ENERGY-EFFICIENT MOTION CONTROL IN FLUID–STRUCTURE INTERACTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Energy-efficient motion through a fluid requires coordinating body movement with fluid forces and torques. This task is challenging because body motion changes the surrounding flow, which in turn affects how the body moves. Here, we study body motion control in flowing channels using a training dataset generated with computational fluid dynamics (CFD). The dataset covers two body shapes, three channel geometries, and both sensor-free and sensor-based settings. We consider direct trajectory control and force-and-torque control within a common framework that enables comparisons of physical predictors and control generators across different settings. For sensor-free circular-body control, experiments identify paths using up to 25.7% less work than the corresponding best training trajectories and demonstrate feasible planning with unseen endpoints. For sensor-based square-body control, the best neural controllers use 36.6% and 64.4% less work than the corresponding best training trajectories in the unobstructed and central-obstacle channels, respectively, and 32.3% less in the more complex two-wall channel. We hope this work will help extend the use of physical operators beyond forward prediction to more challenging control tasks.

1 INTRODUCTION

From eagles flying through air to fish swimming through water, motion in nature often depends on interactions between bodies and fluids. Fish move their bodies and fins to push water, generating forces and torques that propel and turn them. Beyond nature, fluid control is essential in transportation, energy, manufacturing and medicine. For a body moving through a fluid, the energy required depends on how its motion and fluid forces are coordinated. This raises a key question: how can we control such motion to complete a task using as little energy as possible?

Answering this question is computationally demanding even when the governing equations are known. Although a rigid body has only a few degrees of freedom, predicting its motion requires solving the unsteady Navier–Stokes equations with moving boundaries alongside the rigid-body dynamics. Body motion changes the flow, while fluid forces and torques in turn affect the body. Achieving energy-efficient motion requires not only predicting this coupled response but also choosing the applied forces and torques over time. This naturally leads to an optimal control problem that minimises energy use while avoiding obstacles and meeting the required final conditions. Searching for suitable controls requires repeated simulations, further increasing the computational cost.

Learning-based PDE models, such as neural operators (Lu et al., 2021; Li et al., 2021), can reduce prediction costs by learning dynamics from data. Shared datasets support training and evaluation. PDEBench (Takamoto et al., 2022) covers diverse equations and physical regimes, while DrivAerNet++ (Elrefaie et al., 2024) focuses on aerodynamic prediction across different geometries. RealPDEBench (Hu et al., 2026) pairs real-world measurements with simulations to assess prediction and simulation-to-reality transfer. These resources extend forward modelling to larger fields, more complex boundaries and real-world conditions. However, they primarily assess prediction rather than control. In our setting, prediction accuracy alone is insufficient. We must also evaluate whether model-based controls achieve energy-efficient motion while satisfying task constraints.

These developments complement a growing literature on learning-based physical control. Differentiable physics and reinforcement learning have demonstrated learned control of PDE dynamics.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

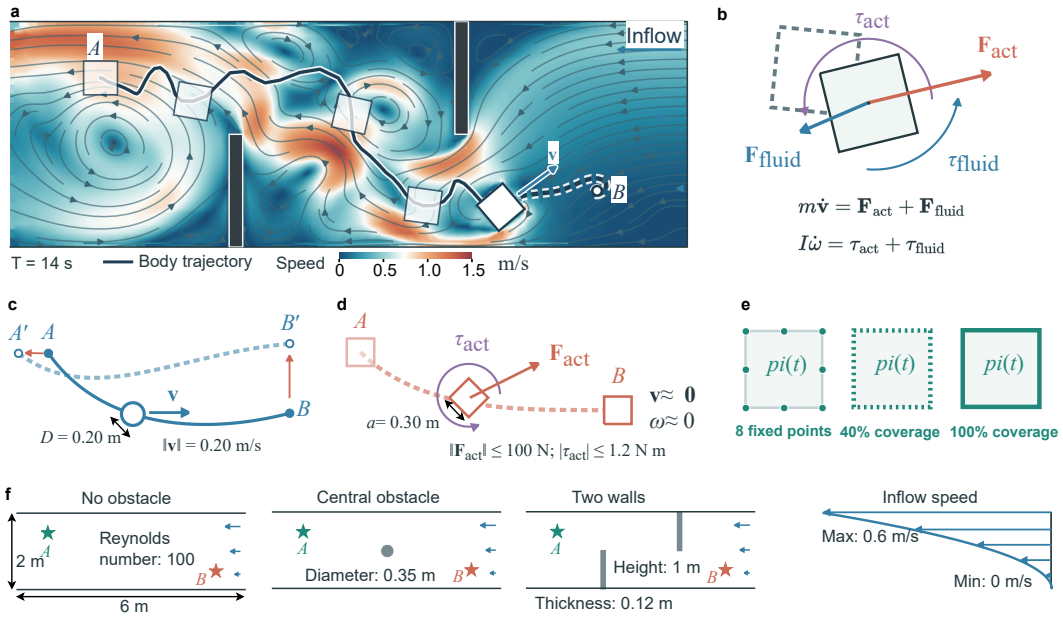

Figure 1: **Coupled motion as a test of prediction and control.** **a**, A square moves from A to B through two walls; the background shows CFD speed at 14 s, together with streamlines and the trajectory. **b**, Applied and fluid forces and torques jointly determine translation and rotation. **c**, Circular-body path control with fixed and shifted endpoints. **d**, Square-body force and torque control, ending near rest. **e**, Pressure-sensor layouts on the body boundary: eight fixed sensors, 40% perimeter coverage and full perimeter coverage. **f**, Body motion from start A to target B in channels with no obstacle, a central circular obstacle or two walls; inflow is from right to left.

Bieker et al. (2019) combine recurrent predictions from sensor histories with model predictive control and online model updates, while HydroGym (Lagemann et al., 2026) provides shared fluid-control environments for developing and comparing policies. However, these settings mainly focus on regulating flows or fluid forces, with less emphasis on control of moving body. For motion control, conventional methods using simplified load models may not fully capture how body motion reshapes the surrounding flow. This matters particularly in confined channels, where the altered flow changes the forces and torques on the body, affecting its motion and energy use.

In this work, we study energy-efficient motion control in flowing channels (Fig. 1). We introduce a dataset covering **two body shapes** (circular and square), **three channel geometries** (no obstacle, a central obstacle and two walls), and **two control settings** (sensor-free and sensor-based control). For motion control of the body, we use direct **trajectory control** for the circular body and **force-and-torque control** for the square body. For model-based planning, physical predictors estimate the response to candidate paths or control sequences, while control generators propose and refine these candidates. A common framework supports comparisons of these components across the different settings. Selected paths and controls are executed in computational fluid dynamics (CFD) simulations to assess energy use, obstacle clearance and task-specific terminal conditions.

In our experiments with this framework, for sensor-free circular-body control, learned models identify paths that use up to 25.7% less energy than the corresponding best training trajectory. All evaluated paths remain feasible with both fixed and shifted endpoints. For sensor-based square-body control, the best neural controllers use 36.6%, 64.4% and 32.3% less energy than the corresponding best training trajectories in the unobstructed, central-obstacle and two-wall channels, respectively. These results demonstrate the potential of learned physical operators to extend from forward prediction to the inverse task of finding energy-efficient controls that satisfy motion constraints.

2 Method

We compare methods for energy-efficient motion, distinguishing the planning framework from its predictor and control-generator components (Fig. 2a–e). We first define the tasks, objectives, con-

straints and observations, then describe the planning and control frameworks. Finally, we introduce the predictor and control-generator candidates evaluated within these frameworks.

2.1 PROBLEM SETTING

We seek controls that move a body from start A to target B within a prescribed time limit while using little positive work. The body and surrounding flow evolve together (Fig. 1a). For the circular body, we choose the path. For the square, we choose the forces and torque. Each task uses one of three channel geometries $\mathcal{G}$ with no obstacle, a central circular obstacle or two walls (Fig. 1f). Time t measures elapsed time during the motion. We write $[\xi]_+ = \max(\xi, 0)$ for the positive part.

Circular-body trajectory control. We optimize a path $\mathcal{P}$ that the body follows at constant speed (Fig. 1c). Let $T_{\mathcal{P}}$ denote the travel time, $\mathbf{v}(t)$ the body velocity and $\mathbf{F}_{\text{fluid}}(t)$ the fluid force. The positive work is

$$E_{\text{circle}} = \int_0^{T_{\mathcal{P}}} [-\mathbf{F}_{\text{fluid}} \cdot \mathbf{v}]_+ \, \mathrm{d}t. \tag{1}$$

The path must connect the endpoints, satisfy the length limit and avoid collisions with obstacles and channel walls. We first optimize paths between fixed endpoints A and B, then repeat the search between shifted endpoints A' and B' using the same trained predictor.

Square-body force-and-torque control. We choose the applied force $\mathbf{F}_{\text{act}}$ and torque τ_{act} to move the square to target B and bring it to rest (Fig. 1d). Applied loads combine with the fluid force $\mathbf{F}_{\text{fluid}}$ and torque τ_{fluid} to determine translation and rotation (Fig. 1b). At control step n, the body state $s_n = (\mathbf{x}_b, \mathbf{v}, \phi, \omega)_n$ contains position, velocity, orientation and angular velocity, and the action $u_n = (F_{\text{act},x}, F_{\text{act},y}, \tau_{\text{act}})_n$ is held constant over an interval of $\Delta t = 0.25\,\text{s}$. The action sequence $U = (u_0, \ldots, u_{N-1})$ covers the allowed motion time $T = N\Delta t$. Its positive actuator work is

$$E_{\text{square}} = \int_0^{T} [\mathbf{F}_{\text{act}} \cdot \mathbf{v} + \tau_{\text{act}}\omega]_+ \, \mathrm{d}t. \tag{2}$$

The motion must remain collision-free, satisfy the force and torque limits, and meet the terminal tolerances for position, orientation, linear velocity and angular velocity.

We consider both sensor-based and sensor-free control. For sensor-based control, we consider different sensor-coverage settings (Fig. 1e) and each action is selected using an eight-frame history h_n of pressures on the square's surface, body states and previous actions. In the sensor-free setting, no observations are received during motion. Details on data generation, physical parameters and terminal tolerances are provided in Appendix B.

2.2 PLANNING FRAMEWORK

We compare methods for choosing energy-efficient paths and force-and-torque sequences. A **predictor** F_θ estimates the response to a candidate plan, including flow fields, fluid loads, body motion or work. A **control generator** chooses paths or actions, either by comparing predicted outcomes or directly from observations. Predictor parameters θ remain fixed during planning.

Sensor-free control. We consider two sensor-free tasks: circular-body trajectory control and square-body force-and-torque control (Fig. 2a). For the circle, the control generator adjusts **trajectory sequence** $\mathcal{P}$, while the predictor estimates the resulting work. For the square, the control generator adjusts the **force-and-torque sequence** U within the action limits, while the predictor estimates the resulting motion and work. Candidates are ranked using predicted work and task constraints. For each combination of predictor, generator and geometry, three plans are selected for CFD evaluation.

Sensor-based control. We consider square-body force-and-torque control with sensor observations (Fig. 2b). To stabilize the motion, we combine a base feedback action with a correction. The history h_n contains recent pressure measurements, body states and previous actions. The base action $u_{\text{FB}}(h_n)$ is computed using disturbance-observer control (DOB) (Chen et al., 2016) to guide the square along waypoints, regulate its orientation and compensate for estimated fluid forces and torques. The control generator provides the correction r_n. The three frameworks below specify how this correction is obtained. We enforce the force and torque limits using clip, giving the action

$$u_n = \text{clip}(u_{\text{FB}}(h_n) + r_n)\,. \tag{3}$$

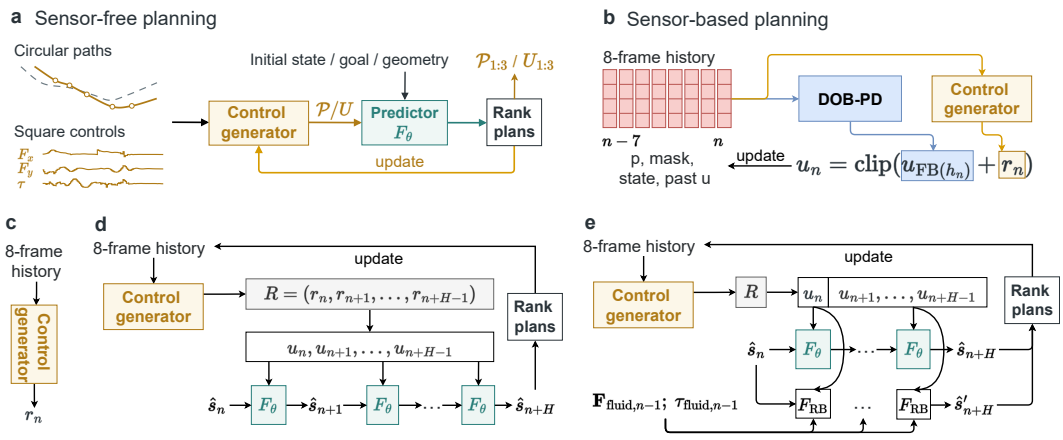

Figure 2: **Planning and control of coupled fluid–body motion. a**, Sensor-free control ranks candidate paths or force-and-torque sequences using predicted outcomes. **b**, Sensor-based control combines a base feedback action with a correction. **c**, Direct control generates the current correction from observations. **d**, Predictor-based planning evaluates correction sequences using the learned predictor. **e**, Hybrid planning evaluates each candidate with two models: a learned predictor and a rigid-body model with fixed fluid forces and torques.

(1) Direct control. A learned policy serves as the control generator, using the observation history h_n to produce the current force-and-torque correction r_n (Fig. 2c). This correction is added to the base feedback action $u_{\mathrm{FB}}(h_n)$ through Eq. 3.

(2) Predictor-based planning. The control generator uses predictions from F_θ to optimize a force-and-torque correction sequence $R = (r_n, r_{n+1}, \ldots, r_{n+H-1})$ over 1 or 4 s (Fig. 2d), where H is the number of steps in the prediction window and r_n is the first correction. Starting from the current body state $\hat{s}_n = s_n$, we add the first correction to $u_{\mathrm{FB}}(h_n)$, and F_θ predicts the next body state $\hat{s}_{n+1}$ and positive work. The base feedback action u_{FB} is then recomputed from $\hat{s}_{n+1}$, the next correction in R is added, and prediction continues to $\hat{s}_{n+H}$. The generator uses the resulting work and progress towards target B to compare and refine R, while checking collision and stopping requirements within the window. If the selected sequence satisfies these constraints and scores no worse than using u_{FB} alone ($R = 0$), only its first correction r_n is applied through Eq. 3; otherwise, $r_n = 0$. New observations then update h_n, and the sequence is optimized again.

(3) Hybrid planning. Each correction sequence R is evaluated using two forecasts over a window of up to 4 s (Fig. 2e). The learned predictor F_θ updates fluid loads and body motion to produce the predicted states $\hat{s}_{n+1}, \ldots, \hat{s}_{n+H}$. The rigid-body model F_{RB} produces $\hat{s}'_{n+1}, \ldots, \hat{s}'_{n+H}$ by advancing the rigid-body equations, holding the estimated fluid force $\mathbf{F}_{\mathrm{fluid},n-1}$ and torque $\tau_{\mathrm{fluid},n-1}$ constant throughout the window. Both forecasts start from the current body state in h_n, with $\hat{s}'_n = \hat{s}_n = s_n$. Both models evaluate the control with corrections R and the reference using u_{FB} alone ($R = 0$). Each evaluation scores predicted work, progress and timely arrival and stopping. The two model scores are averaged separately for the corrected control and the reference. Both controls must satisfy the task constraints under both forecasts. The first correction is applied when the sequence also passes the progress check relative to the reference and improves the average score by a prescribed margin; otherwise, $r_n = 0$. New observations then update h_n, and the sequence is optimized again. Implementation details are provided in Appendix F.

2.3 Predictors and Control Generators

Predictor candidates. We compare predictors of flow fields, fluid loads and work for energy-efficient motion control. Field predictors include FNO, F-FNO, UNO, U-Net, Transolver and Swin (Li et al., 2021; Tran et al., 2023; Rahman et al., 2022; Ronneberger et al., 2015; Wu et al., 2024; Liu et al., 2021). We also compare RSSM and POD–Kalman (Hafner et al., 2019; Proctor et al., 2016; Kalman, 1960). PathCNN and Path Transformer predict total work directly from a circular-body path. Load-GRU, Load-TCN and Load Transformer predict fluid forces and torques,

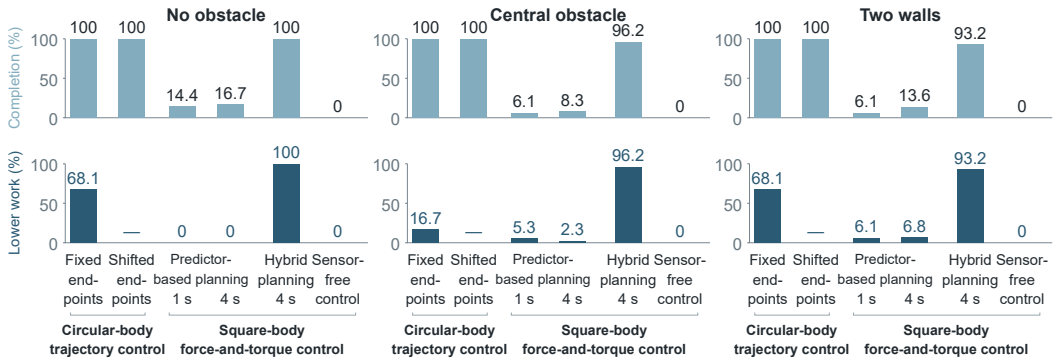

Figure 3: **Task completion and work reduction across control settings.** Columns show the three geometries. Upper bars show task completion; lower bars show completion with less work than the corresponding training reference. Sensor-based planning compares shooting, CEM, MPPI and gradient optimization across the same eleven predictors and three sensor layouts. Hybrid uses fixed four-second predictors. Dashes denote shifted endpoints without a same-task training reference.

which are combined with applied controls to integrate body motion and compute positive work. These model families are described in Appendix A, with training details in Appendix E.

Control generator candidates. For sensor-free control (Fig. 2a), we use CEM, gradient optimization and PPO (de Boer et al., 2005; Schulman et al., 2017) to generate circular-body paths and square-body force-and-torque sequences. For sensor-based control, direct control (Fig. 2c) maps h_n to the current correction r_n using PPO, SAC, TD3, Dreamer (Schulman et al., 2017; Haarnoja et al., 2018; Fujimoto et al., 2018; Hafner et al., 2020), behavioural cloning (BC), Decision Transformer (DT), diffusion policy, IQL or CQL (Chen et al., 2021; Chi et al., 2023; Kostrikov et al., 2022; Kumar et al., 2020). Predictor-based planning (Fig. 2d) optimizes correction sequences R using shooting, CEM, gradient optimization or model predictive path integral control (MPPI) (Williams et al., 2017). Hybrid planning (Fig. 2e) applies the same search methods under both forecasts. It evaluate policy-guided MPPI, which uses correction sequences R^{π} proposed by PPO, SAC, TD3 or Dreamer to guide sampling and evaluates candidates under both forecasts. Implementation details and search budgets appear in Appendices F and B.4, respectively.

3 Results

We compare path planning and force-and-torque control across three channel geometries. OpenFOAM (Weller et al., 1998) evaluates task completion and positive work. Fig. 3 summarizes the results, using four common search methods for sensor-based planning. All circular paths complete both endpoint tasks. Extending the planning window yields modest gains in completion, whereas hybrid planning completes 93.2–100% of configurations across scenes, all with lower work than the training reference. For sensor-free square control, no plan completes the task.

3.1 Circular-body trajectory control and generalization

We use the sensor-free planning framework in Sec. 2.2 to optimize paths for a circular body moving at constant speed. We first compare work against the corresponding best training paths, then shift both endpoints to test generalization without retraining.

Control with training endpoints. We train six field predictors on CFD trajectories and evaluate their velocity-field predictions on test trajectories (Fig. 4a). Errors generally increase during motion and are largest in the two-wall channel. U-Net gives relatively low errors in the unobstructed and central-obstacle channels.

Optimization finds lower-work paths in all three geometries. U-Net–PPO, U-Net–CEM and Transolver–CEM achieve the largest percentage savings in the unobstructed, central-obstacle and two-wall channels, respectively. Their paths require 2.640, 2.554 and 4.072 J, compared with 2.934, 2.614 and 5.478 J for the corresponding training references (Fig. 4b,c). Overall, 48 of the 72 predictor–generator–geometry combinations produce at least one lower-work path, and all 216

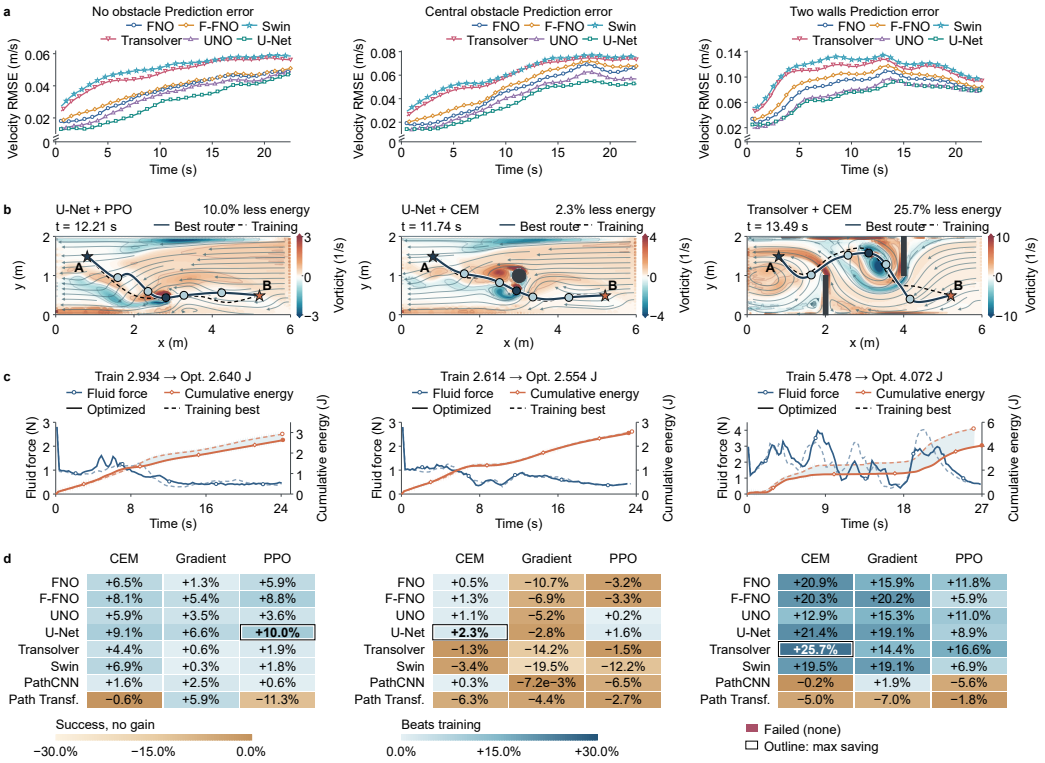

Figure 4: **Circular-body path control with fixed endpoints.** Columns show the unobstructed, central-obstacle and two-wall scenes. **a**, Root mean squared error (RMSE) in predicted velocity; colours and markers identify models. **b**, Paths with the largest percentage work reduction: navy, optimized path; dashed black, training reference; stars, endpoints; circles, body poses. Backgrounds show signed vorticity and streamlines at the indicated time, with scene-specific scales. **c**, Fluid-force magnitude (blue) and cumulative positive work (orange); solid and dashed curves show optimized and reference paths. **d**, Percentage work reduction for each method's lowest-work path, compared with the corresponding model's best training path: blue, savings; amber, increased work. Outlines mark the largest percentage saving in each scene. All evaluated paths are feasible.

evaluated paths satisfy the physical constraints (Fig. 4d). The savings reflect changes in both route length and opposing fluid forces. The largest gain occurs in the two-wall channel, where the optimized path uses 25.7% less work despite being longer than the training reference. Detailed path, force and work analyses are provided in Appendix D.1 and Fig. S6.

Generalization to unseen endpoints. We shift both endpoints and repeat path optimization with the trained predictors fixed. All 216 evaluated paths remain feasible. F-FNO–Gradient, F-FNO–PPO and U-Net–Gradient achieve the lowest work in the unobstructed, central-obstacle and two-wall channels, respectively, at 3.171, 3.193 and 5.466 J (Fig. 5a–c). The optimized routes adjust their obstacle approaches and final ascent to the new endpoints (Fig. 5a). Fluid loads remain low during much of the later motion in the two simpler geometries, whereas work in the two-wall channel is concentrated near the passages (Fig. 5b). Detailed analyses are provided in Appendix D.1 and Fig. S7.

3.2 Square-body force and torque control with sensors

We use the three sensor-based frameworks in Sec. 2.2 to select forces and torques online. The square must reach the destination and stop within the prescribed time. We compare eight fixed pressure sensors, 40% perimeter coverage and full coverage. Prediction is evaluated on CFD (Fig. 6a).

Direct Control. DOB completes the tasks in all three geometries, requiring 49.308 J in the two-wall channel. The BC, DT, diffusion-policy, IQL and CQL baselines show frequent failures, and none achieves lower work than DOB between the walls (Fig. 6e).

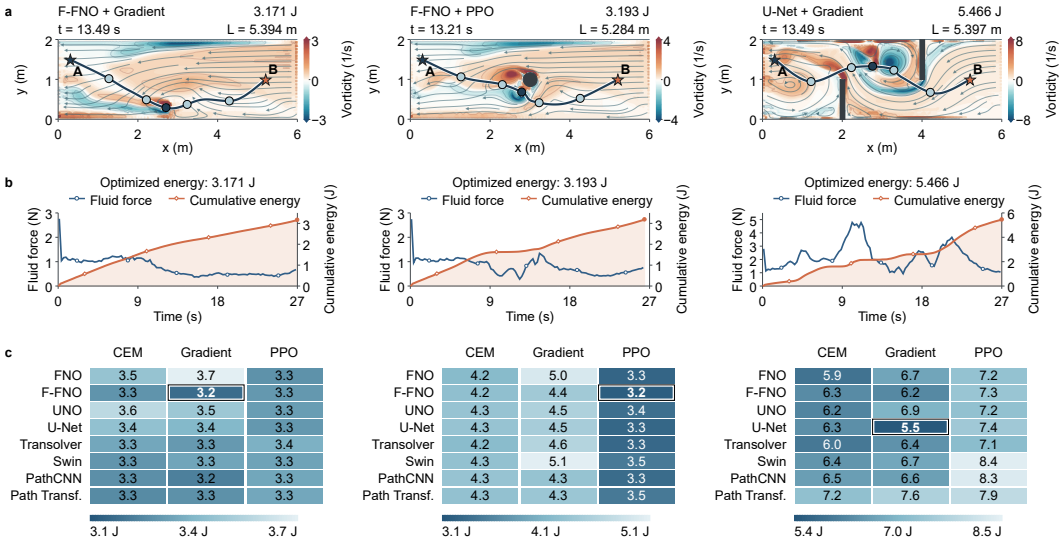

Figure 5: **Circular-body path control after both endpoints move.** Columns show the unobstructed, central-obstacle and two-wall scenes. **a**, Lowest-work paths for the shifted endpoints: navy, optimized route; stars, endpoints; circles, body poses. Backgrounds show signed vorticity and streamlines at the annotated time, with scene-specific scales. The selected model and optimizer, total work and path length are given above each scene. **b**, Fluid-force magnitude (blue) and cumulative positive work (orange) along the selected paths. **c**, Minimum work in joules among three evaluated paths for each model–optimizer pair. Darker blue indicates lower work on each scene's scale; outlines mark the lowest work in each scene.

Predictor-based planning. Across shooting, CEM, MPPI and gradient optimization, extending the planning window from 1 to 4 s increases task completion from 35/396 (8.8%) to 51/396 (12.9%) (Fig. 3). The largest gain occurs in the two-wall channel, where completions increase from 8 to 18. The two windows yield 15 and 12 successful controls below the training-reference work, respectively. The lowest work between the walls decreases from 49.339 to 47.823 J. Detailed comparisons are provided in Appendix D.3 and Figs. S8 and S9.

Hybrid planning. Across 648 hybrid-planning configurations, 615 (94.9%) complete the task, all with lower positive work than the corresponding training reference (Fig. 6e). With the four search methods compared in Fig. 3, hybrid planning completes 382/396 configurations, compared with 51/396 for four-second predictor-based planning using the same trained predictors.

The lowest-work neural controls use 7.653 J in the unobstructed channel and 9.293 J around the central obstacle, using Load-TCN with MPPI and U-Net with gradient optimization, respectively, both with 40% sensor coverage. Between the two walls, U-Net with TD3-guided MPPI and eight pressure sensors requires 35.896 J, 27.2% less than DOB. POD–Kalman with MPPI and 40% coverage achieves the lowest work of 35.321 J. The selected low-work motions reduce translational effort through changes in route and speed (Fig. 6b–d). In the simpler channels, they maintain progress under lower fluid loading. Between the walls, the lowest-work motion follows a shorter route and traverses the descent at lower speed and load. Detailed motion, force and torque analyses are provided in Appendix D.4 and Fig. S10.

More pressure observations do not consistently improve completion. Full coverage gives the most successful predictor–generator combinations around the central obstacle, whereas 40% coverage performs best between the walls. Counts by sensor layout are provided in Appendix C.

3.3 Square-body force and torque control without sensors

We optimize complete force-and-torque sequences using the sensor-free framework in Sec. 2.2 and execute them without feedback. Velocity-prediction errors rise during motion, most strongly in the two-wall channel (Fig. 7a). FNO achieves the lowest test work MAE of 1.821 J. However,

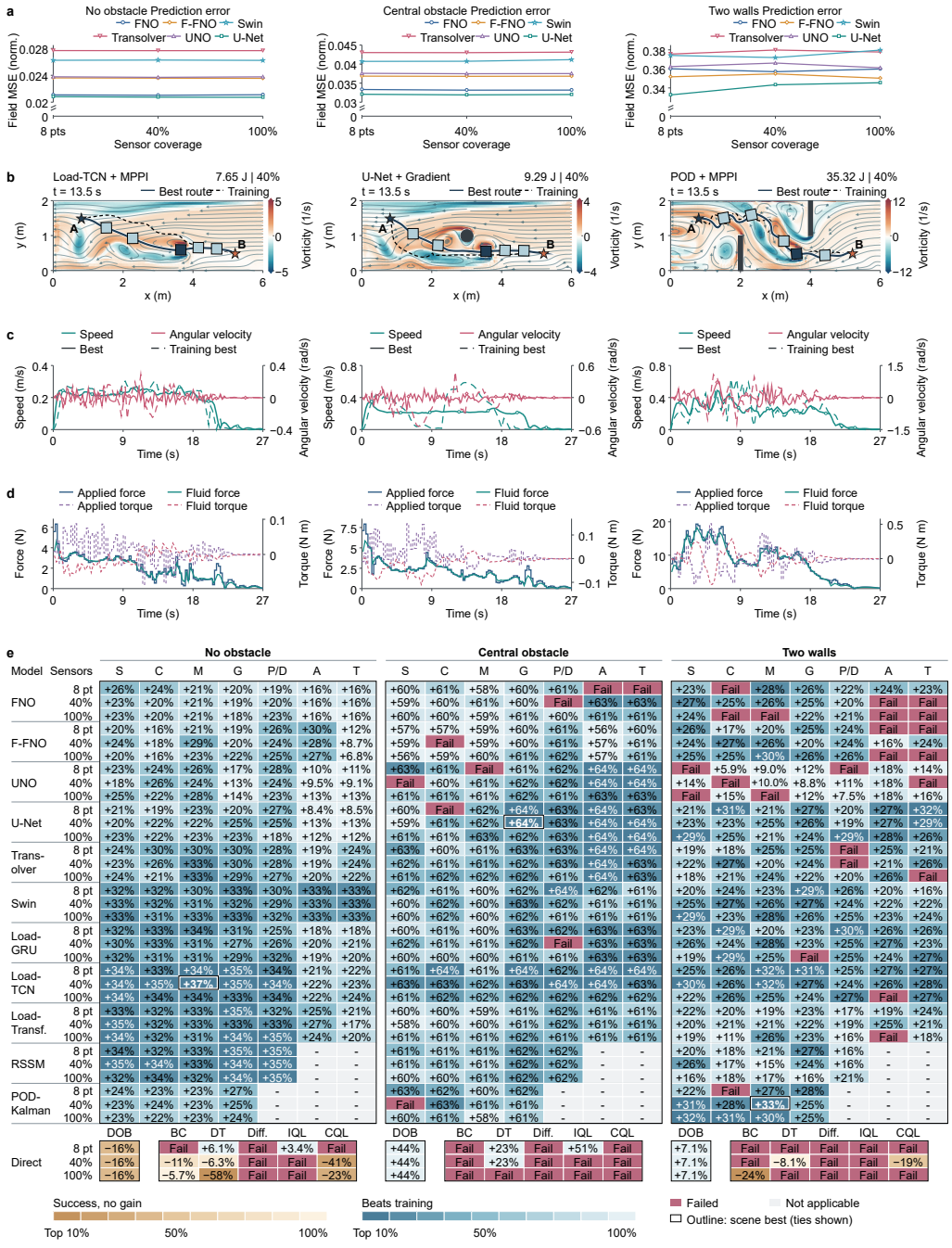

Figure 6: **Boundary observations support energy-efficient force and torque control.** **a**, MSE in normalized velocity and pressure predicted over four seconds, for six models and three sensor layouts. **b**, Lowest-work successful motions. Navy/dashed black: selected/training-reference paths; squares: body poses; stars: endpoints. Vorticity and streamlines are shown at 13.5 s with scene-specific scales. **c**, Speed (teal) and signed angular velocity (pink), with solid/dashed curves for selected/training-reference motions. **d**, Applied/fluid force (blue/teal) and torque (purple dashed/pink dash-dotted); fluid loads are interval averages from momentum balance. **e**, Work reductions from the training minimum: blue, savings; amber, increased work; rose, failure; grey, inapplicable. Colour intensity ranks savings and increases separately within each scene. Outlines mark minima across sensor layouts. S, shooting; C, CEM; M, MPPI; G, gradient. P/D, A and T denote policy-guided MPPI using PPO (Dreamer for RSSM), SAC and TD3, respectively. Completions: 615/648 hybrid-planning configurations and 23/54 direct-control references.

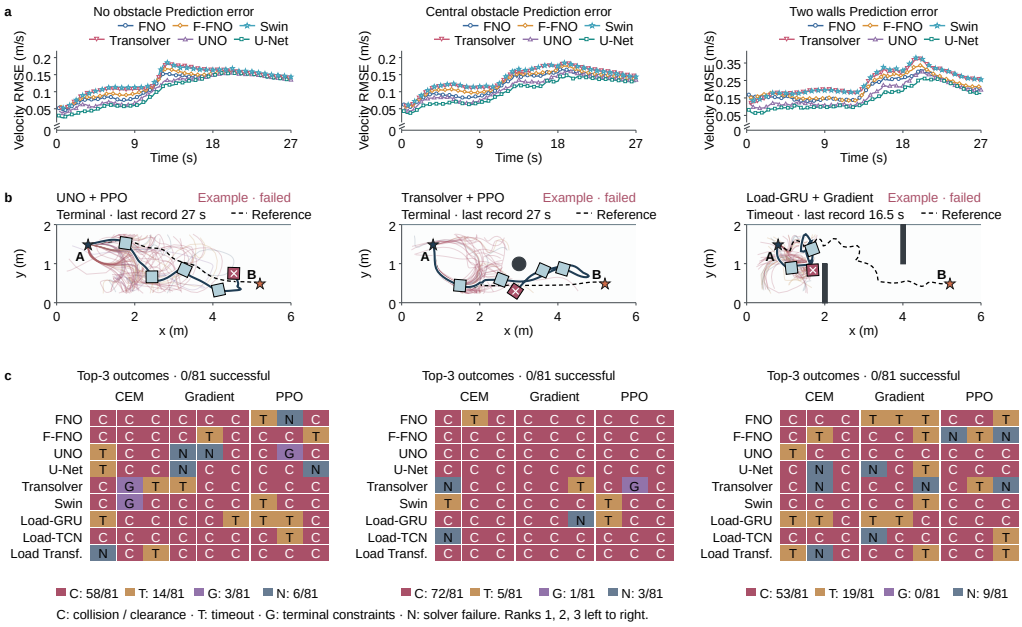

Figure 7: **Sensor-free force and torque control.** Columns show the three channel geometries. **a**, Held-out velocity RMSE versus time for six field predictors, evaluated on ten trajectories per scene; colours and markers identify models. **b**, Illustrative unsuccessful CFD trajectories (navy), with the demonstrated reference dashed black and other submitted paths shown faintly. Stars mark the endpoints, squares the moving body and crosses the last recorded pose. The first two runs fail to reach the required final state within the allowed motion time. The two-wall run ends early after a solver timeout. **c**, Outcomes for nine predictors, three optimizers and three submissions per combination: C, collision or clearance failure (rose); T, solver timeout (ochre); G, terminal-constraint failure (purple); N, solver failure (grey-blue). The matrix contains 243 submitted controls. No submitted control satisfies the complete task.

mean terminal-position and shape-angle errors across predictors exceed the respective tolerances of 0.012 m and 0.75°, highlighting the difficulty of precise stopping (Appendix D.5).

None of the 243 evaluated sequences completes the task (Fig. 7c). Example runs show stalled progress or excessive rotation that prevents successful passage and stopping (Fig. 7b). Failure counts and detailed motion and load analyses are provided in Appendix D.5 and Fig. S11.

4 Discussion

We evaluate learned PDE models through the energy cost of controlled motion, using a CFD-generated dataset and a common framework for comparing physical predictors and control generators under shared task conditions. For circular-body control, we identify paths that require up to 25.7% less work than the corresponding best training trajectories and demonstrate feasible planning for unseen endpoints. For square-body control, hybrid planning improves task completion more than extending the prediction horizon alone. The best neural controllers reduce work by 36.6%, 64.4% and 32.3% relative to the corresponding best training trajectories in the no-obstacle, central-obstacle and two-wall channels, respectively. Sensor-free square-body trials further show how translational and rotational deviations disrupt passage and stopping.

These results connect physical prediction and control in the context of energy-efficient body motion. The shared tasks provide a basis for understanding how prediction quality affects control performance and for improving physical predictors and control generators together. We hope this work will help extend the use of physical operators beyond forward prediction toward more challenging control tasks in complex systems under a range of physical constraints.

AI USE STATEMENT

We used generative AI tools to improve the grammar, wording, clarity and organization of the manuscript. The authors reviewed and revised all AI-assisted text to ensure accuracy and preserve the intended scientific meaning. We take full responsibility for the final content of this work, including all AI-assisted text.

ETHICS STATEMENT

This work studies energy-efficient fluid–body control using simulation-generated data. Real-world deployment would require further validation and safety assessment.

REPRODUCIBILITY STATEMENT

We provide detailed supplementary information to support reproducibility. Appendix B describes the detailed physical settings and task conditions, trajectory generation, sensor settings, and search settings. Appendix E reports the training details and convergence of the predictors and control generators. Appendix F provides additional methodological details for different control problems and CFD integration. The code and dataset will be publicly released after the peer-review process.

REFERENCES

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

Katharina Bieker, Sebastian Peitz, Steven L Brunton, J Nathan Kutz, and Michael Dellnitz. Deep model predictive control with online learning for complex physical systems. *arXiv preprint arXiv:1905.10094*, 2019.

Florent Bonnet, Ahmed Jocelyn Mazari, Paola Cinnella, and Patrick Gallinari. AirfRANS: High fidelity computational fluid dynamics dataset for approximating Reynolds-Averaged Navier–Stokes solutions. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. In *Advances in Neural Information Processing Systems*, 2021.

Wen-Hua Chen, Jun Yang, Lei Guo, and Shihua Li. Disturbance-observer-based control and related methods—an overview. *IEEE Transactions on Industrial Electronics*, 63(2):1083–1095, 2016. doi: 10.1109/TIE.2015.2478397.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems*, 2023.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1724–1734, 2014. doi: 10.3115/v1/D14-1179.

Pieter-Tjerk de Boer, Dirk P. Kroese, Shie Mannor, and Reuven Y. Rubinstein. A tutorial on the cross-entropy method. *Annals of Operations Research*, 134:19–67, 2005. doi: 10.1007/s10479-005-5724-z.

Mohamed Elrefaie, Florin Morar, Angela Dai, and Faez Ahmed. DrivAerNet++: A large-scale multimodal car dataset with computational fluid dynamics simulations and deep learning benchmarks. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, 2018.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2555–2565, 2019.

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

Philipp Holl, Vladlen Koltun, and Nils Thuerey. Learning to control PDEs with differentiable physics. In *International Conference on Learning Representations*, 2020.

Peiyan Hu, Haodong Feng, Hongyuan Liu, Tongtong Yan, Wenhao Deng, Tianrun Gao, Rong Zheng, Haoren Zheng, Chenglei Yu, Chuanrui Wang, Kaiwen Li, Zhi-Ming Ma, Dezhi Zhou, Xingcai Lu, Dixia Fan, and Tailin Wu. RealPDEBench: A benchmark for complex physical systems with real-world data. In *International Conference on Learning Representations*, 2026.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960. doi: 10.1115/1.3662552.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2020.

Christian Lagemann, Sajeda Mokbel, Miro Gondrum, Mario Rüttgers, Yuning Wang, Pol Suárez, Ludger Paehler, Deniz A. Bezgin, Aaron B. Buhendwa, Jared L. Callaham, Samuel Ahnert, Nicholas Zolman, Xiao Shao, Jean-Christophe Loiseau, Nikolaus A. Adams, Matthias Meinke, Wolfgang Schröder, Kai Lagemann, Esther Lagemann, Ricardo Vinuesa, and Steven L. Brunton. The HydroGym reinforcement learning platform for fluid dynamics. *Nature*, 657:369–376, 2026. doi: 10.1038/s41586-026-10917-6.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021.

Ruben Ohana, Michael McCabe, Lucas Meyer, Rudy Morel, Fruzsina J. Agocs, Miguel Beneitez, Marsha Berger, Blakesley Burkhart, Stuart B. Dalziel, Drummond B. Fielding, Daniel Fortunato, Jared A. Goldberg, Keiya Hirashima, Yan-Fei Jiang, Rich R. Kerswell, Suryanarayana Maddu, Jonah Miller, Payel Mukhopadhyay, Stefan S. Nixon, Jeff Shen, Romain Watteaux, Bruno Régaldo-Saint Blancard, François Rozet, Liam H. Parker, Miles Cranmer, and Shirley Ho. The Well: a large-scale collection of diverse physics simulations for machine learning. In *Advances in Neural Information Processing Systems*, volume 37, pp. 44989–45037, 2024.

Joshua L. Proctor, Steven L. Brunton, and J. Nathan Kutz. Dynamic mode decomposition with control. *SIAM Journal on Applied Dynamical Systems*, 15(1):142–161, 2016. doi: 10.1137/15M1013857.

Jean Rabault, Miroslav Kuchta, Atle Jensen, Ulysse Réglade, and Nicolas Cerardi. Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control. *Journal of Fluid Mechanics*, 865:281–302, 2019. doi: 10.1017/jfm.2019.62.

Md Ashiqur Rahman, Zachary E. Ross, and Kamyar Azizzadenesheli. U-NO: U-shaped neural operators. *arXiv preprint arXiv:2204.11127*, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, 2015. doi: 10.1007/978-3-319-24574-4_28.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Daniel MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. PDEBench: An extensive benchmark for scientific machine learning. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 35, 2022.

Alasdair Tran, Alexander Mathews, Lexing Xie, and Cheng Soon Ong. Factorized fourier neural operators. In *International Conference on Learning Representations*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

H. G. Weller, G. Tabor, H. Jasak, and C. Fureby. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in Physics*, 12(6):620–631, 1998. doi: 10.1063/1.168744.

Grady Williams, Andrew Aldrich, and Evangelos A. Theodorou. Model predictive path integral control: From theory to parallel computation. *Journal of Guidance, Control, and Dynamics*, 40 (2):344–357, 2017. doi: 10.2514/1.G001921.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for PDEs on general geometries. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53681–53705, 2024.

APPENDIX CONTENTS

A Related Work **14**
A.1 Physical Prediction and Control 14
A.2 Predictor Candidates 14
A.3 Control Generator Candidates 14

B Data Generation and Settings **15**
B.1 Physical Setting and Task Conditions 15
B.2 Trajectory Generation Setting 15
B.3 Sensor Setting 16
B.4 Search Setting 17

C Results of Predictor-based Planning **19**

D Case Study on Work Reduction **22**
D.1 Sensor-Free Control of Circular-Body Motion 22
D.2 Direct Control of Square-body Motion 22
D.3 Predictor-based Planning of Square-body Motion 22
D.4 Hybrid Planning of Square-body Motion 23
D.5 Sensor-free Control of Square-body Motion 23

E Training Details and Convergence **30**
E.1 Training Predictors for Sensor-free Circular Control 30
E.2 Training Predictors for Sensor-based Square Control 30
E.3 Training Predictors for Sensor-free Square Control 30
E.4 Training Control Generators 31

F Supplementary Method Details **34**
F.1 Details on Sensor-free Control of Circular-body Motion 34
F.2 Details on Predictor-based Planning of Square-body Motion 34
F.3 Details on Hybrid Planning of Square-body Motion 34
F.4 Details on Sensor-free Control of Square-body Motion 35
F.5 Details on CFD integration 35

A Related Work

A.1 Physical Prediction and Control

PDE datasets and predictive methods have developed together. PDEBench and The Well support comparisons across equations and physical regimes (Takamoto et al., 2022; Ohana et al., 2024). AirfRANS and DrivAerNet++ provide aerodynamic data for airfoils and vehicles (Bonnet et al., 2022; Elrefaie et al., 2024), while RealPDEBench pairs physical measurements with numerical simulations (Hu et al., 2026). For control, reinforcement learning and differentiable physics use observations or predicted dynamics to select actions (Rabault et al., 2019; Holl et al., 2020). Recurrent prediction has also been combined with model predictive control and online learning (Bieker et al., 2019). HydroGym provides shared environments for learning and evaluating fluid-control policies (Lagemann et al., 2026).

A.2 Predictor Candidates

For flow field predictors, the models span spectral, convolutional and attention-based architectures. FNO and F-FNO use spectral operators (Li et al., 2021; Tran et al., 2023); UNO and U-Net combine features across resolutions (Rahman et al., 2022; Ronneberger et al., 2015); Transolver uses learned physical slices, and Swin uses shifted-window attention (Wu et al., 2024; Liu et al., 2021). For latent representation predictors, RSSM uses deterministic and stochastic latent states (Hafner et al., 2019). Our POD–Kalman implementation combines a rank-24 joint representation with controlled linear regression and Kalman observation updates (Proctor et al., 2016; Kalman, 1960). In our sensor-based implementations, a history encoder reconstructs the field, after which predictions update fields, boundary pressures, loads and body motion.

PathCNN and Path Transformer estimate work directly from circular-body paths and their geometric context. Load-GRU, Load-TCN and Load Transformer instead predict fluid forces and torques from observation histories using gated recurrence, temporal convolutions and self-attention, respectively (Cho et al., 2014; Bai et al., 2018; Vaswani et al., 2017). The predicted loads are combined with applied controls to integrate rigid-body motion and compute the resulting positive work, allowing candidate controls to be evaluated without reconstructing the full flow field.

A.3 Control Generator Candidates

CEM updates a candidate distribution from an elite set (de Boer et al., 2005); shooting evaluates sampled plans, and gradient search differentiates the predictive objective. MPPI refines a nominal sequence using exponentially weighted perturbations (Williams et al., 2017). Circular paths and sensor-free square motion use CEM, gradient search and PPO to generate complete plans. Sensor-based planners optimize corrections and execute the first action before replanning.

PPO, SAC, TD3 and Dreamer learn action proposals through policy optimization or imagined dynamics (Schulman et al., 2017; Haarnoja et al., 2018; Fujimoto et al., 2018; Hafner et al., 2020). In the predictor-based comparisons, their policies produce the current correction directly; in the hybrid comparison, they propose sequences for MPPI. The offline policies comprise behavioural cloning, Decision Transformer, diffusion policy, IQL and CQL (Chen et al., 2021; Chi et al., 2023; Kostrikov et al., 2022; Kumar et al., 2020). Their training and execution interfaces are specified in Appendices E and F.

B Data Generation and Settings

Both tasks use the same three channels. The circle follows a prescribed path; applied forces and torque drive the square. OpenFOAM records the resulting flow, motion and loads. Fig. S1 shows the initial flows, and Figs. S2 and S3 show the trajectory collections.

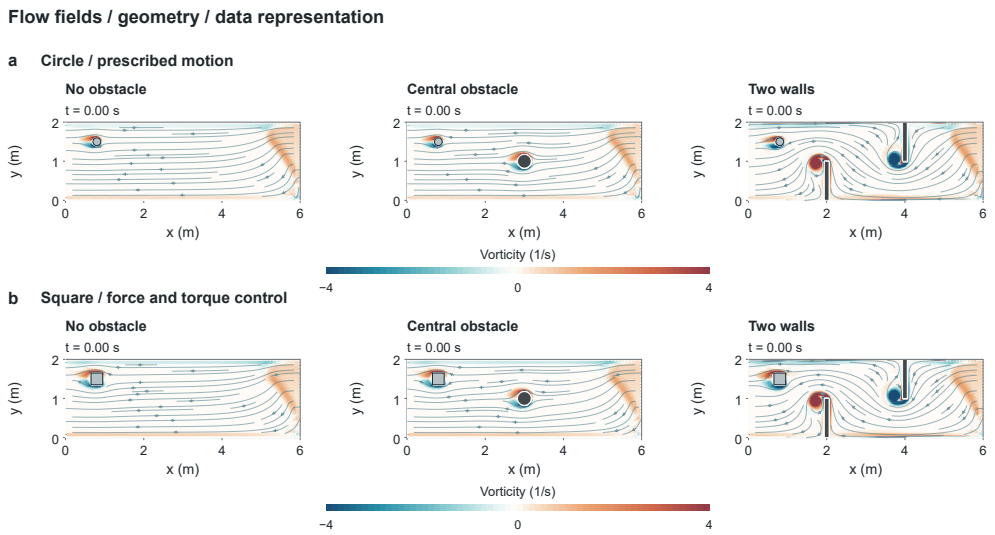

Figure S1: **Initial flow conditions. a,b**, CFD vorticity and streamlines for the circular and square bodies in the unobstructed, central-obstacle and two-wall channels. The body is held at its initial position during 2 s of preflow; the displayed time $t = 0$ marks the start of controlled motion. Streamline arrows show the right-to-left flow.

B.1 Physical Setting and Task Conditions

The channel is 6×2 m with a 0.05 m extrusion for quasi-two-dimensional CFD. Fluid density is 1,000 kg m^{-3} and $Re = 100$. The inlet profile is $u_{f,x}(y) = -0.6(y/2)^2$ m s^{-1}, giving right-to-left flow with a height-averaged speed of 0.20 m s^{-1}. Kinematic viscosities are 4×10^{-4} and 6×10^{-4} m^2 s^{-1} for the circle and square. Each run begins with 2 s of preflow while the body is fixed. The central obstacle has diameter 0.35 m; the two walls have thickness 0.12 m and extend 1 m from opposite boundaries. Velocity components and pressure are represented on a 192×64 training grid after generation on the native CFD mesh.

For body state $s = (\mathbf{x}_b, \mathbf{v}, \phi, \omega)$, the planar rigid-body equations are

$$\dot{\mathbf{x}}_b = \mathbf{v}, \qquad \dot{\phi} = \omega, \qquad m\dot{\mathbf{v}} = \mathbf{F}_{\text{act}} + \mathbf{F}_{\text{fluid}}, \qquad I\dot{\omega} = \tau_{\text{act}} + \tau_{\text{fluid}}. \tag{4}$$

The circle has diameter 0.20 m and travels at 0.20 m s^{-1}. Paths have length at most 5.4 m, duration at most 27 s and clearance at least 0.08 m. Fixed endpoints are $(0.8, 1.5)$ and $(5.2, 0.5)$ m. Endpoint transfer uses $(0.3, 1.5)$ and $(5.2, 1.0)$ m with the trained predictors held fixed.

The square has side length 0.30 m, mass 4.5 kg and moment of inertia 0.0675 kg m^2. It moves from $(0.8, 1.5)$ to $(5.2, 0.5)$ m within $T = 27$ s. Each action is constant for $\Delta t = 0.25$ s, giving 108 actions and 109 recorded states. Force and torque obey $\|\mathbf{F}_{\text{act}}\| \le 100$ N and $|\tau_{\text{act}}| \le 1.2$ N m. Collision checks use the circumscribed-circle envelope with a 0.025 m margin. At 26.25, 26.5, 26.75 and 27 s, position error must be at most 0.012 m, speed at most 0.015 m s^{-1}, shape-angle error modulo $\pi/2$ at most $\pi/240$, and angular speed at most 0.015 rad s^{-1}.

B.2 Trajectory Generation Setting

Circular paths use random control points, Catmull–Rom smoothing and arc-length sampling. The field dataset contains 600 completed trajectories, 200 per geometry. The scene-wise train-

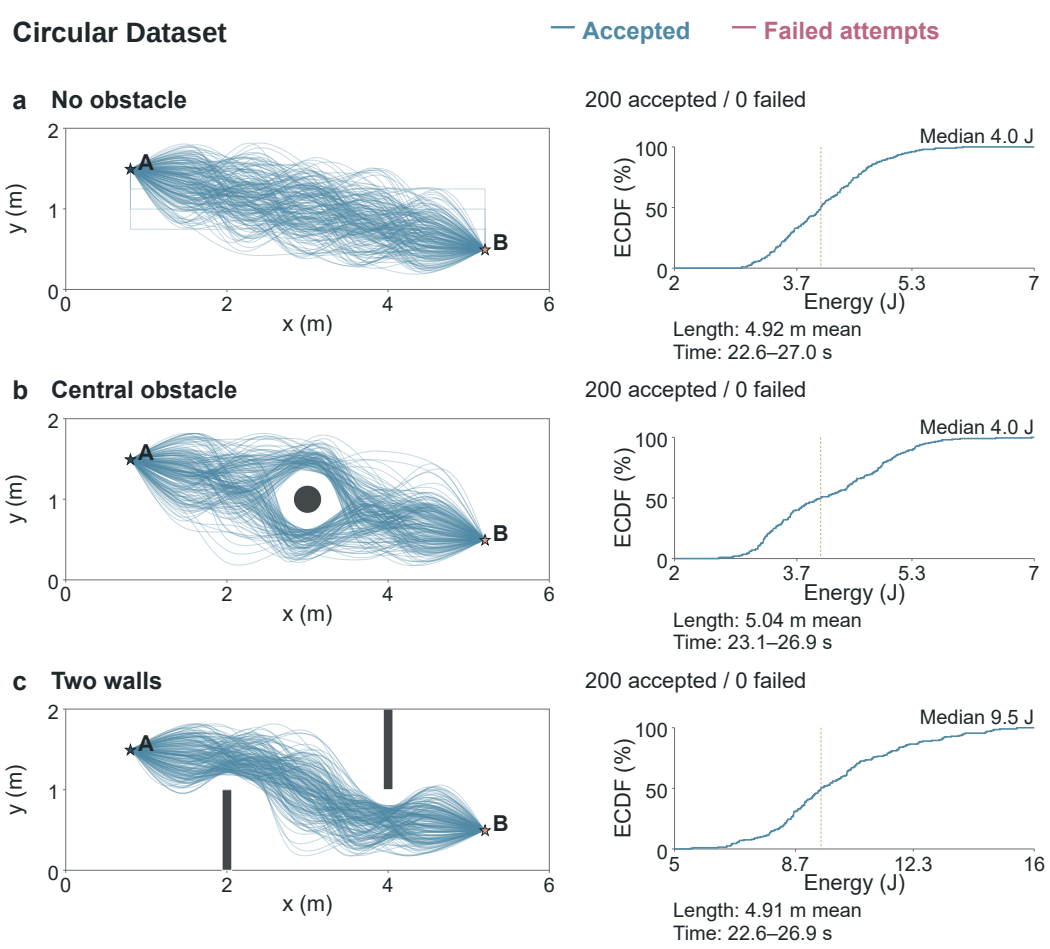

Figure S2: **Circular-body trajectories and work distributions. a–c**, Unobstructed, central-obstacle and two-wall channels, each containing 200 recorded paths. Left, all centre trajectories (blue), with fixed obstacles in grey and spatial endpoints marked A and B. Right, the empirical cumulative distribution function (ECDF) of positive work: the ordinate gives the percentage of paths with work at or below the abscissa. Dotted lines mark median work. Annotations give mean path length and the range of motion durations. All 600 CFD trajectories in this field dataset finish successfully.

ing/validation/test partitions are 162/22/16, 164/16/20 and 160/22/18, giving 486/60/54 overall. The field collections have median work of 4.035, 4.032 and 9.479 J, respectively, with mean path lengths of 4.915, 5.044 and 4.905 m.

Square demonstrations use feedback-generated actions and ten prescribed rotation modes. Acceptance checks the terminal conditions and the unwrapped terminal angle within 5° of its requested value. The collection contains 300 accepted motions, 100 per geometry, and 284 unsuccessful or interrupted attempts. Median accepted work is 29.965, 35.120 and 86.134 J. Of the unsuccessful attempts, 281 fail terminal conditions, one collides, one ends with a solver timeout and one is interrupted. Sensor-free predictors use 80/10/10 accepted trajectories per geometry; sensor-based predictors use 70/15/15. Splits are made by complete trajectory.

B.3 Sensor Setting

The square's perimeter has 120 pressure bins of length 1 cm, populated from CFD boundary faces by face-area weighting. The eight-point mask selects indices $0, 15, \ldots, 105$. The 40% mask selects indices with remainder zero or two modulo five, giving 48 bins; full coverage uses all 120. An

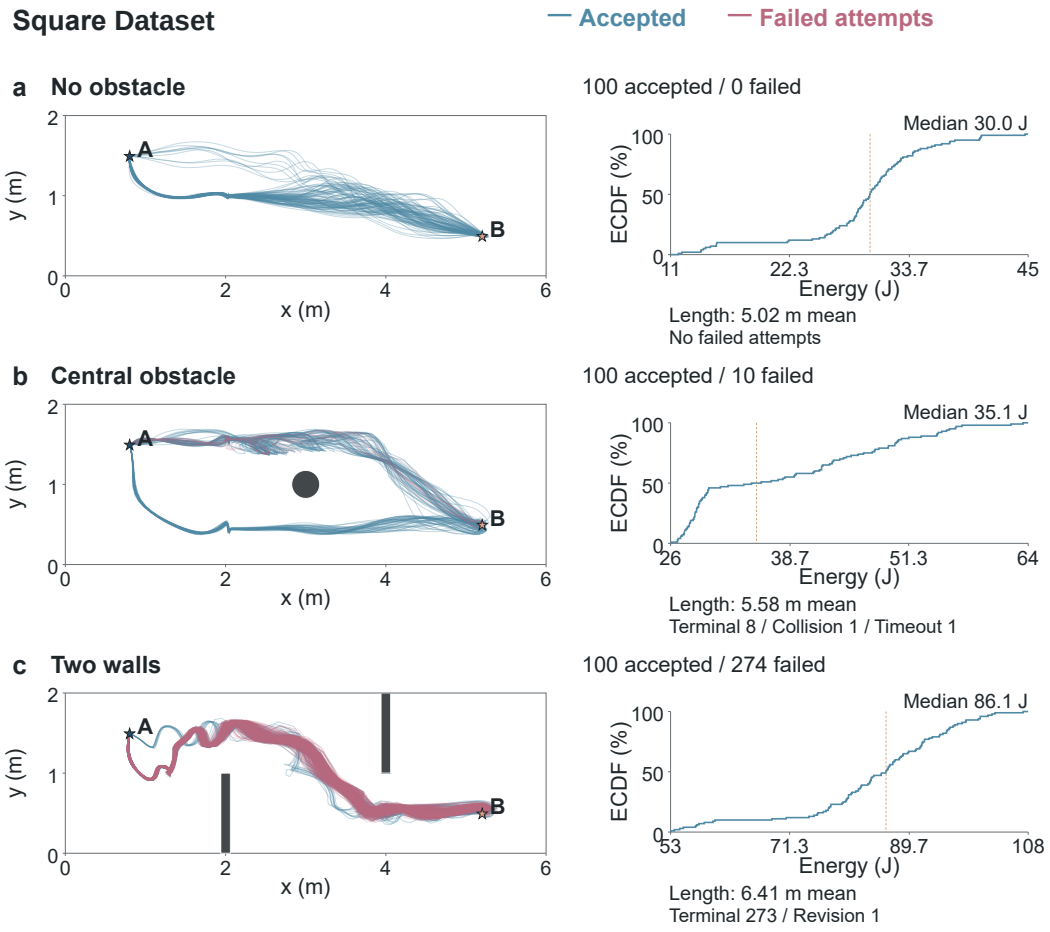

Figure S3: **Square-body demonstrations and terminal-control difficulty. a–c**, Unobstructed, central-obstacle and two-wall channels. Left, all 300 accepted trajectories (blue) and 284 unsuccessful or interrupted attempts (rose); grey shapes denote obstacles and A/B mark the spatial endpoints. Right, ECDFs of positive actuator work for the 100 accepted motions in each scene, with dotted median lines and mean path lengths. Failure annotations distinguish terminal conditions, collision and solver timeout; *Revision* marks one interrupted generation run. Dense rose paths near B in **c** reflect the difficulty of satisfying orientation, angular-speed and holding requirements after reaching the destination.

observation contains 120 masked pressures, 120 mask indicators and 17 features describing body state, previous action, time, destination and geometry. The history h_n contains eight observations with validity indicators at the start of a run. Each predictor is trained with all three masks and evaluated with each mask separately.

Sensor-based control receives both boundary pressure and body-state observations during execution. The load observer uses recorded velocities and known actions, as defined in Eq. 8. Sensor-free square control receives the prescribed initial condition, geometry and target, then executes its selected action sequence without online pressure or body-state observations.

B.4 Search Setting

Circular control pairs eight predictors with three generators in three geometries. Each pair submits three paths, giving 216 CFD evaluations for each endpoint setting. CEM uses 64 candidates for 16 iterations; gradient search uses 16 restarts with 80 steps; PPO uses 128 samples over 60 search iterations. Sensor-free square control pairs nine predictors with three generators and submits three controls per geometry, giving 243 evaluations. Its search uses 1,280 optimization rollouts followed

by screening of 200 candidates. Sensor-based planning has a budget of 8,192 model-transition equivalents per decision, covering the predictive computations used by that framework.

Completion and completion-with-work-improvement are reported over all evaluated configurations. The latter requires both task completion and $E < E_{\mathrm{ref}}$. Percentage savings are $100(E_{\mathrm{ref}} - E)/E_{\mathrm{ref}}$, where E_{ref} is the minimum work in the corresponding model's training population. For the square, sensor-based references are 12.065, 26.104 and 53.051 J; sensor-free references are 11.734, 26.104 and 53.051 J. Circular references are shared within the field-predictor and path-predictor groups (Appendix D.2). Fixed-endpoint illustrations select the largest relative saving, whereas shifted-endpoint illustrations select the lowest absolute work. The matrices summarize model–generator–geometry–mask configurations; three submitted plans are three selected candidates from a search.

C Results of Predictor-based Planning

Figs. S4, S5 and 6 report 702 configurations in each full comparison, including direct-control references. The same CQL policy and its CFD evaluations serve as a shared direct-control reference across all three comparisons. The 1 s and 4 s comparisons contain 61 and 82 completions, respectively, of which 27 and 28 use less work than the training reference. Scene-wise completions rise from 31/15/15 to 36/20/26. Predictor-based 4 s planning and hybrid planning use the same trained predictors; their scoring, action mapping and acceptance rules are specified separately in Appendix F.

P/D, A and T denote direct PPO or Dreamer, SAC and TD3 in the 1 s and 4 s matrices. In the hybrid matrix, they denote policy-guided MPPI. The separately labelled Direct block contains the observation-based reference controllers. In the full hybrid comparison, completion is 227/234, 213/234 and 198/234 across the three geometries. Its 64 failures comprise 48 collision or clearance failures, 14 terminal failures and two solver failures. These outcomes show the gain from the complete hybrid control configuration.

For eight sensors, 40% coverage and full coverage, hybrid-comparison completions are 75/78, 76/78 and 76/78 in the unobstructed channel; 71/78, 69/78 and 73/78 around the central obstacle; and 66/78, 70/78 and 62/78 between the walls. Overall, 638 configurations complete the task and 626 complete it with lower work than the training reference.

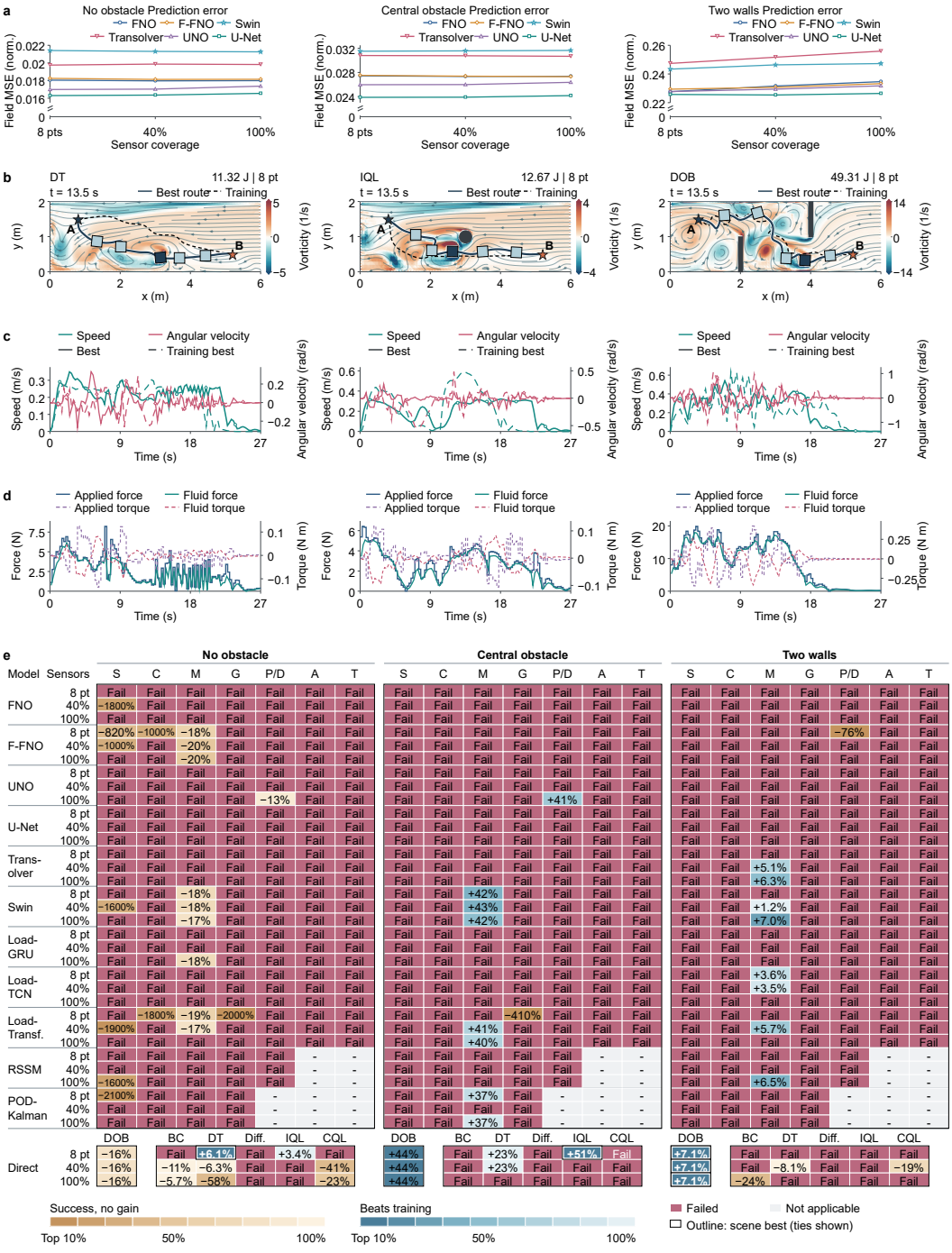

Figure S4: **Sensor-based control with 1 s prediction.** **a**, Test MSE in normalized velocity and pressure for six predictors and three sensor layouts. **b**, Lowest-work controls: DT, IQL and DOB, respectively, all with eight pressure sensors. Navy/dashed black: selected/reference paths; squares: body poses; stars: endpoints. Vorticity and streamlines are shown at 13.5 s. **c**, Speed (teal) and angular velocity (pink); solid/dashed curves show selected/reference motions. **d**, Applied/fluid force (blue/teal) and torque (purple/pink). **e**, Of 702 evaluations, 61 succeed and 27 improve work. Blue/amber: savings/increases; rose: failure; grey: inapplicable. Colour intensity ranks savings and increases separately within each scene; outlines mark minimum work. S, shooting; C, CEM; M, MPPI; G, gradient. P/D, A and T denote direct PPO (Dreamer for RSSM), SAC and TD3 control. Other baselines are named individually.

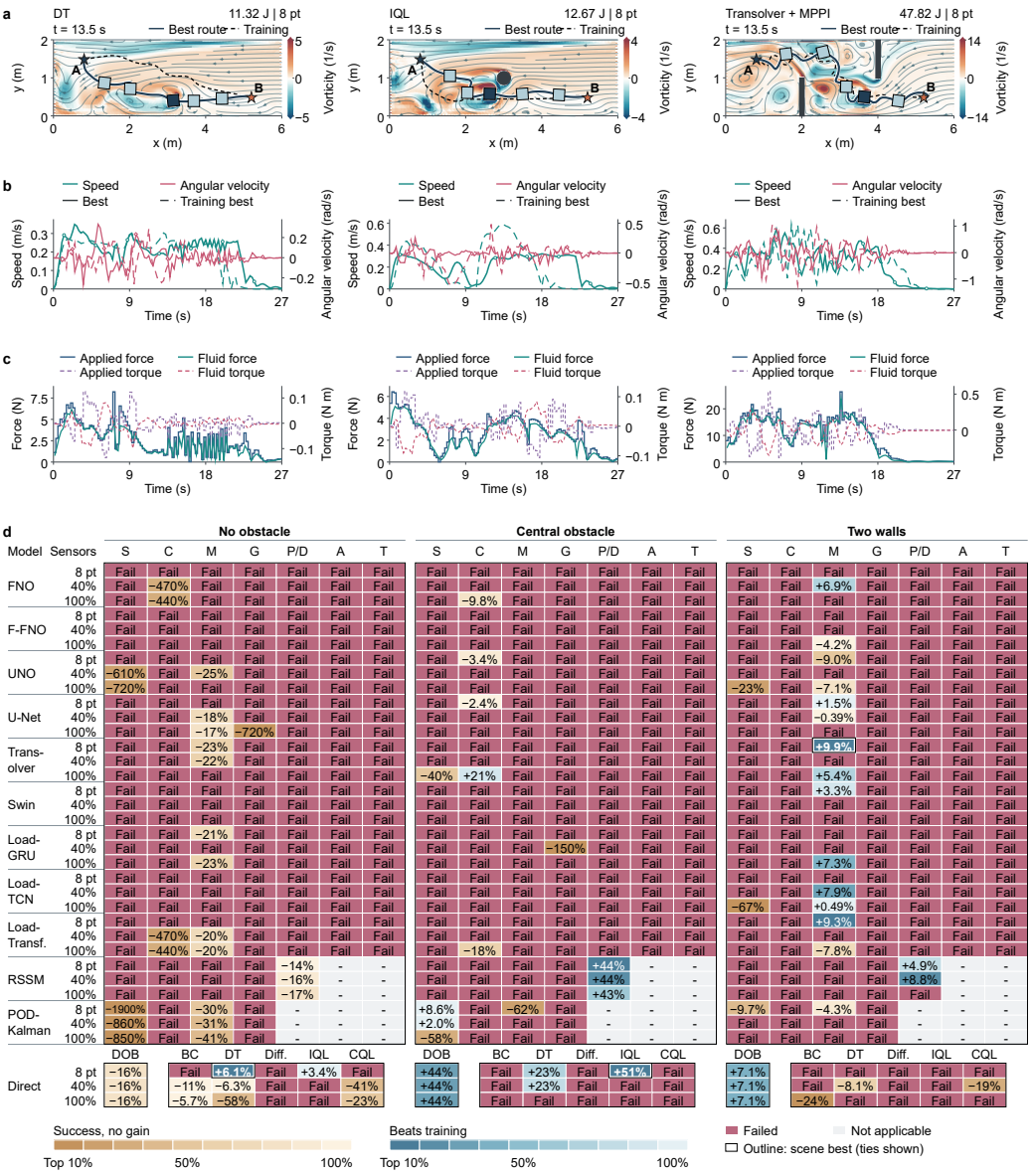

Figure S5: **Sensor-based control with 4 s prediction.** Prediction and planning each span sixteen 0.25 s steps. **a**, Lowest-work successful motions in each scene: DT, IQL and Transolver–MPPI with eight pressure sensors. Navy paths, body squares and endpoint stars describe the selected motion; dashed black paths denote the training reference. Vorticity snapshots and dark body poses correspond to 13.5 s, with separate scene scales. **b**, Speed (teal) and signed angular velocity (pink); solid/dashed curves show selected/training-reference motions. **c**, Applied and fluid-force magnitudes (blue and teal) and signed torques (purple dashed and pink dash-dotted). **d**, Complete results for 702 evaluations, with 82 successes and 28 improvements over the training minimum. Blue denotes savings, amber increased work, rose failure and grey inapplicable combinations; colour intensity ranks savings and increases separately within each scene, and outlines mark the lowest work. S, shooting; C, CEM; M, MPPI; G, gradient. P/D, A and T denote direct PPO (Dreamer for RSSM), SAC and TD3 control, respectively. Other baselines are named individually. Predictor-based and hybrid four-second planning use the same trained predictors.

D CASE STUDY ON WORK REDUCTION

The force and work histories locate the stages that contribute to the final work difference. The highlighted one-second intervals maximize

$$G(t) = \big[E_{\mathrm{ref}}(t+0.5) - E_{\mathrm{ref}}(t-0.5)\big] - \big[E(t+0.5) - E(t-0.5)\big]. \tag{5}$$

Centres are sampled every 0.25 s over the common recorded duration. We select the largest value and then the largest non-overlapping interval, labelled A and B in time order. These windows describe local differences between the executed trajectories. The full histories show the net outcome. Circular work depends on $[-\mathbf{F}_{\mathrm{fluid}} \cdot \mathbf{v}]_+$. Pressure and viscous contributions are integrated over samples with positive total power. Square-body translational and rotational contributions use the same positive-total-power samples. These signed contributions sum to the reported positive work.

D.1 SENSOR-FREE CONTROL OF CIRCULAR-BODY MOTION

Control with training endpoints. In the unobstructed channel, U-Net–PPO reduces work from 2.934 to 2.640 J while path length and mean force magnitude remain similar (Fig. S6a). Its steeper descent changes the force component opposing motion. The two highlighted windows save 0.295 J, close to the full-path reduction of 0.294 J.

Around the central obstacle, U-Net–CEM shortens the route from 4.712 to 4.646 m and reduces work from 2.614 to 2.554 J (Fig. S6b). Both distance and work per metre decrease. Between the walls, Transolver–CEM increases route length from 5.207 to 5.396 m but reduces work from 5.478 to 4.072 J (Fig. S6c). Work per metre falls by 28.3%; reduced pressure work accounts for about 89% of the net saving. The two windows near the second wall contribute 0.798 J of the 1.407 J reduction. All 216 submitted paths are feasible, and 48 of 72 model–generator–geometry groups improve on their own training reference. The illustrated fixed-endpoint paths maximize relative saving, using each model's training minimum as the reference.

Generalization to unseen endpoints. All 216 paths remain feasible after the endpoints change. F-FNO–Gradient, F-FNO–PPO and U-Net–Gradient give the lowest works of 3.171, 3.193 and 5.466 J in the three geometries (Fig. S7a–c). The first two routes descend before approaching the raised destination; the wall route changes the timing and direction of both passages. Original-endpoint trajectories provide spatial and flow context in the figure. Performance for the changed task is compared in joules in Fig. 5c.

D.2 DIRECT CONTROL OF SQUARE-BODY MOTION

We consider only the case $r_n = 0$, corresponding to pure DOB control. The minimum-work training trajectory provides an empirical reference for assessing improvements over the motions recorded in the dataset, while DOB serves as a feedback-control baseline for the same A-to-B task. Table 1 reports both comparisons for the three neural controllers highlighted in the main text. All three controllers require less work than both references across all three geometries.

Prescribed rotation modes diversify the square-body demonstrations. Their acceptance check compares the unwrapped terminal angle with the requested net turn, whereas control evaluation uses the terminal shape angle modulo $\pi/2$. The three reference trajectories have requested net turns of 0°, -90° and 0° (Table 1). A zero net turn allows rotation during motion. The central-obstacle training subset also contains zero-net-turn demonstrations: its lowest-work example, path 0010, uses 28.629 J. The selected U-Net–Gradient control uses 9.293 J, 67.5% less than this reference. Its improvement over recorded demonstrations therefore persists when the reference is restricted to zero net turn. These comparisons concern arrival and stopping under the evaluation conditions; reproducing a demonstration's angular history is not part of the control task.

D.3 PREDICTOR-BASED PLANNING OF SQUARE-BODY MOTION

DT and IQL produce the same selected low-work motions in the first two geometries in both comparisons (Figs. S8a,b and S9a,b). DT uses 11.324 J, a 6.1% saving, with lower loading during its descent. IQL uses 12.674 J, a 51.4% saving, by passing beneath the obstacle with smaller applied and fluid forces during the reference trajectory's costly stage.

Table 1: **Work relative to training demonstrations and DOB.** The selected controls are Load-TCN–MPPI, U-Net–Gradient and U-Net with TD3-guided MPPI, respectively. Reference IDs identify trajectories within each geometry; requested net turns apply to demonstration generation. All work values are in joules. Reductions are $\Delta_{\mathrm{train}} = 100(E_{\mathrm{ref}} - E)/E_{\mathrm{ref}}$ and $\Delta_{\mathrm{DOB}} = 100(E_{\mathrm{DOB}} - E)/E_{\mathrm{DOB}}$, in percent.

Geometry	Ref. ID	Net turn	E_{ref}	E	E_{DOB}	Δ_{train}	Δ_{DOB}
No obstacle	0040	0°	12.065	7.653	13.972	36.6	45.2
Central obstacle	0086	-90°	26.104	9.293	14.721	64.4	36.9
Two walls	0080	0°	53.051	35.896	49.308	32.3	27.2

Between the walls, DOB uses 49.308 J in the one-second comparison (Fig. S8c). Transolver–MPPI uses 47.823 J with four-second planning, compared with the 53.051 J training reference (Fig. S9c). The selected motion crosses the first wall and descends with lower speed and load. Its highlighted windows save 13.315 J, partly offset by additional work at the second passage, leaving a net reduction of 5.228 J.

D.4 Hybrid Planning of Square-body Motion

Load-TCN–MPPI uses 7.653 J in the unobstructed channel, 36.6% below the training reference (Fig. S10a). Its route is only slightly shorter, while mean applied force falls from 2.763 to 1.897 N. Around the central obstacle, U-Net–Gradient uses 9.293 J, a 64.4% reduction (Fig. S10b). Both selected motions sustain translation through the lower channel with smaller loads.

Between the walls, POD–Kalman–MPPI uses 35.321 J (Fig. S10c). Mean applied force decreases from 11.314 to 8.083 N, and the route shortens from 6.073 to 5.625 m. The two windows after the first wall contribute 12.392 J of the total 17.730 J saving. The best neural combination, U-Net with TD3-guided MPPI, uses 35.8962 J, or 32.3% less than the training reference. Both results retain the terminal stopping requirements.

D.5 Sensor-free Control of Square-body Motion

On 30 held-out square trajectories, FNO gives the lowest pooled work MAE at 1.821 J. Across the six field predictors, mean terminal-position errors range from 0.0236 to 0.0364 m and mean shape-angle errors from 16.93° to 23.76°. These are prediction errors; Fig. 7a also reports velocity-field error.

None of the 243 selected sensor-free controls completes the task. Collision or clearance failures, solver-related interruptions and terminal failures number 58/20/3, 72/8/1 and 53/28/0 in the three geometries, respectively. Overall, these comprise 183 collision or clearance failures, 56 solver-related interruptions and four terminal failures. The selection procedure is given in Appendix F.4.

Fig. S11a,b shows departures from the reference motion followed by terminal failure. In panel S11c, the body remains before the first wall and the record ends at 16.50 s. The histories connect changes in translation and rotation to the work accumulated during these unsuccessful executions.

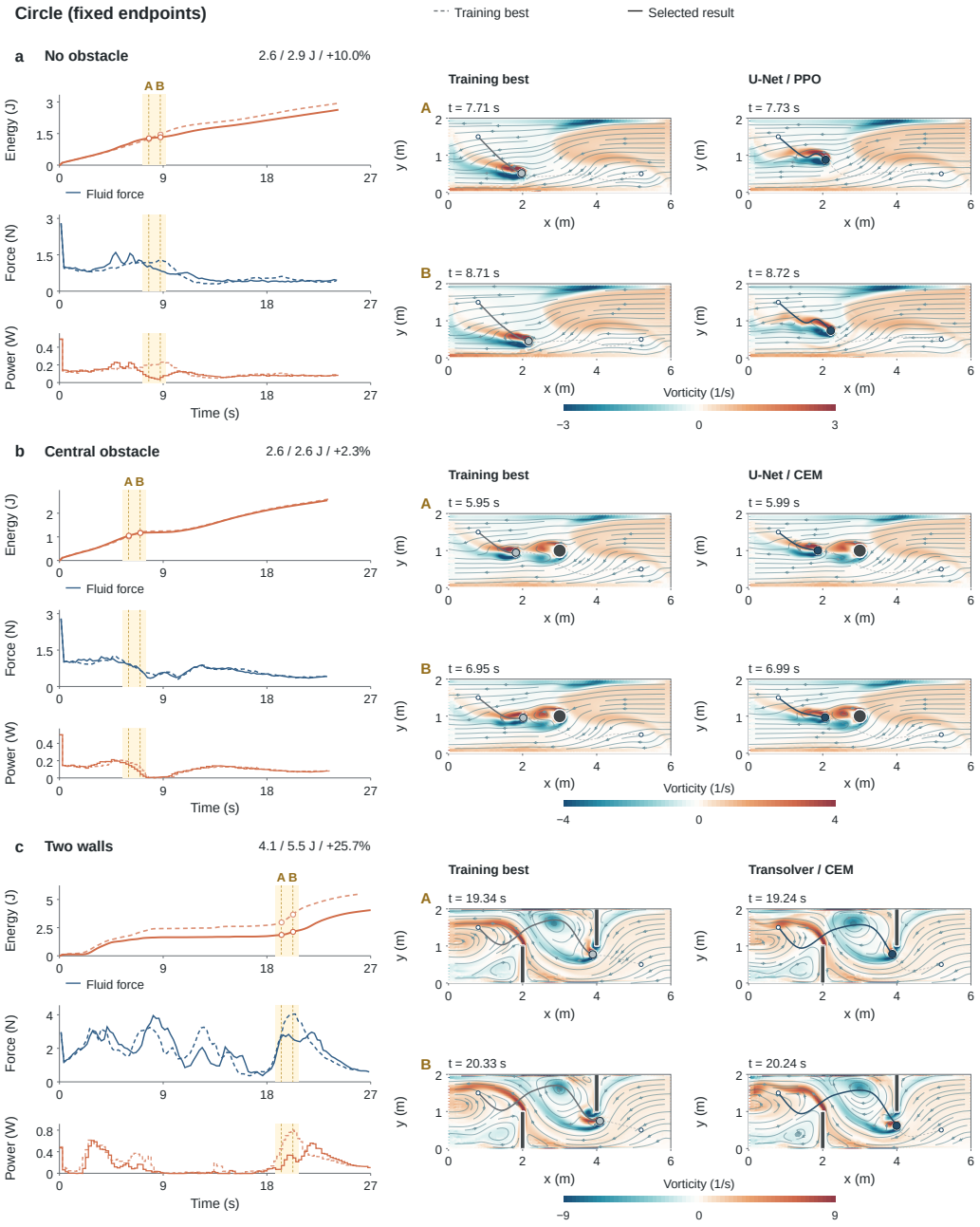

Figure S6: **Where fixed-endpoint paths reduce work.** **a–c**, U-Net–PPO, U-Net–CEM and Transolver–CEM in the unobstructed, central-obstacle and two-wall channels. Left, cumulative positive work (orange), interval-averaged fluid-force magnitude (blue) and interval-averaged positive power (orange); dashed/solid curves denote the training reference/selected path. Gold bands mark one-second windows A and B selected by Eq. 5. Right, reference and selected CFD vorticity fields at each window centre, with actual snapshot times shown. Streamlines and arrows describe the flow; dark paths show the motion up to that time and pale paths the complete routes. Each scene shares one signed-vorticity scale across its four snapshots. Values above each scene give selected/reference work and percentage saving.

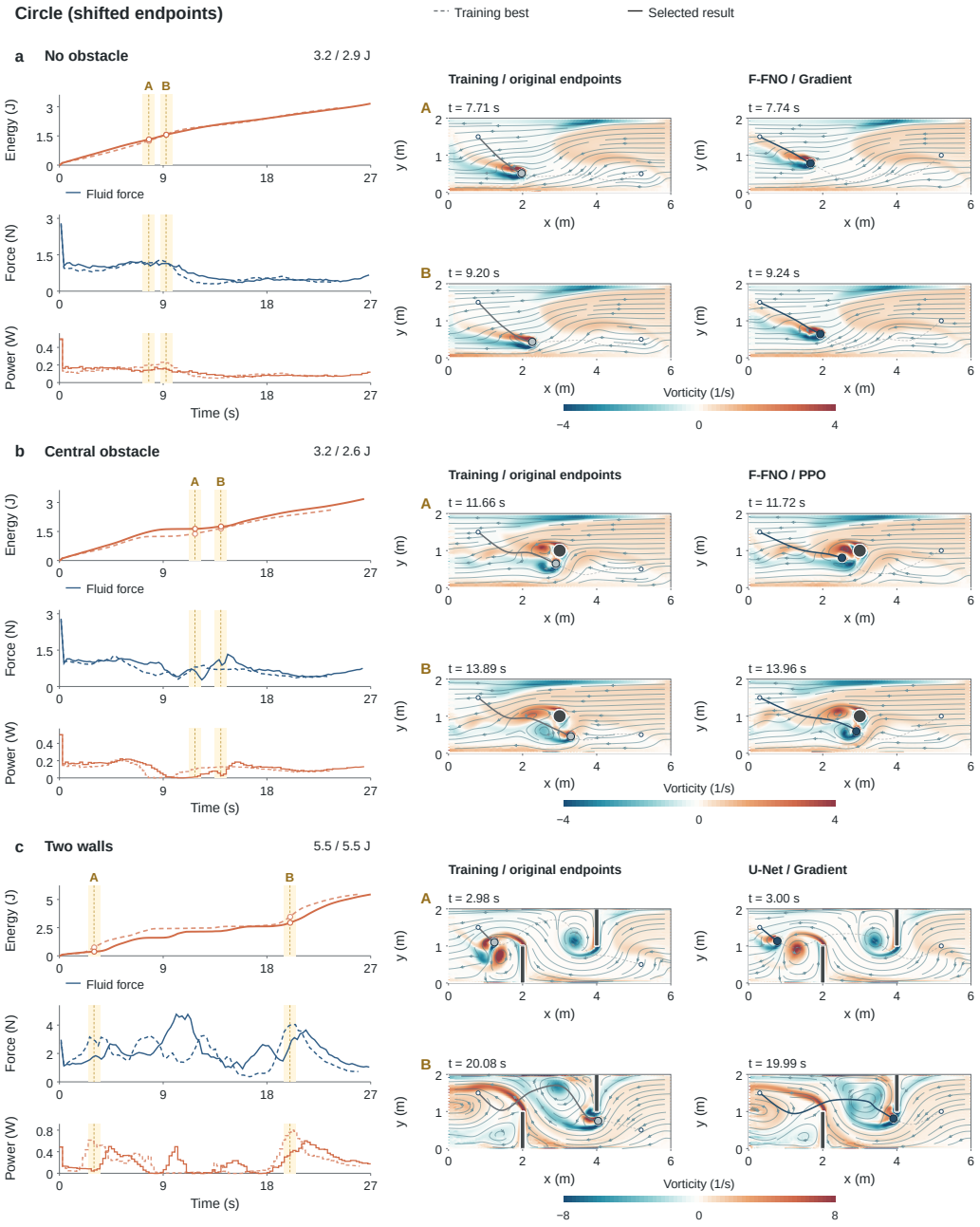

Figure S7: **Flow and work along paths with shifted endpoints. a–c**, F-FNO–Gradient, F-FNO–PPO and U-Net–Gradient in the three geometries. Solid curves and right-hand snapshots show the selected changed-endpoint paths; dashed curves and left-hand snapshots show the original-endpoint training references. Left, cumulative work, fluid-force magnitude and positive power. Right, CFD vorticity, streamlines and body paths in windows A and B, selected as in Fig. S6. Actual snapshot times and scene-specific vorticity scales are shown. The original-endpoint paths provide context for changes in route and task progress; changed-endpoint performance is reported as absolute work.

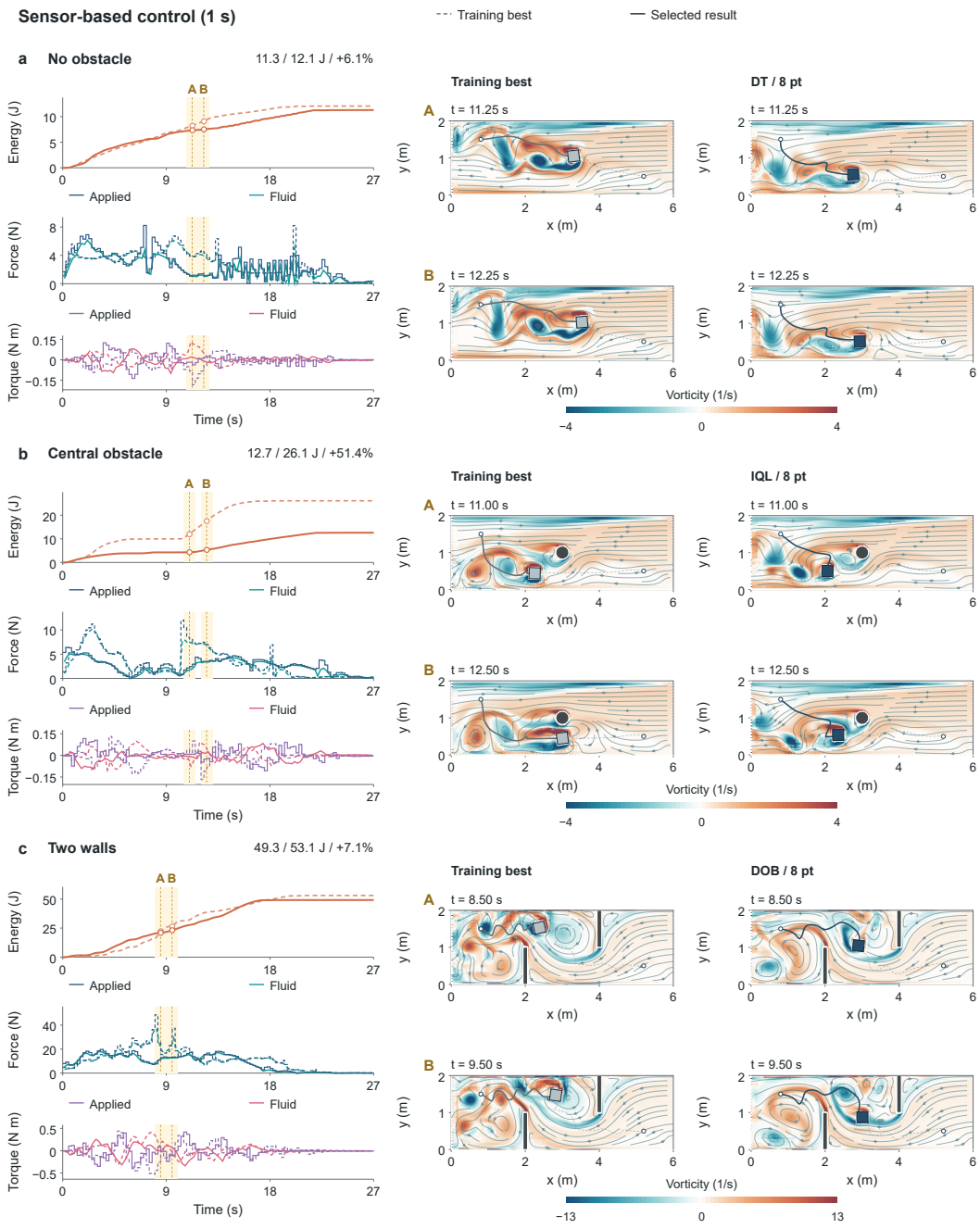

Figure S8: **Sources of work reduction in the one-second comparison. a,b**, DT and IQL in the unobstructed and central-obstacle channels. **c**, DOB between two walls. All three controls use the eight-sensor setting. Left, cumulative positive work (orange), applied/fluid-force magnitudes (blue/teal) and applied/fluid torques (purple/pink). Dashed/solid curves denote the training reference/selected control. Gold bands mark one-second windows A and B. Right, reference/selected CFD vorticity and streamlines at the annotated times, with completed paths in dark blue and complete routes shown faintly. Each scene uses a common colour scale across its four snapshots. Window selection follows Eq. 5.

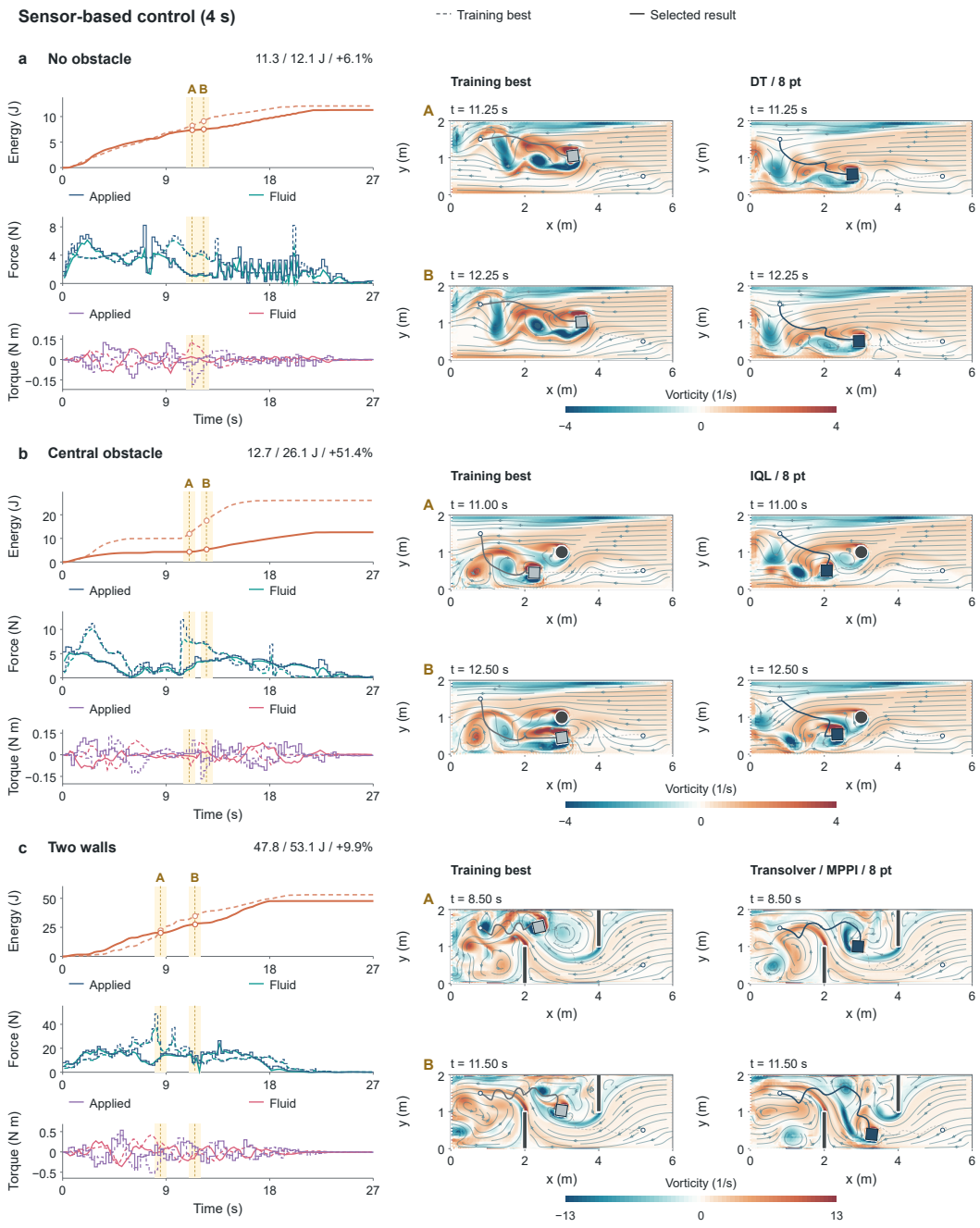

Figure S9: **Work redistribution with four-second planning.** **a,b**, DT and IQL with eight pressure sensors in the first two channels. **c**, Transolver–MPPI with eight pressure sensors between the walls. Left, cumulative work, applied and fluid-force magnitudes, and signed torques; colours and dashed/solid reference/selected curves follow Fig. S8. Gold windows A and B identify large local reductions in work. Right, paired CFD vorticity and streamline snapshots at those times, with body positions and paths. The first-wall passage produces large savings in **c**; the complete work curve also shows the additional expenditure during the later passage. The final wall-task work is 47.823 J.

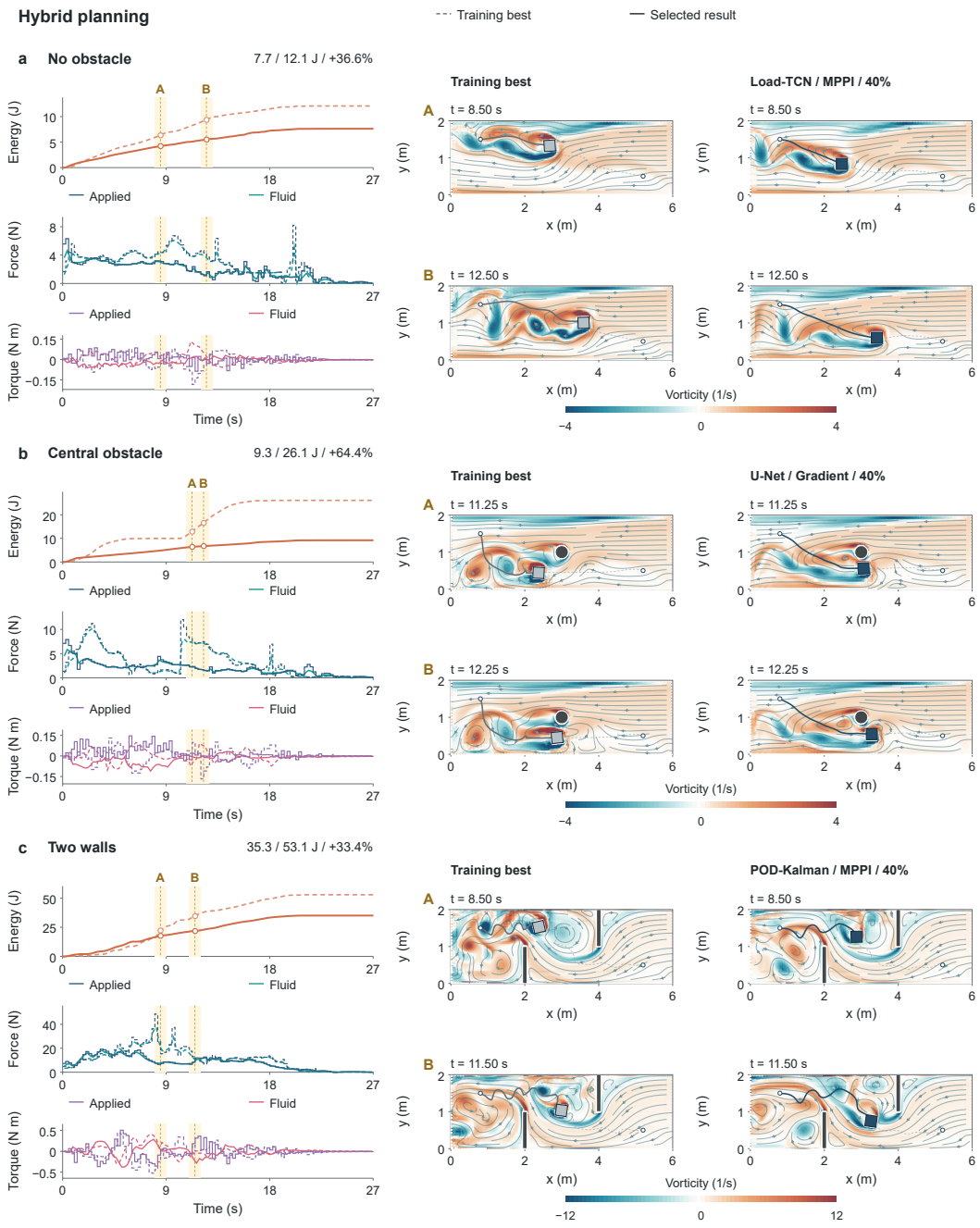

Figure S10: **Motion and fluid loading in the hybrid comparison. a**, Load-TCN–MPPI with 40% pressure coverage. **b**, U-Net with gradient optimization and 40% coverage. **c**, POD–Kalman–MPPI with 40% coverage. Left, cumulative positive work, applied/fluid-force magnitudes and applied/fluid torques; dashed curves show the training reference and solid curves the selected controller. Colours follow Fig. S8. Gold windows A and B link the work differences to paired CFD snapshots on the right. Backgrounds show signed vorticity, with streamlines, flow arrows and body paths. The first two scenes benefit from sustained low-load translation; in **c**, much of the saving occurs during descent after the first wall.

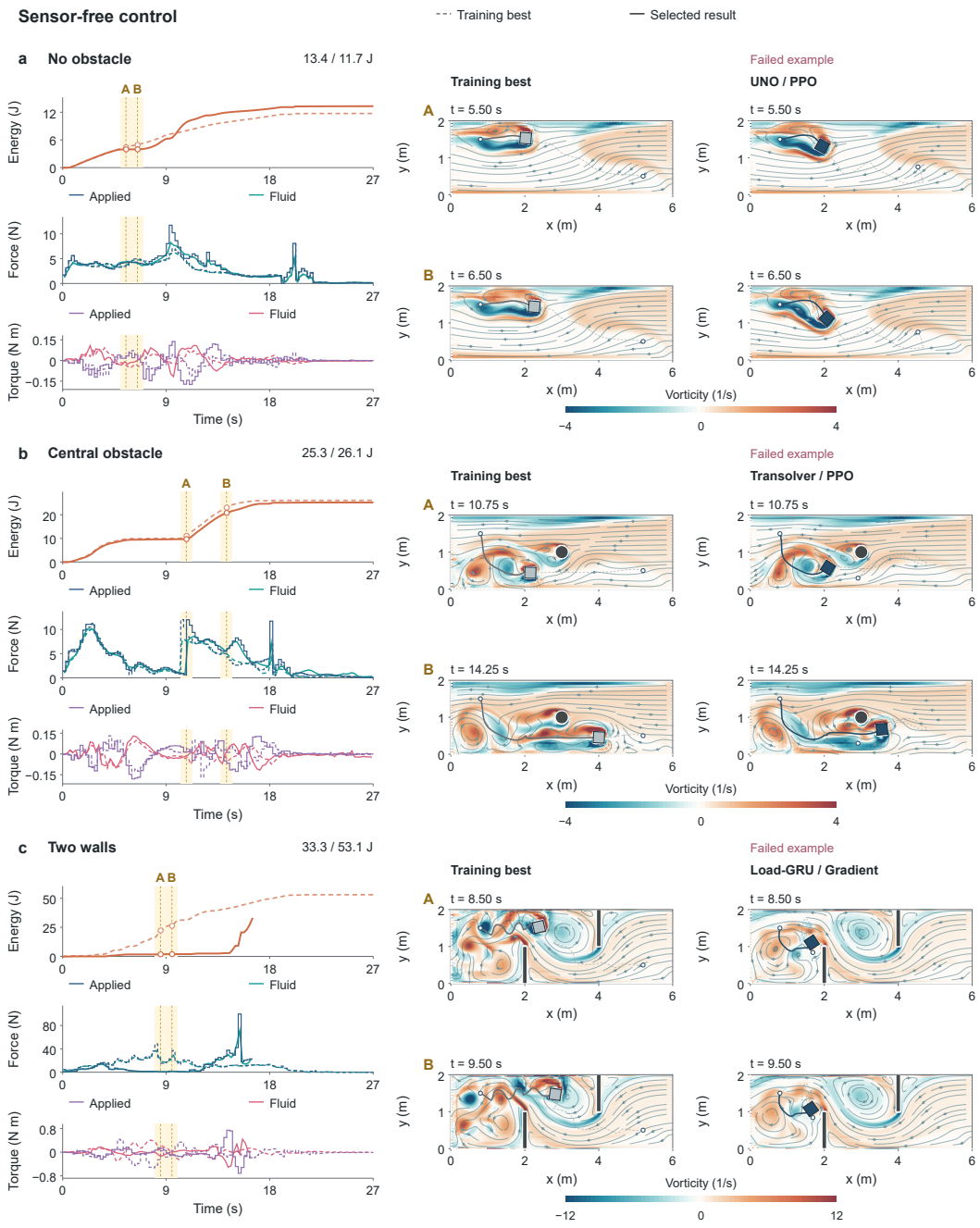

Figure S11: **Motion and work during unsuccessful sensor-free execution.** **a–c**, UNO–PPO, Transolver–PPO and Load-GRU–Gradient in the three geometries. Left, cumulative work, applied/fluid-force magnitudes and signed torques. Dashed curves show the successful training reference; solid curves show the failed sensor-free execution. Gold windows A and B locate the largest local differences in accumulated work using Eq. 5. Right, paired CFD vorticity, streamlines and trajectories at the annotated times. Panels **a,b** show departures from the reference followed by terminal failure. In **c**, the body remains before the first wall and the recorded trajectory ends after 16.50 s of motion. Colours and path conventions follow Fig. S8.

E Training Details and Convergence

Training uses the trajectory partitions in Appendix B. Models are selected on validation data and then held fixed for control evaluation. Figs. S12 and S13 show the training and validation histories.

E.1 Training Predictors for Sensor-free Circular Control

Field inputs combine initial velocity and pressure, signed-distance geometry, query position and time, body velocity, Reynolds number and inlet velocity. A path segment has ten features: start/end coordinates, velocity, duration, start/end times and progress. The path encoder has width 64, two layers and four attention heads, with eight channels passed to the spatial predictor. The sequence contains at most 108 segments. Its loss is

$$\mathcal{L}_{\text{circle}} = \mathcal{L}_{\text{field}} + 0.1\mathcal{L}_{\text{force}} + 0.1\mathcal{L}_{\text{segment}} + 0.02\mathcal{L}_{\text{pressure force}} + 0.1\mathcal{L}_{\text{difference}} + 0.2\mathcal{L}_{\text{path}}. \quad (6)$$

The terms measure normalized squared errors in fields, force, interval work, pressure-derived force, consecutive-frame field differences and mean work over valid path intervals. Spatial losses use fluid-cell masks. Channel scales are $(0.6, 0.6, 0.36)$; force and interval-work scales are 5 N and 0.2 J.

AdamW trains the models for 50 epochs with batch size four, learning rate 10^{-4}, weight decay 10^{-5}, cosine decay, gradient clipping at norm 1 and seed 20260714. FNO, F-FNO, UNO and U-Net initialize their spatial components from single-step predictors; Transolver and Swin initialize randomly. The lowest validation-objective checkpoint is used in both endpoint settings (Fig. S12a,b).

PathCNN and Path Transformer use five-model ensembles per geometry with the data partitions given in Appendix B. The loss combines Smooth-L1 regression of standardized work and a pairwise ranking term weighted by 0.12. AdamW uses learning rate 1.5×10^{-3}, weight decay 2×10^{-4} and batch size 32. Validation loss determines learning-rate reduction, early stopping and checkpoint selection. PathCNN runs for 81–195 epochs and Path Transformer for 78–289 epochs across scenes and ensemble members. Search uses the ensemble mean and variation in predicted work.

E.2 Training Predictors for Sensor-based Square Control

Each training window samples one of the three pressure masks uniformly. Horizons contain four or sixteen steps. Field predictors minimize initial-field reconstruction error plus future-step errors in body state, loads, pressure and fields. All are mean squared errors, with fluid masking for fields. Body scales are $(0.3, 0.3, 0.3, 0.3, 1, 0.5)$ and load scales $(100, 100, 1.2)$; angle residuals are wrapped to $[-\pi, \pi]$. Load predictors retain body, load and pressure terms. RSSM adds posterior observation reconstruction and a per-step KL term weighted by 0.01.

Neural models use AdamW for 50 epochs with learning rate 3×10^{-4}, weight decay 10^{-5}, batch size 32 and norm-10 gradient clipping. Four-second models accumulate gradients from microbatches of 16. The validation score sums errors in the quantities predicted by each model, averaging forecast errors over the horizon; it selects the checkpoint. POD–Kalman fits its reduced basis from up to 512 training snapshot pairs, then trains controlled transition matrices and an offset for 50 epochs at 10^{-4} while holding the basis and observation covariance fixed. The same four-second checkpoints are used in predictor-based and hybrid planning.

E.3 Training Predictors for Sensor-free Square Control

Field predictors receive the initial field, body state, geometry and full control schedule. Sixteen features per interval describe actions, timing, cumulative and integrated impulses, and scene identity. The temporal encoder has width 96, three layers and four heads, supplying eight spatial channels. Outputs comprise fields, loads, body states and interval work. The 240 training trajectories contain 25,920 intervals and 24 examples of each rotation mode.

Loss weights are 1/0.3/0.1 for field/temporal-load/spatial-load errors; 0.5 for body state; 0.1/0.5/0.75 for shape angle/unwrapped angle/angular velocity; and 1/0.2/1.5 for terminal state/shape angle/unwrapped angle. Rigid-body and successive-state consistency have weights 0.2/0.1. Interval work, spatial-head interval work, total work, total-work correction and field differences have weights

0.2/0.05/0.1/0.1/0.05. Field scales are $(0.6, 0.6, 0.5)$, load scales $(10, 10, 0.3)$, and interval/total-work scales 2/20 J. State scales for $(x, y, v_x, v_y, \phi, \omega)$ are $(3, 1, 1, 1, 2\pi, 2)$; the separate angular-velocity and shape-angle losses use 0.5 rad s^{-1} and $\pi/4$.

All six models initialize randomly and train for 50 epochs with batch size four and seed 20260716. AdamW uses learning rate 10^{-4}, doubled for the rotation branch, weight decay 10^{-5}, cosine decay to 10^{-6} and separate norm-1 clipping for rotation and other parameters. The first five epochs train dynamics with true preceding states; field decoding begins in epoch six. Over the next ten epochs, preceding-state inputs shift linearly to model predictions. Validation uses predicted states and selects the lowest-loss checkpoint after field training begins (Fig. S12c,d).

Load-GRU, Load-TCN and Load Transformer predict fluid loads for rigid-body integration. Their objective is $\mathcal{L}_{\mathrm{fluid}} + 0.35\mathcal{L}_{\mathrm{state}} + 0.15\mathcal{L}_{\mathrm{work}}$, using training-set load standardization, scaled one-step state error and Smooth-L1 work loss with transition scale 0.2 J. Training lasts 50 epochs with seed 20260716, AdamW weight decay 2×10^{-4}, cosine decay to 2×10^{-5} and gradient clipping at 2. Validation loss selects the checkpoints (Fig. S12e,f).

E.4 Training Control Generators

PPO, SAC and TD3 each use 200,000 transitions from their associated predictor; Dreamer uses the same budget with RSSM. Training starts from sampled demonstration histories and follows four- or sixteen-step model rollouts. Applicable learners initialize by behavioural cloning. BC, Decision Transformer, diffusion policy, IQL and CQL train for 50 epochs with batch size 256. BC and DT fit the next action; diffusion predicts eight-action sequences. The supervised optimizer is AdamW with learning rate 3×10^{-4} and weight decay 10^{-5}. Validation imitation loss selects BC, DT and diffusion checkpoints; IQL and CQL use their final actors.

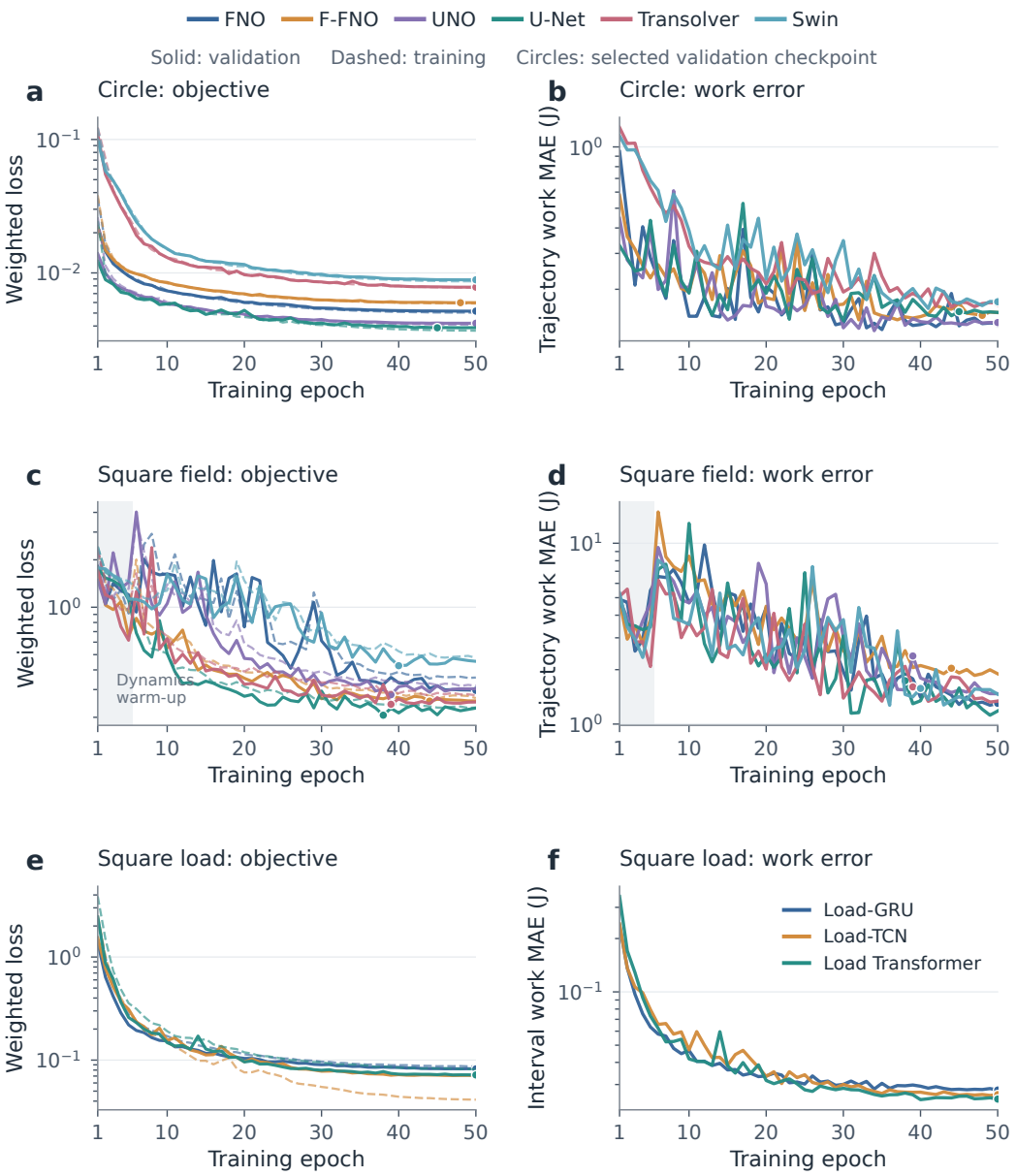

Figure S12: **Training complete-trajectory predictors.** **a,b**, Circular-path field models: the weighted training objective and validation error in total path work. **c,d**, Square-body field models: the corresponding objective and total-work error. Grey shading marks the first five epochs of dynamics-only training; field decoding begins in epoch six. **e,f**, Square-body load models: the weighted objective and validation error in work per control interval. In **a,c,e**, dashed and solid curves denote training and validation losses. Work errors in **b,d,f** are validation MAEs in joules. Colours identify architectures; the upper legend applies to **a–d** and the lower legend to **e,f**. Circles mark the checkpoint selected by the relevant validation objective and its work error at the same epoch. Curves show all 50 training epochs on logarithmic vertical axes. Circular models use 486/60 training/validation trajectories; square models use 240/30.

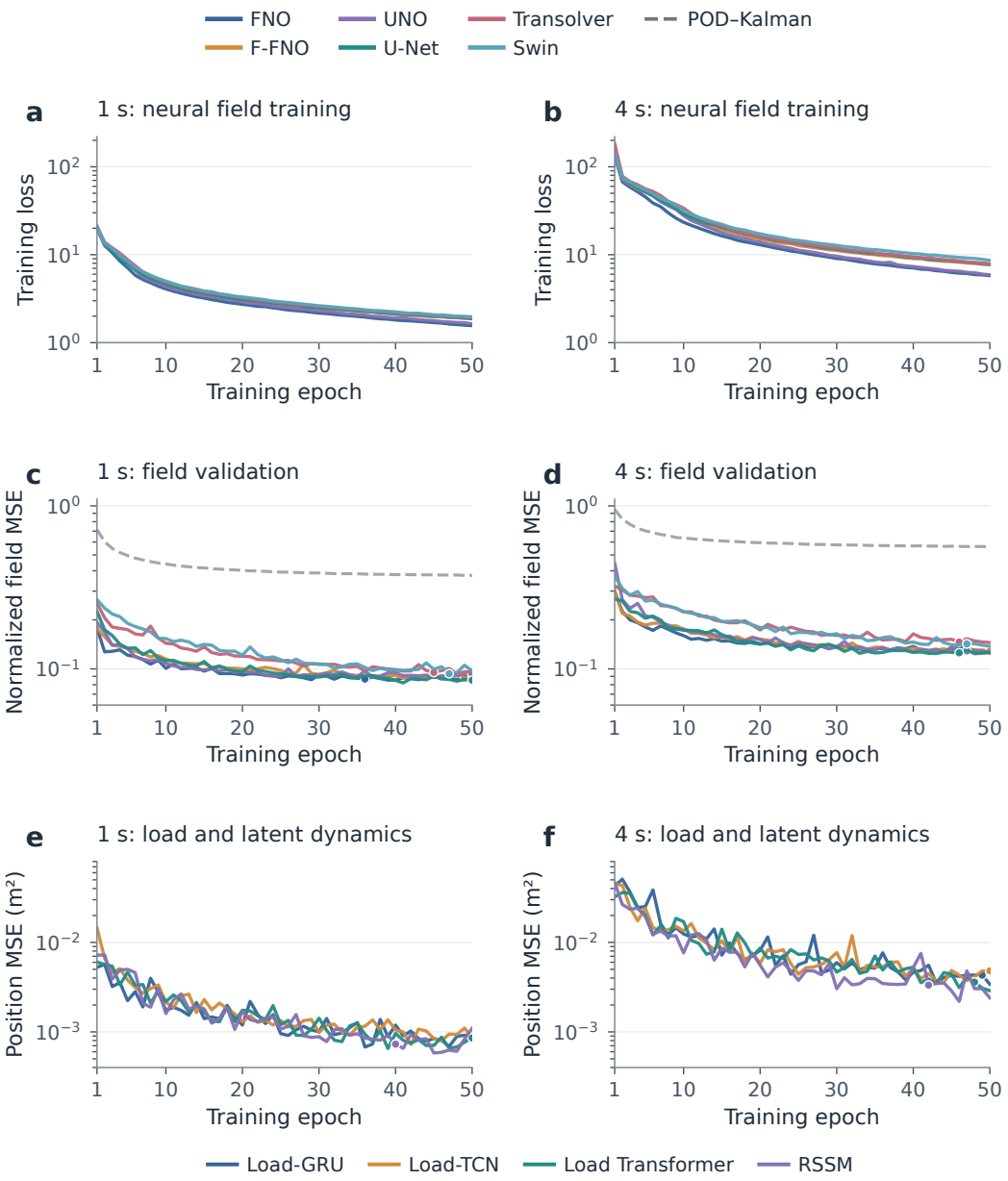

Figure S13: **Training predictors from boundary observations.** Columns show four-step (1 s) and sixteen-step (4 s) prediction. **a,b**, Multi-step training losses of the six neural field models. The objective sums future-step errors over the respective horizon and includes reconstruction of the initial field. **c,d**, Validation MSE of the normalized velocity and pressure channels, averaged over fluid cells and forecast steps. The grey dashed curve shows POD–Kalman after fitting its reduced basis and optimizing its transition parameters. **e,f**, Validation position MSE, averaged over the two position components and prediction steps, for the three load predictors and RSSM. The upper and lower legends identify the models in **a–d** and **e,f**, respectively. Circles mark the epochs selected by the full validation score for each neural model. Every curve contains 50 training epochs; vertical axes are logarithmic and use matching limits across horizons within each row. Training and validation use 210 and 45 trajectories with mixed sensor masks.

F SUPPLEMENTARY METHOD DETAILS

We use n for the decision index, $t_n = n\Delta t$ for elapsed motion time and s_n for body state, matching Method. Predictors remain fixed during search. The optimization variable is a path, a complete action sequence or a sequence of corrections. Algorithm 1 summarizes hybrid planning.

Algorithm 1 Hybrid control with fixed four-second predictors

```
Initialize CFD, observation history and a zero-correction plan.
for n = 0, ..., N − 1 do
    Update h_n, estimate the fluid load and set H_n = min(16, N − n).
    Shift the previous plan; when applicable, construct R^π for MPPI.
    Search candidates with both forecasts, recomputing base actions through A.
    Evaluate the candidate and R = 0 with Eq. 9.
    Check feasibility in both forecasts, relative progress and the improvement margin.
    Apply the first accepted correction, or zero correction, for one CFD interval.
end for
Return the executed motion, completion outcome and positive work.
```

F.1 DETAILS ON SENSOR-FREE CONTROL OF CIRCULAR-BODY MOTION

The generator adjusts control points while fixing the task endpoints. Each path is sampled at the prescribed speed and checked for length, duration and clearance. Field models estimate loads and work along the path; scalar models estimate total work. Endpoint transfer changes the endpoints before search and uses the same trained predictor.

F.2 DETAILS ON PREDICTOR-BASED PLANNING OF SQUARE-BODY MOTION

At decision n, the planner optimizes $R = (r_n, \dots, r_{n+H_n-1})$ with $H_n = \min(H, N-n)$ and $H \in \{4, 16\}$. The unexecuted previous plan initializes search, repeating its last correction to fill the window. Each forecast recomputes u_{FB} from its predicted state, adds the candidate correction and enforces the physical action bounds. The zero-correction plan is evaluated over the same window.

For a feasible candidate, the predictive cost is

$$C(R) = \sum_{j=0}^{H_n-1} \left[\frac{\widehat{W}_j}{10} + 0.05 \|\widehat{\mathbf{x}}_{j+1} - \mathbf{x}_B\| \right] + 5 \|\widehat{\mathbf{x}}_{H_n} - g(\widehat{s}_{H_n}, \mathcal{G})\|^2, \tag{7}$$

where $\widehat{W}_j$ is interval work and g returns the next guidance waypoint. Feasible costs map to $C/(1+C)$. Terminal violations, collisions and invalid numerical predictions occupy successively higher score classes. The selected candidate is accepted if feasible and no worse than the zero-correction plan; otherwise $r_n = 0$. Direct policies produce r_n from the observation history without this candidate-ranking step. DOB applies u_{FB} with zero correction.

F.3 DETAILS ON HYBRID PLANNING OF SQUARE-BODY MOTION

The hybrid implementation combines learned and constant-load forecasts with a relative-progress objective, arrival and stopping terms, and an acceptance margin. It uses the fixed four-second learned predictor. For each completed history interval $j < n$, the disturbance observer obtains an interval-average load sample from body motion and the applied action:

$$\widehat{\boldsymbol{\ell}}_j = \left(m \frac{\mathbf{v}_{j+1} - \mathbf{v}_j}{\Delta t} - \mathbf{F}_{\mathrm{act},j}, \quad I \frac{\omega_{j+1} - \omega_j}{\Delta t} - \tau_{\mathrm{act},j} \right). \tag{8}$$

The current estimate is a normalized weighted mean, with weights decaying by 0.8 per older valid interval. The base action combines velocity tracking, orientation regulation and load compensation. Corrections are parameterized along/across the guidance direction and in torque, with bounds $[-0.35, 3]$ N, ± 1.25 N and ± 0.02 N m. Their amplitude decreases linearly as remaining guided distance falls from 0.6 to 0.1 m. The combined force is projected onto the 100 N norm bound and torque is clipped to ± 1.2 N m. This action map $\mathcal{A}$ is used in both prediction and execution.

Each candidate and the zero-correction reference are rolled forward with both models from the current state. The learned forecast updates fluid loads; F_{RB} holds their current estimates constant while integrating Eq. 4. Both recompute the base action along their predicted motion. Let d_0 be current remaining guided distance, and $(\widehat{E}_{0,b}, \widehat{d}_{0,b})$ and $(\widehat{E}_b, \widehat{d}_b)$ be reference and candidate work/distance under model b. The score is

$$J(R) = \frac{1}{2} \sum_{b \in \{\mathrm{learned}, \mathrm{RB}\}} \left[\widehat{E}_b + \lambda_b (\widehat{d}_b - \widehat{d}_{0,b}) + P_{\mathrm{deadline},b} + P_{\mathrm{stop},b} \right], \tag{9}$$

with $\lambda_b = \mathrm{clip}_{[1,100]}\{\widehat{E}_{0,b} / \max(d_0 - \widehat{d}_{0,b}, 0.05)\}$. For forecast end time $t_f = t_n + H_n \Delta t$, $P_{\mathrm{deadline}} = 200[\widehat{d} - 0.05 - 0.34[24 - t_f]_+]_+^2$. Within 0.3 m of the target, $P_{\mathrm{stop}} = 20\widehat{d}^2 + 3\|\widehat{\mathbf{v}}\|^2 + 3\widehat{\bar{\phi}}^2 + 0.1\widehat{\omega}^2$, where $\bar{\phi} = \frac{1}{4}\,\mathrm{atan2}(\sin 4\phi, \cos 4\phi)$; it is zero outside that neighbourhood. The travelling reference speed uses remaining guided distance and a 21 s arrival reference, clipped to 0.18–0.34 m s^{-1}.

Acceptance requires candidate and reference feasibility in both forecasts, candidate remaining distance no more than 0.02 m above the reference in either forecast, and score improvement greater than $0.005 \max(1, |J(0)|)$. Otherwise the controller applies the base action. Shooting, CEM, MPPI and gradient search use this score and these checks. For policy-guided MPPI, alternating policy proposals and learned predictions produces R^π. The sampling mean starts at the average of R^π and the shifted previous plan, initially zero. Each iteration includes this mean, R^π, feedback alone and perturbed sequences. The final comparison includes the updated mean, best feasible sample and feedback alone. Learned, rigid-body and policy rollouts, together with gradient computations, share the decision budget.

F.4 Details on Sensor-free Control of Square-body Motion

Search starts from a training control sequence and adjusts smooth corrections defined by sixteen interpolation points across 108 intervals. The predictor evaluates 200 final candidates from the prescribed initial state. We select three candidates per model–generator–geometry group, subject to minimum deviation from training controls and pairwise separation. Predicted-feasible candidates with lower work take priority; remaining places are filled by the lowest objective values, combining work and constraint losses. The selected sequences are executed in OpenFOAM without observations during motion.

F.5 Details on CFD integration

Square-body work integrates the positive part of combined translational and rotational actuator power. With a constant action in each interval, the recorded endpoint rule is

$$E = \sum_{n=0}^{N-1} \frac{\Delta t}{2} \left([\mathbf{F}_{\mathrm{act},n} \cdot \mathbf{v}_n + \tau_{\mathrm{act},n} \omega_n]_+ + [\mathbf{F}_{\mathrm{act},n} \cdot \mathbf{v}_{n+1} + \tau_{\mathrm{act},n} \omega_{n+1}]_+ \right). \tag{10}$$

Both endpoints use the action applied during interval n. Circular work follows Eq. 1. Early CFD termination counts as failure. All submitted controls enter completion and completion-with-improvement counts; work comparisons use completed motions.