), but CMA-ES exhibits catastrophic failures on 2 of 5 seeds (123.4 and 254.2 mm). With population size 12 and budget 40, CMA-ES completes only $\sim$3 generations, too few for reliable convergence in 16 dimensions [8]. Swim2Real achieves this reliability with no hand-designed search stages, operating directly from video comparisons. Fig. 4 shows the qualitative difference: Swim2Real reproduces the body shape and forward progression of the real fish, while CMA-ES exhibits incorrect posture. With N=5 seeds, formal significance tests have low power, so we report all individual seed results and focus on effect sizes. The 1.1 mm gap from BayesOpt is within seed variance. The meaningful advantages are reliability (all seeds within a 3 mm range) and downstream RL transfer performance.

TABLE II: Calibration performance across 5 seeds. Best: mean $\pm$ std of per-seed final L2 marker error (averaged across all markers). Worst: highest per-seed final error. AUC: mean best-so-far error over 40 evaluations (omitted for ablations, whose different per-round costs make the metric non-comparable). Swim2Real achieves the lowest error and tightest variance, with all five seeds falling within a 3 mm range.

Method	Best (mm)↓	Worst (mm)↓	AUC (mm)↓
Random	82.0 ± 34.3	141.8	129.3 ± 26.6
CMA-ES	112.7 ± 83.9	254.2	156.4 ± 91.7
BayesOpt	52.4 ± 2.1	55.6	94.2 ± 17.6
Swim2Real (Ours)	$\mathbf{51.3 \pm 1.2}$	**53.2**	$\mathbf{85.9 \pm 30.9}$
w/o line search	52.4 ± 2.8	57.3	–
w/ Gemini 3.1	54.6 ± 4.2	62.0	–
Warm start	91.1 ± 49.5	177.4	–

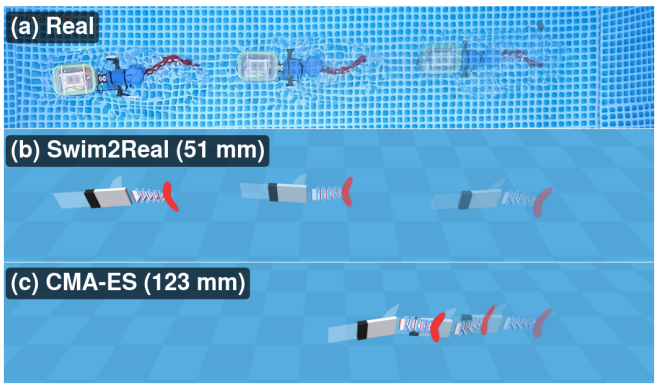

Fig. 4: Chronophotography of swimming at 1.5 Hz (3 snapshots, increasing opacity). *Top:* real fish (overhead). *Middle:* Swim2Real-calibrated simulator (51 mm error). *Bottom:* CMA-ES calibration (123 mm error). Swim2Real reproduces the body shape and forward progression of the real fish, while CMA-ES exhibits incorrect posture and reduced thrust.

B. Sample Efficiency

Fig. 5a shows best-so-far error trajectories across all methods. To compare sample efficiency, we report the area under the convergence curve (AUC), which integrates both convergence speed and final accuracy into a single metric (Table II). Swim2Real achieves the lowest AUC (85.9 mm), ahead of BayesOpt (94.2 mm), random search (129.3 mm), and CMA-

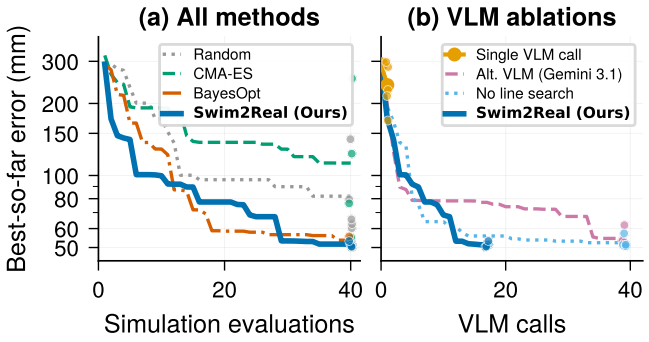

Fig. 5: Best-so-far L2 error (mean across 5 seeds, with dots showing individual seed final values). (a) All five Swim2Real seeds fall within 50.2–53.2 mm, while CMA-ES collapses on 2 of 5. (b) The line search triples the accept rate (42% vs. 14%), so Swim2Real reaches ∼51 mm in ∼16 VLM calls while the no-line-search ablation requires 39.

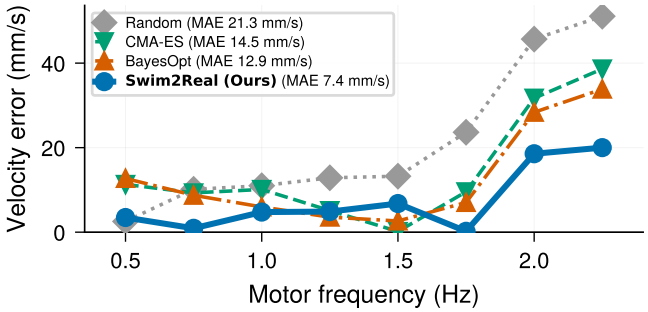

Fig. 6: Velocity error $|v_{\text{sim}} - v_{\text{real}}|$ across motor frequencies. Swim2Real tracks the real fish most closely (MAE = 7.4 mm/s, half that of BayesOpt at 12.9 mm/s), with the gap widening at higher frequencies where calibration quality matters most.

ES (156.4 mm). Swim2Real's higher AUC variance (±30.9 vs. ±17.6 for BayesOpt) reflects seed-dependent VLM convergence speed rather than final accuracy, since all five seeds reach ≤ 53.2 mm. BayesOpt's GP surrogate converges rapidly in mid-optimization, while Swim2Real's line search enables steady refinement across all phases.

Fig. 5b shows the VLM call efficiency. The line search makes each VLM call 2.5× more productive. Swim2Real uses a mean of 15.6 VLM calls per run (42% accepted), compared to 39 for the no-line-search ablation (14% accepted), while both reach ∼51 mm. Each VLM call costs 2.5 simulation evaluations (line search overhead), but this investment triples the accept rate and halves the total VLM calls needed.

C. Ablation Studies

Line search. Across 78 VLM rounds (5 seeds), the line search accepts 33 of 78 proposals. Among accepted proposals, 39.4% use the full step ($\beta^0 = 1.0$), 33.3% use a half step ($\beta^1 = 0.5$), and 27.3% require a quarter step ($\beta^2 = 0.25$), meaning 61% of successful updates would have been rejected without the line search. Each VLM call costs a mean of 2.5 simulation evaluations (line search overhead), but recovers 3× more proposals, reducing the cost per accepted update from 7.2 to 5.9 evaluations.

Alternative VLM. Replacing Gemini 2.5 Pro with Gemini 3.1 Pro in the no-line-search configuration yields 54.6 ± 4.2 mm vs. 52.4 ± 2.8 mm, a 2.2 mm difference within seed variance, suggesting that the approach is robust to model version within the Gemini family.

Single VLM call. The warmstart ablation (one VLM proposal followed by random search) produces 91.1 ± 49.5 mm with a worst seed at 177.4 mm, confirming that iterative VLM feedback is essential.

Interpretable reasoning. To illustrate the VLM's diagnostic capability, the following is a verbatim diagnosis from the first accepted round (seed 0):

> *"The simulation exhibits a critical failure at high frequencies (≥1.25 Hz), where the tail motion is severely attenuated, leading to a near-complete loss of thrust [...]. This 'locking up' behavior points to excessive damping forces that become dominant at high angular velocities. Conversely, at low frequencies, the simulated tail appears too flexible, over-swinging compared to the real fish."*

Based on this diagnosis, the VLM decreased fluid drag and increased hinge stiffness. The proposal was accepted at β^0=1.0, reducing error from 168.8 to 56.5 mm. Scalar objective functions cannot provide this kind of physically grounded correction.

D. Forward Swimming Velocity

L2 marker error captures body-shape fidelity, but forward velocity is not part of the calibration objective (Eq. 1), making it an independent test of simulator quality. Forward velocity is computed as the slope of a linear fit to the head marker position along the swimming axis over the full trial duration, applied identically to simulated and real trajectories. We compare sim-predicted forward swimming speed against the real fish at eight motor frequencies spanning the full calibration range (0.5–2.25 Hz). Fig. 6 shows that Swim2Real-calibrated parameters yield the closest velocity match (MAE = 7.4 mm/s), half that of BayesOpt (12.9 mm/s), followed by CMA-ES (14.5 mm/s) and random (21.3 mm/s). This ranking is consistent with calibration accuracy, and separation increases at higher frequencies where calibration quality matters most.

E. Downstream RL Transfer

We evaluate downstream task performance using the RL setup described in Sec. IV.

Forward swimming. We train policies for 5M steps (3 seeds each) using the observation and reward structure of [12]. Calibration quality directly determines policy quality. Swim2Real-calibrated policies swim 7.6 ± 0.0 m (3 seeds), compared to 6.8 ± 0.1 m for BayesOpt, 6.0 ± 0.0 m for random calibration, and 4.0 ± 0.6 m for CMA-ES. The ranking is monotonic with calibration accuracy, confirming that the L2 calibration metric captures real differences

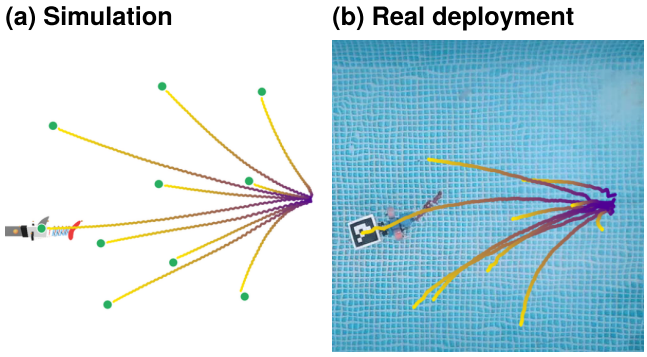

Fig. 7: Sim-to-real transfer of the Swim2Real-calibrated RL policy. (a) Simulated target-reaching trajectories to diverse goals (green dots). (b) Real-world deployment with motor commands transferred open-loop at 50 Hz. Color encodes time (purple = start, yellow = end). The leftward arc in (b) results from open-loop execution and a hardware steering bias (see text). Despite this, the fish produces directed swimming from commands generated entirely in simulation.

in simulator fidelity. The out-of-objective velocity match (Fig. 6) explains why Swim2Real outperforms BayesOpt despite similar marker error. Swim2Real captures thrust-producing dynamics more faithfully across the full frequency range, a difference invisible to the aggregate marker metric but critical for locomotion.

Real-world deployment. To validate the full sim-to-real pipeline, we deploy a target-reaching policy trained in the Swim2Real-calibrated simulator onto the physical fish. Motor commands generated in simulation are transferred open-loop at 50 Hz, with an overhead camera tracking the resulting trajectory. In simulation (Fig. 7a), the policy navigates to diverse goal positions. On hardware (Fig. 7b), the fish produces directed swimming, though real trajectories exhibit a consistent leftward arc absent in simulation. We hypothesize that this discrepancy arises from two sources. First, the deployment is open-loop without state feedback, so the policy cannot correct for drift. Second, the robot exhibits a left steering bias, suspected to arise from friction between the tendons and the routing holes. The calibration metric operates in the fish's local body frame (removing global position and heading), so the steering bias is invisible to the calibration process and cannot be corrected by any marker-based calibration method. Despite these limitations, the deployed policy drives the fish towards the targets, confirming that the calibrated simulator captures sufficient body-dynamic fidelity for motor-command transfer.

VI. DISCUSSION

A. The 50 mm Error Floor

Per-marker analysis reveals a head-to-tail error gradient. Head markers (M0–M2) average 23 mm while tail markers (M6–M8) average 80 mm, confirming that the 50 mm floor is dominated by the discretized tail's kinematic chain. Five rigid hinge joints approximate a continuously bending tail, and the approximation error accumulates along the chain. The tail tip (M8) alone reaches 101 mm. Reducing this floor would require per-segment fluid coefficients or a higher-fidelity fluid model.

B. Design Rationale

The algorithmic simplicity of the approach is deliberate. The VLM already provides structured reasoning about physical discrepancies, and the line search corrects the one systematic error (magnitude overestimation) that the VLM consistently makes. This parallels how gradient-based optimization pairs a descent direction with a line search to determine the step size. In our approach, the VLM replaces the gradient with visual-physical reasoning.

The 58% rejection rate (even with line search) reflects cases where the VLM adjusts too many parameters simultaneously, creating coupling effects that increase overall error. The line search recovers directionally correct but oversized proposals, yet cannot compensate for fundamentally wrong directions. Analysis of rejected rounds reveals two dominant failure modes. First, the VLM often conflates fluid coefficient effects with joint stiffness effects. Both influence tail amplitude but through different physical mechanisms (external hydrodynamic forcing vs. internal restoring torque), and adjusting both simultaneously can cancel the intended correction. Second, motor arm length has no direct visual signature in the skeleton overlay, so VLM adjustments to this parameter are essentially guesses informed by overall thrust mismatch rather than a specific visual cue. Feeding rejection feedback to the VLM enables course correction in subsequent rounds. The iterative loop is essential, as the warmstart ablation (single VLM call, 91.1 ± 49.5 mm) confirms.

VII. CONCLUSIONS

We presented Swim2Real, an end-to-end pipeline that replaces manually tailored, multi-stage calibration with VLM-guided system identification, calibrating all 16 parameters of a tendon-driven fish simulator simultaneously from video input alone. A backtracking line search triples the VLM accept rate, with all five seeds falling within a 3 mm range. The calibrated simulator produces the closest match to real fish dynamics across all evaluation axes, including marker error, swimming velocity, and downstream RL performance, with motor commands transferring successfully to the physical fish. These results suggest that VLMs can serve as effective physics reasoners for system identification, matching Bayesian optimization in accuracy while providing interpretable diagnostic feedback that scalar optimizers cannot. The pipeline currently relies on a proprietary VLM (Gemini), though the Gemini 3.1 ablation demonstrates robustness across model versions and the prompt requires only standard vision-language capabilities, suggesting portability to open-source VLMs. VLM API costs ($\sim$15 calls per run) triple wall-clock time relative to simulation-only baselines (19 vs. 6 min), though both are minor compared to real-world data collection.

Future work includes closed-loop policy execution with onboard state estimation to eliminate drift and steering bias, extending to higher-dimensional parameter spaces where GP surrogates scale poorly, and applying the pipeline to other robotic platforms where simulation-reality discrepancies are visually observable.

REFERENCES

[1] 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.

[2] 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 *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2017, pp. 23–30.

[3] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2018, pp. 3803–3810.

[4] C. Della Santina, C. Duriez, and D. Rus, "Model-based control of soft robots: A survey of the state of the art and open challenges," *IEEE Control Systems Magazine*, vol. 43, no. 3, pp. 30–65, 2023.

[5] A. D. Marchese, C. D. Onal, and D. Rus, "Autonomous soft robotic fish capable of escape maneuvers using fluidic elastomer actuators," *Soft Robotics*, vol. 1, no. 1, pp. 75–87, 2014.

[6] L. Ljung, *System Identification: Theory for the User*, 2nd ed. Prentice Hall, 1999.

[7] F. Ramos, R. Possas, and D. Fox, "BayesSim: Adaptive domain randomization via probabilistic inference for robotics simulators," in *Proc. Robotics: Science and Systems (RSS)*, 2019.

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

[9] K. Qiu, Y. Zhang, M. Cygan, and J. Hughes, "Vid2sid: Videos can help close the sim2real gap," *arXiv preprint arXiv:2602.19359*, 2026.

[10] Gemini Team, R. Anil, S. Borgeaud, *et al.*, "Gemini: A family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2023.

[11] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2012, pp. 5026–5033.

[12] M. Y. Michelis, N. Obayashi, J. Hughes, and R. K. Katzschmann, "Simple models, real swimming: Digital twins for tendon-driven underwater robots," *arXiv preprint arXiv:2602.23283*, 2026.

[13] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. Springer, 2006.

[14] Y. Chebotar, A. Handa, V. Makoviychuk, M. Macklin, J. Issac, N. Ratliff, and D. Fox, "Closing the sim-to-real loop: Adapting simulation randomization with real world experience," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2019, pp. 8973–8979.

[15] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, p. eaau5872, 2019.

[16] W. Yu, J. Tan, C. K. Liu, and G. Turk, "Preparing for the unknown: Learning a universal policy with online system identification," in *Proc. Robotics: Science and Systems (RSS)*, 2017.

[17] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 25, 2012.

[18] Y. Du, O. Watkins, T. Darrell, P. Abbeel, and D. Pathak, "Auto-tuned sim-to-real transfer," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2021, pp. 1290–1296.

[19] K. M. Jatavallabhula, M. Macklin, F. Golemo, V. Voleti, L. Petrini, M. Weiss, B. Considine, J. Parent-Lévesque, K. Xie, K. Erleben, *et al.*, "gradSim: Differentiable simulation for system identification and visuomotor control," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2021.

[20] J. Z. Zhang, Y. Zhang, P. Ma, E. Nava, T. Du, P. Arm, W. Matusik, and R. K. Katzschmann, "Sim2real for soft robotic fish via differentiable simulation," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 12 598–12 605.

[21] J. Gao, M. Y. Michelis, A. Spielberg, and R. K. Katzschmann, "Sim-to-real of soft robots with learned residual physics," *IEEE Robotics and Automation Letters*, vol. 9, no. 10, pp. 8523–8530, 2024.

[22] R. K. Katzschmann, J. DelPreto, R. MacCurdy, and D. Rus, "Exploration of underwater life with an acoustically controlled soft robotic fish," *Science Robotics*, vol. 3, no. 16, p. eaar3449, 2018.

[23] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gober, K. Gopalakrishnan, *et al.*, "Do as I can, not as I say: Grounding language in robotic affordances," in *Proc. Conf. Robot Learning (CoRL)*, 2022.

[24] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, *et al.*, "RT-2: Vision-language-action models transfer web knowledge to robotic control," in *Proc. Conf. Robot Learning (CoRL)*, 2023.

[25] K. Qiu, W. Pałucki, K. Ciebiera, P. Fijałkowski, M. Cygan, and Ł. Kuciński, "Robomorph: Evolving robot morphology using large language models," *arXiv preprint arXiv:2407.08626*, 2024.

[26] K. Qiu and M. Cygan, "Debate2create: Robot co-design via large language model debates," *arXiv preprint arXiv:2510.25850*, 2025.

[27] Y. J. Ma, W. Liang, G. Wang, D.-A. Huang, O. Bastani, D. Jayaraman, Y. Zhu, L. Fan, and A. Anandkumar, "Eureka: Human-level reward design via coding large language models," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2024.

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

[29] C. Yang, X. Wang, Y. Lu, H. Liu, Q. V. Le, D. Zhou, and X. Chen, "Large language models as optimizers," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2024.

[30] T. Liu, N. Astorga, N. Seedat, and M. van der Schaar, "Large language models to enhance Bayesian optimization," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2024.

[31] N. Obayashi, A. Anastasiadis, J. Gumowski, K. Junge, K. L. Walker, K. Mülleners, and J. Hughes, "ScaFi: Length-scalable, compliant, parametric robotic fish design for operation in multiple environmental niches," 2025.

[32] X. Lin, X. Liu, and Y. Wang, "Learning agile swimming: An end-to-end approach without CPGs," *IEEE Robotics and Automation Letters*, vol. 10, pp. 1992–1999, 2025.

[33] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. Int. Conf. Machine Learning (ICML)*, 2018, pp. 1861–1870.