
Learning to Control Free-Form Soft Swimmers (Supplementary Materials)

Changyu Hu¹ Yanke Qu¹ Qiuhan Yang^{2,1} Xiaoyu Xiong^{2,1}
Kui Wu³ Wei Li^{3,4} Tao Du^{1,2}

¹ Tsinghua University ² Shanghai Qi Zhi Institute

³ LIGHTSPEED ⁴ Shanghai Jiao Tong University

This supplementary document provides algorithmic implementations, simulation parameters, training configurations, and additional experimental results to support the main paper, "Learning to Control Free-Form Soft Swimmers." It includes mathematical formulations, specific parameter values, and figures to aid reproducibility.

The supplement is organized as follows: Sec. 1 details swimmer modeling and Sec. 2 details fluid-solid coupling; Sec. 3 describes the experiment setup and details about baselines; Sec. 4 provides additional experiment results; Sec. 5 gives detailed results of our ablation studies. This document aims to enhance transparency and reproducibility of the study.

1 Swimmer Modeling Details

This section provides mathematical and implementation details for the kinematic control and dynamic correction of free-form soft swimmers.

Weights The radial basis function that we use as the LBS weight on each vertex is defined as

$$\tilde{w}_i(\|\mathbf{X} - \mathbf{p}_i\|_{\text{geo}}) = \exp\left(\frac{-\|\mathbf{X} - \mathbf{p}_i\|_{\text{geo}}^2}{\sigma}\right), \quad (1)$$

where σ is a constant related to the extend of model. After computing the weights of all LBS control points, the weights are normalized on each vertex:

$$w_i = \frac{\tilde{w}_i}{\sum_{j=1}^m w_j}. \quad (2)$$

This ensures non-negativity and the partition of unity property, which are critical for maintaining the physical plausibility and numerical stability of the soft swimmer model.

Dynamic correction To implement the dynamic correction projection, we leverage the Newton's method to solve this energy minimization problem,

$$\mathbf{u}_d^* = \arg \min_{\mathbf{u}_d} \Psi(\mathbf{X} + \mathbf{u}_k + \mathbf{u}_d) + \frac{1}{2}k\|\mathbf{u}_d\|_2^2, \quad (3)$$

due to its high stability. To do so, we treat the minimization problem as solving the quasi-static state of an elastic body, deformed with a spring-like potential term $\frac{1}{2}k\|\mathbf{u}_d\|_2^2$.

2 Fluid Solid Coupling Details

Our simulation implementation is composed of three main components: solid, fluid, and coupling. The solid component is detailed in Sec. 1. For the fluid component, we follow the HOME-LBM (Li et al., 2023) algorithm. This section provides additional details about our weak coupling implementation.

2.1 Elastic-Fluid Coupling

Overview We present a weak two-way coupling scheme between the incremental-potential elastics and the lattice-Boltzmann fluids. We prefer this scheme to the alternative strong coupling scheme because it better utilizes the state-of-the-art GPU-accelerated solvers dedicated to elastics (Chen et al., 2024) and fluids (Li et al., 2023), respectively. Our time stepping scheme contains three steps: detect elastic-fluid intersection, update fluids, and update elastics.

Elastic-fluid intersection At the beginning of each time step, elastic-fluid intersections are detected for the following fluid and elastic updates. A lattice cell is defined as *cut cell* if it intersects with any codimension-1 boundary element (line segments in 2D and triangles in 3D) of the mesh describing the elastic swimmer. Then, *cut-cell node* is the node belonging to a cut cell. We further define *lattice link* $l_i(\mathbf{x})$ as the line segment extending from a lattice node \mathbf{x} along the i -th direction to a neighboring node (see Fig. 1).

The *elastic-fluid intersection* is defined between a codimension-1 boundary element and a lattice link within a cut cell. Given the large number of FEM elements and lattice links within the grid, fast and GPU-based intersection detection is crucial to building a full GPU simulator in our environment. We adapt the approach from Lyu et al. (2023), which parallelizes over boundary elements to check elastic-fluid intersections. Building on this, we further parallelize the computation by considering both the boundary elements and the 26 lattice directions in 3D (8 in 2D), increasing the granularity of parallelism. This approach leads to a decent performance improvement, achieving a speedup of $2\times$ in elastic-fluid intersection detection compared with previous methods.

Updating fluids Once we collect all intersections, we couple elastics and fluids by exchanging momenta at these intersections between the related boundary element and cut cell. When updating the fluid, we treat the elastic-fluid interface as a kinematically moving no-slip boundary condition and incorporate it into the HOME-LBM fluid solver as described in Li et al. (2023).

Updating elastics One side product from HOME-LBM fluid update is momentum exchange at each elastic-fluid intersection point, which we adopt for fluid-elastics coupling. Consider a boundary element $\triangle ABC$ in 3D intersected with a lattice link at a point D (Fig. 1). Let \mathbf{p}^D be the momentum exchange at D computed from the previous HOME-LBM fluid update. We distribute \mathbf{p}^D to the triangle vertices with λ , the barycentric coordinates of D in $\triangle ABC$:

$$\mathbf{p}^* = \lambda^* \mathbf{p}^D, \quad (4)$$

where \star can be A , B , or C . In practice, one can compute λ efficiently via the area-based method (ratios of sub-triangle areas). The same method applies to 2D cases, where we replace triangles with line elements. We accumulate the momenta at each vertex of all boundary elements of the mesh and apply them as explicit external forces in the elastic solver, which concludes our elastics update.

Leakage-free coupling Fig. 2 tests the leakage-free property of our elastic-fluid coupling scheme. We build a fluid domain on a grid resolution of $300 \times 200 \times 200$ and immerse a square piece of cloth with fixed corners. A fluid inlet on the top face of the domain generates a vertical inlet flow targeting the center of the cloth. A side view shows that the fluid flow does not pass through the cloth but splashes above it, which confirms empirically that our coupling scheme is leakage-free.

Remarks Our elastic-fluid coupling fills the gap for kinetic-based fluid and deformable body coupling solver in soft-body swimmer learning, efficiently spreading the fluid and solid intersection’s momentum to mesh vertices for two-way coupling and providing a more efficient parallel elastic-fluid intersection implementation than previous works (Lyu et al., 2023).

- First, the above scheme naturally handles coupling with elastics in codimension-0, codimension-1, or a conforming mixture of the two in a unified manner. Indeed, intersection detection does not need to distinguish a stand-alone boundary triangle from a boundary triangle on a tetrahedron.

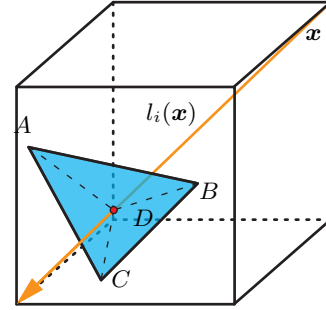


Figure 1: Lattice link $l_i(\mathbf{x})$ intersecting with a surface triangle.

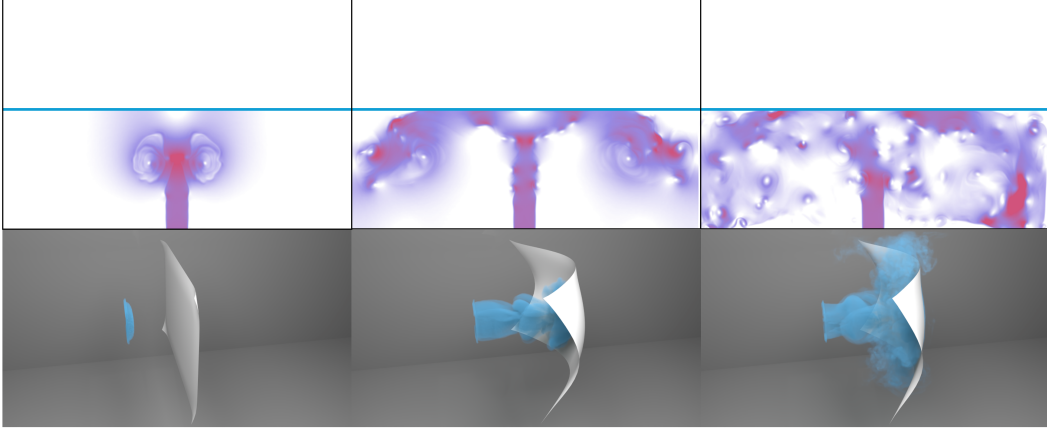


Figure 2: 2D and 3D Leakage Test. Top: A blue fixed line is placed at the center of the scene, causing the inlet to spread apart as it approaches the line. Bottom: The inlet moves toward a piece of deformable cloth with its four corners fixed. The three frames (from left to right) show the initial, middle, and final stages of the interaction.

- Second, due to the mesoscopic boundary treatment with the elastic-fluid intersection, our solver can support *leakage-free* coupling, also shown in our empirical test (Fig. 2), a crucial property without which many high-performing swimming skills would be impossible, such as jellyfish.

Compared to the simplified fluid model used in Ma et al. (2021); Min et al. (2019), our simulation demonstrates better capability in capturing intricate fluid dynamics characteristics, thereby enabling the acquisition of advanced swimming patterns governed by hydrodynamic principles (see Sec. 5). In contrast to Material Point Method (MPM) simulations (Wang et al., 2023b,a) that employ a unified particle representation for both fluid and solid phases - an approach prone to material distortion and tearing artifacts (see supplemental videos) - our methodology exhibits enhanced numerical stability and physical plausibility.

3 Experimental Setup and Baseline Implementations

3.1 Simulation Setup

The simulation parameters are fixed for each task when training different swimmers. In addition, the material parameters of the fluid and soft body remain the same in every task. Only the domain resolution of fluid is different to suit each task objective. The specific parameters of simulation setup we use is shown in Tbl. 1.

3.2 RL Training Setup

The swimming strategy is learned using Soft Actor-Critic (SAC) (Haarnoja et al., 2018). The training hyperparameters are shown in Tbl. 2. The two penalty coefficients we use in reward during training are $\lambda_{smooth} = 0.02$ and $\lambda_{reg} = 0.01$. The action vector can be divided into translation part and rotation part. We set different ranges for the two parts of action, which are 0.1 for translation and 0.4 for rotation. Both the penalty coefficients and the action ranges are fixed in all of our experiments.

3.3 Morphology Collection

When selecting soft swimmers, we primarily consider different structural features and potential swimming modes, aiming to demonstrate the adaptability of our approach to various morphologies.

The numbers of LBS control points that we use in swimmers depend mainly on the morphology. Due to the selection of geodesic distance in LBS process, the LBS control points tend to distribute at

Table 1: Simulation parameters

Parameter			Value
Fluid	Domain Resolution (Forward)	(n_x, n_y, n_z)	(512, 128, 128)
	Domain Resolution (Navigation)	(n_x, n_y, n_z)	(300, 300, 300)
	Grid Spacing	dx	0.02
	Time Step	dt	0.005
	Density	ρ_f	500
	Kinematic Viscosity	ν	0.02
Elastic Body	Density	ρ_s	1000
	Shear Modulus	μ	3.57×10^4
	Lamé Constants	λ	1.67×10^5
	Substeps	n_s	4
	Acceleration Ratio (of VBD)	ω	0.65
	Iterations (of VBD)	n_{itr}	40

Table 2: Training hyperparameters.

Hyperparameter	Value
Learning Rate (actor)	3×10^{-5}
Learning Rate (critic)	3×10^{-4}
Discount Factor	0.99
Replay Buffer Size	10^6
Batch Size	1024
Update Iteration	32
Optimizer	Adam
Activation Function	ReLU
Hidden Dimension	1024
Hidden Layers	4
Entropy Target	$- \mathcal{A} $
Target Smoothing Coefficient	0.01
Policy Update Interval	1
Target Update Interval	1

the tips of the protruding parts of model (such as octopus tentacles or turtle limbs). This allows us to express a swimmer’s primary deformation patterns using the minimal number of control points. The numbers we employ can serve as a reference value. In practical applications, according to the results of our experiment, provided that the number of control points suffices to capture the primary deformation modes of the swimmer, increasing or reducing a few points will not exert a decisive influence on the final swimming performance (see Sec. 5 for more details). The specific parameters of swimmers are shown in Tbl. 3.

3.4 Baseline Implementation Details

To ensure experimental fairness, all comparative studies presented in Table 1 in the main paper are conducted under strictly controlled experimental conditions: 1) Identical reinforcement learning training pipelines and simulation environments were maintained across all test groups; 2) The control architectures represented the sole variable being tested; 3) All neural network inputs employed geometrically consistent sampling points derived from a unified geodesic farthest-point sampling strategy. This standardization procedure eliminates confounding factors by ensuring identical observation spaces and input feature representations across all experimental configurations. The systematic control of variables provides statistically meaningful comparisons between different control paradigms.

Domain-expert controller The actuation distribution of domain-expert designed models are shown in Fig. 3. Details about each model are listed below:

Table 3: Model parameters

Model Name	Vertices	Number of	
		Tetrahedrons	LBS Points
Clownfish	1009	3543	6
Eel	1099	5225	5
Octopus	500	1389	7
Leaf	1178	3830	8
Turtle	1282	3855	7
Jellyfish	691	2310	4
Torus	605	1491	4
Eight	1111	3328	7
Spiral	652	1653	4
Trumpet	1148	3247	9
Tube	1504	5027	4
Enneper	473	1294	6

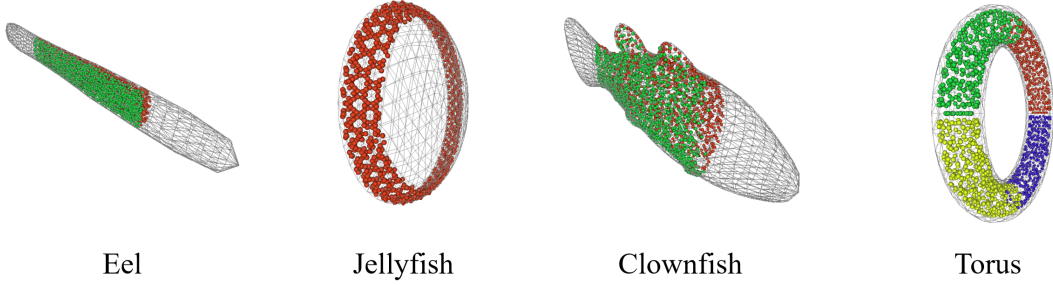


Figure 3: Domain-expert actuation designs. The colored point clouds indicate different muscle groups. The hollowed-out portion represents the passive region that does not apply actuation.

- The clownfish model (length: 1m) features axial muscle activation within the 0.20.7m posterior region. Following Lin et al. (2019), we implemented bilateral antagonistic muscle control through a single action parameter. However, dorsoventral asymmetry caused ineffective vertical control during initial trials. Our solution adopted quadruple muscular segmentation (torus-like) with separate horizontal / vertical oscillation controllers, achieving better performance.
- For the eel swimmer we apply the same actuation distribution as they have similar topology and swimming strategy. But this actuation transfer proves ineffective. Through optimization of both spatial distribution and dimensional parameters of the muscles, it achieves performance comparable to those obtained through our method. Detailed visual evidence supporting these findings is available in supplemental videos.
- Following the design in Lin et al. (2019), the jellyfish is a hemispherical shell, with the thickness of one-third of head region increased to form a plane. The non-thickened part (the width is 0.13m) is the active region with radial actuation, with its intensity increasing with the radial distance.
- The torus model is divided into four quadrants, with the same radial actuation directions as the jellyfish, following the practice of Du et al. (2021).

Clustering-based controller The process of clustering-based controllers follows the practice in Wang et al. (2023a,b): (1) converts mesh to point cloud, (2) performs K-means clustering with user-specified body groups, and (3) applies PCA on each cluster to determine muscle orientation. To align with experimental settings, we initialize the K-means clusters at the same positions as our method’s LBS control points, ensuring the number of clusters matches the number of LBS control points used in our approach.

4 Additional Experimental Results and Analysis

In addition to the 3D forward swimming task described in Section 5 of the main text, we evaluate performance on three additional locomotion tasks: 3D navigation, 2D forward swimming, and 2D navigation.

Our fluid-structure interaction (FSI) simulation framework natively supports 2D scenario modeling. The proposed control methodology can be seamlessly extended to 2D cases with minimal adaptation cost. For a 2D system with m control points, the control dimension reduces to \mathbb{R}^{3m} —comprising $2m$ translational and m rotational degrees of freedom. The deformation field calculation follows the same process discussed in Sec.3 of the main paper.

4.1 3D Navigation Task

In this task, the swimmer reaches arbitrary generated target placed at a random angular deviation from the forward direction $(l, 0, 0)$.

$$\mathbf{x}_{\text{target}} = \mathbf{x}_{\text{center}} + (l \cos \theta, l \sin \theta \cos \phi, l \sin \theta \sin \phi), \theta \in [0, 2/9\pi], \phi \in [0, 2\pi]$$

where $\mathbf{x}_{\text{target}}$ is the target point, $\mathbf{x}_{\text{center}}$ is the center of the swimmer, l is a fix length, θ and ϕ are randomly generated angle. The rewards are similar to the forward swimming task.

We evaluate on the jellyfish model using our proposed method alongside two baseline algorithms, with comparative results illustrated in Fig. 4. Our methodology demonstrates the capability to generate deformations enabling directional steering in jellyfish locomotion. In contrast, clustering-based methods only produce oscillatory movements without net displacement, similar to observations from forward swimming tasks. Domain-expert actuation designs restrict to uniform tangential contractions proved inadequate for generating asymmetric deformations required for directional control—a limitation necessitating additional muscle categorization for mission-specific adaptations.

4.2 2D Forward Swimming Task

We conduct the task on a 2D fish-like swimmer to move as fast as possible to the forward direction. The task setting and reward are similar to the 3D case.

After training, the swimmer learns to oscillate its tail, generating symmetric vortex structures that propel it forward. The quantitative result can be seen in Fig. 5.

4.3 2D Navigation Task

In 2D navigation Task, the swimmer continuously reaches sequentially generated targets, each placed at a random angular deviation from the previous direction after reaching the current one, which is more complicated compared to the 3D navigation task. We combine trajectory segmentation (e.g. Liu et al., 2022) and velocity alignment (e.g. Wang et al., 2023a) where target positions are dynamically generated within an angular sector $\Theta = \theta \pm \delta\theta$, centered around the current velocity vector \vec{v} of the swimmer. Result can be seen in Fig. 5.

4.4 2D Flow Resistance Task

This task evaluates the agent’s ability to maintain stability against a constant environmental disturbance. Our task formulation is elegantly adapted for this scenario: the target position is simply set to the swimmer’s initial location. The reward function encourages the agent to minimize its deviation from this point. To improve training efficiency, an episode is terminated early if the swimmer drifts beyond a predefined threshold distance from the origin. Furthermore, a small, constant "survival bonus" is added to the reward at each timestep, incentivizing the agent to remain active and stable for as long as possible. As shown in Table 4, the learned policy successfully generates continuous counter-thrust to actively resist the flow, significantly outperforming baseline strategies. The high deviation of the 'No Action' baseline confirms that a purely passive body is simply washed downstream by the current. Similarly, the poor performance of the 'Random Policy' baseline demonstrates that undirected actuations are also ineffective, highlighting that the trained policy learns to generate not just active, but coordinated and purposeful counter-thrust.

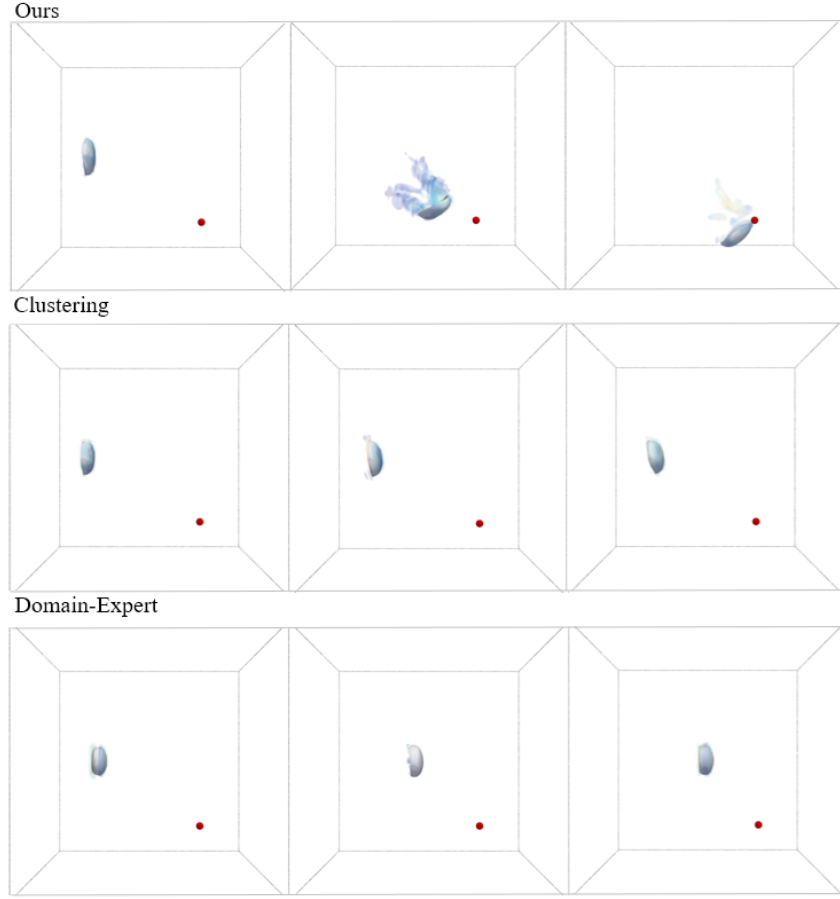


Figure 4: Navigation task for jellyfish model. Gray wireframes denote the boundaries of the fluid domain, while red spheres indicate the positions of target points during the testing phase.

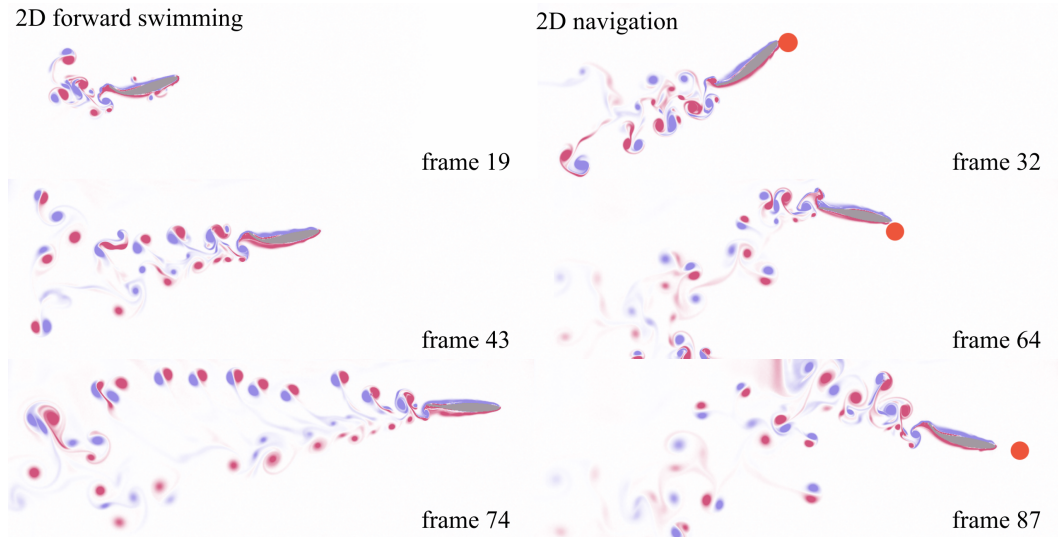


Figure 5: Forward swimming and navigation tasks for 2D fish-like swimmer. The gray object represents the swimmer, with a red circle indicating the target position. The wake behind the fish illustrates the flow dynamics, where the color of the wake signifies the intensity and orientation of the vortex field generated during the fish's motion towards the target.

Table 4: Flow resistance performance in 2D. The trained policy actively resists the flow to minimize positional deviation.

Policy	Average Deviation (m)
Trained	0.030
No Action	0.191
Random Policy	0.226

4.5 Training Stability Analysis

To validate the training stability and reproducibility of our framework, we conducted an in-depth analysis of the learning process. As reinforcement learning performance can be sensitive to random initialization, we performed multiple independent training runs for the 2D fish morphology on the forward swimming task (see Fig. 5). Each run used a different random seed but shared identical hyperparameters.

Fig. 6 shows the learning curves for six separate training runs. Each curve represents the episode reward, smoothed using an exponential moving average with a smoothing factor of 0.9. The solid blue line indicates the mean reward across all runs, and the shaded area represents one standard deviation.

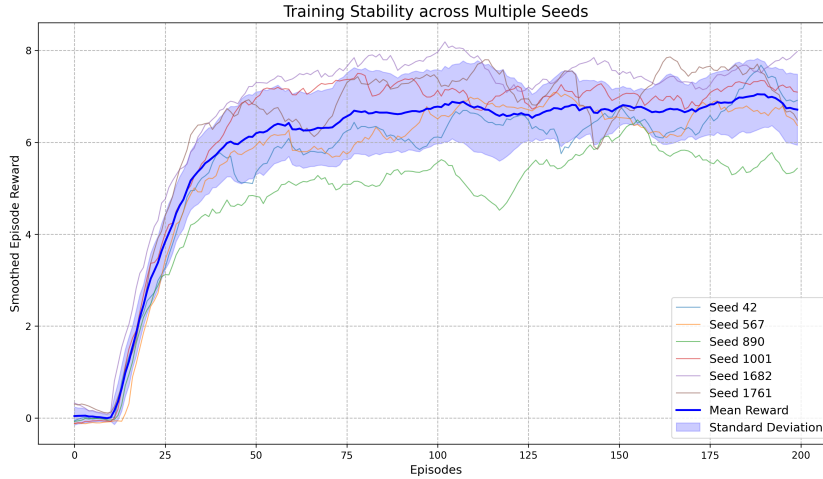


Figure 6: Training stability across five random seeds for the 2d fish swimmer. Individual runs (thin lines) show consistent learning progress, and the low variance (shaded area) indicates that all runs converge to a similar performance level.

As shown in the figure, all training runs exhibit a stable and monotonic increase in performance, rapidly learning an effective swimming gait within the first 50 episodes and converging to a high-reward policy. The low variance between runs confirms that our method is robust to initialization and reliably finds high-quality solutions. This stability is crucial for the practical application of our framework and for fair comparison against other methods.

4.6 Energy Cost Formulation

In Section 6.4 of the main paper, we analyze the energetic efficiency of learned gaits by penalizing an energy cost term E . We define the energy cost based on the positive mechanical work done by the swimmer on the surrounding fluid. This metric represents the energy the swimmer actively expends to generate propulsion.

Our fluid-solid interaction simulator directly computes the hydrodynamic forces, $\mathbf{f}_{\text{fluid}}$, exerted by the fluid onto each boundary node of the swimmer. The instantaneous power $P(t)$ that the swimmer exerts *on the fluid* is the negative of the work rate of these forces. It is calculated by summing over all

boundary nodes i :

$$P(t) = - \sum_i (\mathbf{f}_{\text{fluid}})_i \cdot \mathbf{v}_i \quad (5)$$

where \mathbf{v}_i is the velocity of the boundary node i .

This power can be positive, representing energy the swimmer expends to push the fluid, or negative, which occurs when the swimmer extracts energy from the fluid’s motion (e.g., through wake capture or beneficial pressure gradients). To measure the actual energetic cost, we must only account for the energy the swimmer actively outputs. Therefore, we integrate only the positive component of this power over time.

The total energy cost E for an episode is the cumulative sum of the positive work done over all discrete time steps Δt :

$$E = \sum_t \max(0, P(t) \cdot \Delta t) \quad (6)$$

This metric quantifies the energy the swimmer actively transfers to the fluid to generate motion and provides a direct, physically-grounded objective for learning energy-efficient gaits.

5 Ablation Study

This section reports the details of the four ablation studies mentioned in Section 6.4. of the main text, including the selection of geodesic distance in LBS process, the effect of control point count on motion, the choice of LBM for fluid simulation, and momentum conservation enabled by internal actuators.

Geodesic Distance Our framework’s geodesic sampling strategy proves critical for capturing geometry-aware deformation modes. As illustrated in Fig. 7, geodesic control points on the octopus model naturally cluster near tentacle tips. Activating a single control point induces localized tentacle bending, whereas Euclidean sampling scatters points across multiple tentacles, causing unintended coupled deformations.

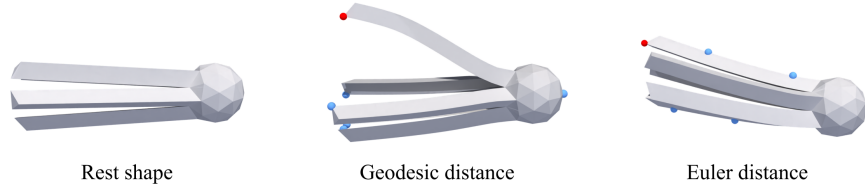


Figure 7: Comparison of geodesic distance and Euler distance in sampling and LBS weights computation. The activated point is red while the other control points are blue.

Control Point Count The number of control points affects both the magnitude and the complexity of the motion. We examined this effect on the eel and turtle models, which vary in morphological structure (Fig. 8). With fewer control points, both swimmers tend to exhibit similar undulatory motion. As the number of control points increases, the turtle begins to utilize its front and hind limbs to paddle through the water, thereby enhancing its propulsion. For simple shapes like the eel, control point count has limited impact on overall strategy but may affect motion amplitude due to overlapping influences among control points.

LBM vs. Simplified Fluids Some existing learning environments (Ma et al., 2021; Min et al., 2019) adopt a simplified fluid model that computes fluid force via an analytical function defined by the relative velocity and the angle of attack at the solid-fluid interface, which is extremely fast to compute, benefiting many sample-intensive deep RL algorithms. However, this approach misses key fluid properties like incompressibility (Min et al., 2019) critical for modeling high-performance swimming skills such as in jellyfish (Bale et al., 2014). We demonstrate this issue by simulating a jellyfish with regular pulsation with simplified fluid, which barely generates any momentum to advance it. In contrast, the same jellyfish in our simulator exhibits plausible motion (Fig. 9).

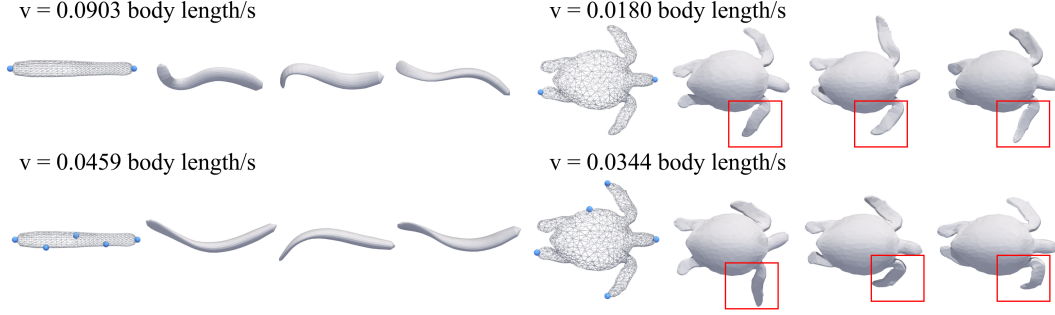


Figure 8: Optimized motion patterns and average swimming speed with different numbers of control points. While the eel shows similar behavior, the turtle begins paddling its front limbs (highlighted with red boxes) when increasing points from 2 to 5.



Figure 9: Comparison of the darting jellyfish between our simulator and simplified fluid methods. Up: our method with flow field visualization; down: simplified fluid methods.

We then validate that our simulator simulates physically plausible solid-fluid interaction by reproducing the *drafting-kissing-tumbling* phenomenon mentioned in Carlson et al. (2004). In our test, two spheres immersed in a fluid domain with a grid resolution of $128 \times 128 \times 256$ fall under gravity from close but different heights. The solid-fluid interaction draws two spheres to tumble, eventually sending the top sphere to the bottom first (Fig. 10). Our result is qualitatively similar to Fig. 5 of Carlson et al. (2004).

Momentum Conservation To validate physical plausibility, we ablate fluid interactions and test the trained policies in vacuum environments. The resulting swimmers exhibit near-zero systemic velocity ($< 10^{-5} m \cdot s^{-1}$), confirming propulsion arises solely from fluid-structure coupling rather than injecting non-physical momentum.

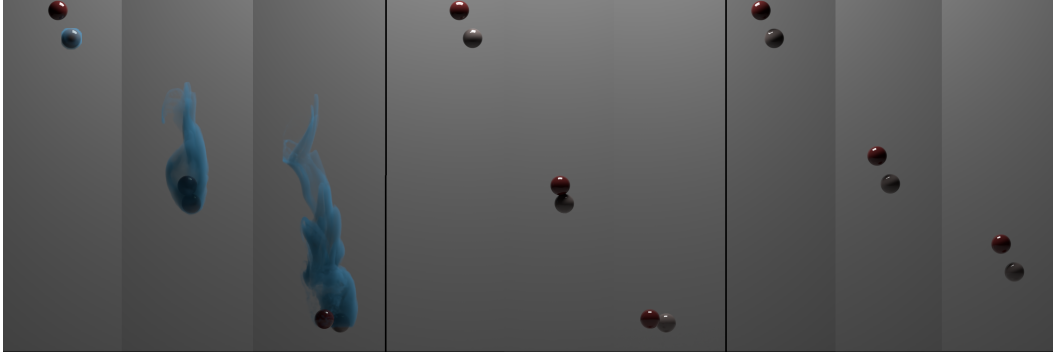


Figure 10: Tumbling. Left: our method with flow field visualization; middle: our method visualized with solid objects only; right: simplified fluid methods. Each part demonstrates three typical frames in the motion.

References

- Rahul Bale, Max Hao, Amneet Pal Singh Bhalla, and Neelesh A. Patankar. 2014. Energy efficiency and allometry of movement of swimming and flying animals. *Proceedings of the National Academy of Sciences* 111, 21 (2014), 7517–7521.
- Mark Carlson, Peter J. Mucha, and Greg Turk. 2004. Rigid fluid: animating the interplay between rigid bodies and fluid. *ACM Trans. Graph.* 23, 3 (Aug. 2004), 377–384.
- Anka He Chen, Ziheng Liu, Yin Yang, and Cem Yuksel. 2024. Vertex Block Descent. *ACM Trans. Graph.* 43, 4, Article 116 (July 2024), 16 pages.
- Tao Du, Kui Wu, Pingchuan Ma, Sebastien Wah, Andrew Spielberg, Daniela Rus, and Wojciech Matusik. 2021. DiffPD: Differentiable Projective Dynamics. *ACM Trans. Graph.* 41, 2, Article 13 (nov 2021), 21 pages.
- Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018 (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer G. Dy and Andreas Krause (Eds.). PMLR, 1856–1865. <http://proceedings.mlr.press/v80/haarnoja18b.html>
- Wei Li, Tongtong Wang, Zherong Pan, Xifeng Gao, Kui Wu, and Mathieu Desbrun. 2023. High-Order Moment-Encoded Kinetic Simulation of Turbulent Flows. *ACM Trans. Graph.* 42, 6, Article 190 (Dec. 2023), 13 pages.
- Zhaowu Lin, Andrew Hess, Zhaosheng Yu, Shengqiang Cai, and Tong Gao. 2019. A fluid–structure interaction study of soft robotic swimmer using a fictitious domain/active-strain method. *J. Comput. Phys.* 376 (2019), 1138–1155.
- Wenji Liu, Kai Bai, Xuming He, Shuran Song, Changxi Zheng, and Xiaopei Liu. 2022. Fishgym: A high-performance physics-based simulation framework for underwater robot learning. In *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 6268–6275.
- Chaoyang Lyu, Kai Bai, Yiheng Wu, Mathieu Desbrun, Changxi Zheng, and Xiaopei Liu. 2023. Building a Virtual Weakly-Compressible Wind Tunnel Testing Facility. *ACM Trans. Graph.* 42, 4, Article 125 (July 2023), 20 pages.
- Pingchuan Ma, Tao Du, John Z Zhang, Kui Wu, Andrew Spielberg, Robert K Katschmann, and Wojciech Matusik. 2021. DiffAqua: A Differentiable Computational Design Pipeline for Soft Underwater Swimmers with Shape Interpolation. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 132.

- Sehee Min, Jungdam Won, Seunghwan Lee, Jungnam Park, and Jehee Lee. 2019. Softcon: Simulation and control of soft-bodied animals with biomimetic actuators. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 1–12.
- Tsun-Hsuan Wang, Pingchuan Ma, Andrew Everett Spielberg, Zhou Xian, Hao Zhang, Joshua B Tenenbaum, Daniela Rus, and Chuang Gan. 2023a. Softzoo: A soft robot co-design benchmark for locomotion in diverse environments. *arXiv preprint arXiv:2303.09555* (2023).
- Tsun-Hsuan Johnson Wang, Juntian Zheng, Pingchuan Ma, Yilun Du, Byungchul Kim, Andrew Spielberg, Josh Tenenbaum, Chuang Gan, and Daniela Rus. 2023b. Diffusebot: Breeding soft robots with physics-augmented generative diffusion models. *Advances in Neural Information Processing Systems* 36 (2023), 44398–44423.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [NA].
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes]" is generally preferable to "[No]", it is perfectly acceptable to answer "[No]" provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No]" or "[NA]" is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading “NeurIPS Paper Checklist”,**
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers.**

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope?

Answer: [Yes]

Justification: The main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: The paper discuss the limitations of the work performed by the authors.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and conclusions of the paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We open-source our full codebase—including the GPU-accelerated simulator and training pipelines—at <https://github.com/changyu-hu/FreeFlow>.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [\[Yes\]](#)

Justification: The paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [\[Yes\]](#)

Justification: The paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [\[Yes\]](#)

Justification: The paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: There is no societal impact of the work performed.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The creators or original owners of assets (e.g., code, data, models), used in the paper, are properly credited and the license and terms of use are explicitly mentioned and properly respected.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The new assets introduced in the paper are well documented and the documentation is provided in our open source repository: <https://github.com/changyu-hu/FreeFlow>

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.