ANISOLIFT: ANISOTROPIC LATENT REPRESENTATIONS FOR COARSE PARTICLE LIQUID ENHANCEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Particle-based liquid simulation is widely used in graphics and physical modeling, but high-resolution rollouts remain computationally expensive. Consequently, many methods aim to recover fine-scale dynamics and dense transport patterns from coarse particle simulations. However, these methods typically rely on additional particle generation, which still incurs considerable computational overhead and leads to poor representation. We propose **AnisoLift**, a structured latent closure framework that augments each coarse particle with learnable anisotropic ellipsoidal components. This allows the model to capture directional local structure from the underlying high-resolution flow without introducing extra particles. Given a coarse simulation, our model predicts residual corrections to particle states to bring the updated state closer to the aligned high-resolution teacher. Our training objective jointly supervises particle dynamics and anisotropic geometric structure, encouraging consistency with the reference particle states and local geometric structure. Experiments show that our approach enhances coarse liquid simulations through improving fidelity to fully resolved flow behavior.

1 INTRODUCTION

Particle-based liquid simulation is widely used in graphics and physical modeling for the ability to represent free-surface motion and complex liquid interactions (Ohana et al., 2024; Liu et al., 2026; Xu et al., 2025; Enomoto et al., 2024; Gingold & Monaghan, 1977; Koshizuka & Oka, 1996; Müller et al., 2003; Koschier et al., 2022; Jiang et al., 2015; Hu et al., 2018; Sulsky et al., 1995; Groot & Warren, 1997). However, the high-quality results are grounded in high-resolution rollouts, which in turn remain computationally expensive, especially over long time horizons and large simulation domains (Nielsen & Bridson, 2011; Nabizadeh et al., 2024; Wang et al., 2025). In light of this, coarse particles have been applied in practice. Nevertheless, they usually fail to resolve the fine-scale motion, interface detail, and local structure that distinguish high-quality liquid behavior. As a result, a gap remains between the efficiency of the coarse simulation and the fidelity of fully resolved liquid behavior (Um et al., 2018; Roy et al., 2021). To reduce this gap, existing work has largely taken two directions. One focuses on learned particle simulators, using graph-based or continuous-kernel models to improve rollout accuracy and long-horizon stability in Lagrangian fluid dynamics (Sanchez-Gonzalez et al., 2020; Ummenhofer et al., 2020; Prantl et al., 2022; Toshev et al., 2024a). These methods primarily aim to approximate the underlying particle dynamics, seeking to accurately learn update rules over extended rollouts with stability, but they largely overlook the limited expressiveness of a coarse particle representation. The other addresses the problem through coarse-to-high enhancement, seeking to restore fine-scale liquid detail from coarse simulations (Roy et al., 2021; Um et al., 2018). Although these methods target visual and structural recovery, they typically introduce additional particles. Consequently, neither direction succeeds in enhancing coarse liquid simulation quality under a fixed particle budget.

Our starting point is that coarse solvers often preserve the global motion of the flow, while the particles lack the capacity to encode directional local structures in the high-resolution state. This suggests enriching the local state carried by each coarse particle generated by solvers like Smoothed Particle Hydrodynamics (SPH) (Gingold & Monaghan, 1977) and Moving Particle Simulation (MPS) (Koshizuka & Oka, 1996), rather than increasing the number of particles.

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.

We therefore introduce **AnisoLift**, a coarse particle liquid enhancement framework built around anisotropic latent representations, shown in the Fig. 2. During training, aligned reference trajectories provide supervision for learning both particle-level residual corrections and anisotropic local structure. Each coarse particle carries ellipsoidal latent components that encode directional geometry beyond an isotropic point sample, while the model predicts residual corrections on top of a stable coarse solver. Joint supervision on residual motion and anisotropic geometry encourages the learned component to capture unresolved local structure while retaining the coarse solver as the transport backbone. As shown in Fig. 1, our method enhanced the quality of the coarse.

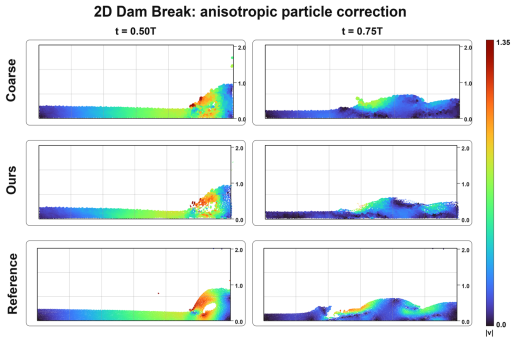

Figure 1: Qualitative correction on the 2D dam-break scene. Coarse particles, our corrected anisotropic coarse state, and the high-resolution reference are shown at two time instants, with particles colored by velocity magnitude.

Our experiments on Lagrangian 2D and 3D datasets demonstrate that AnisoLift improves coarse particle simulations. We show that per-particle enrichment can improve simulation fidelity, particularly when coarse particles under-resolve interface structure and local transport. Our contributions are summarized as follows,

- We formulate fixed-budget coarse-particle liquid enhancement, where the enhanced state preserves the original coarse particle count N_c and improves coarse simulations without particle densification or solver rollout.
- We propose AnisoLift, a structured latent closure that augments each coarse particle with current-frame residual corrections and an anisotropic ellipsoidal footprint.
- We introduce coarse-supported reference alignment, which constructs particle-wise supervision by locally aggregating reference particles around each coarse particle.
- We validate AnisoLift on 2D and 3D particle benchmarks generated by SPH and MPS solvers, and analyze its efficiency, ablations, and resolution scaling behavior.

2 Related Work

Particle-based liquid simulation. Particle-based methods are widely used for liquid simulation because they naturally handle free surfaces, large deformation, splashes, and topology changes. Classical Lagrangian solvers, including SPH (Monaghan, 1992; 2005; Toshev et al., 2024b) and MPS (Koshizuka & Oka, 1996), represent liquids as moving material samples and update particle states through local interactions. Graphics-oriented particle and particle-grid methods, such as WCSPH (Becker & Teschner, 2007), PCISPH (Solenthaler & Pajarola, 2009), APIC (Jiang et al., 2015), FLIP/PIC methods (Müller et al., 2003), and MPM-based approaches (Sulsky et al., 1995; Hu et al., 2018), further improve stability, incompressibility, and visual fidelity. However, high-quality results usually require dense particle sampling and long rollouts, making fully resolved simulations expensive. Coarse simulations are therefore attractive, but their resolved state is often dominated by sparse particle samples and local interaction neighborhoods. This limits the ability of coarse particles to represent unresolved directional structures like thin sheets, stretched interfaces, and local transport patterns. In contrast, our work focuses on this representational bottleneck rather than replacing the solver used to generate the coarse trajectory. Further discussion is shown in Appendix B.

Learned particle dynamics and detail enhancement. Learning-based methods have been developed to improve particle simulation accuracy and efficiency. Early work estimates particle accelerations from local features (Ladický et al., 2015), while continuous convolution models (Ummenhofer et al., 2020), graph network simulators (Sanchez-Gonzalez et al., 2020), and Neural SPH (Toshev et al., 2024a) learn particle update rules from data. These approaches primarily aim to approximate particle dynamics over rollouts, but the particles are usually still represented as point samples connected through isotropic neighborhoods or learned graph interactions. Another related direction recovers fine-scale liquid detail from coarse simulations. Methods like MLFLIP (Um et al., 2018)

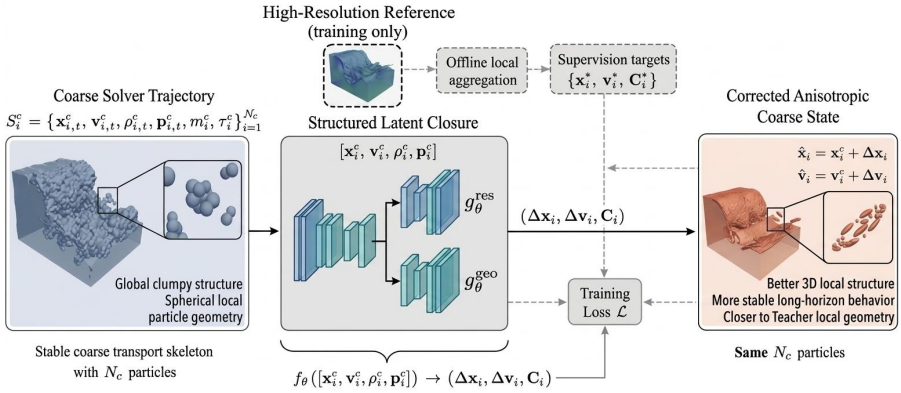

Figure 2: Overview of AnisoLift. Given a precomputed coarse trajectory, the model predicts current-frame residuals $(\Delta x_i, \Delta v_i)$ and anisotropic geometry C_i from coarse particle features. The corrected state preserves the same coarse particle count while enriching each particle with an anisotropic footprint. High-resolution reference particles are used only during training to construct local supervision targets, with no differentiable solver rollout or particle densification.

and Neural UpFlow (Roy et al., 2021) improve visual richness or transport detail by generating additional particles or fine-scale structures. In contrast, AnisoLift studies a fixed particle-count setting: the explicit particle set remains unchanged, while unresolved sub-particle structure is encoded through anisotropic latent components attached to existing particles. Further detail is in Appdix C.

Anisotropic and Gaussian representations. Anisotropic representations have been explored in both particle methods and graphics. ASPH (Owen et al., 1998; Shapiro et al., 1996) replaces scalar smoothing lengths with tensor-valued kernels to adapt particle support to direction-dependent resolution. Anisotropic kernels are used for liquid surface reconstruction and thin-structure preservation (Yu & Turk, 2013). These methods show the value of ellipsoidal support, but the anisotropy is typically derived from local particle statistics, solver heuristics, or reconstruction objectives. Recent Gaussian representations further demonstrate the expressiveness of anisotropic primitives, including 3D Gaussian Splatting for radiance-field rendering (Kerbl et al., 2023) and Gaussian Fluids for continuous velocity-field simulation (Xing et al., 2025). Note that the anisotropic components in AnisoLift are different from both anisotropic kernels and Gaussian field representations. We attach components to coarse Lagrangian particles, learned from aligned high-resolution trajectories, and used as a structured closure for residual state correction. This allows a coarse particle to carry unresolved directional structure without spawning additional particles or modifying the solver.

3 Methodology

3.1 Problem Formulation

We consider particle-based simulation in 2D and 3D, with spatial dimension $d \in \{2, 3\}$. For each scene, we assume paired coarse and high-resolution reference sequences sampled at the same times $\{t_0, \ldots, t_{T-1}\}$. The coarse state is

$$\mathcal{S}_t^c = \left\{ \left(\mathbf{x}_{i,t}^c, \mathbf{v}_{i,t}^c, \rho_{i,t}^c, p_{i,t}^c, m_i^c, \tau_i^c \right) \right\}_{i=1}^{N_c}, \tag{1}$$

where $\mathbf{x}_{i,t}^c, \mathbf{v}_{i,t}^c \in \mathbb{R}^d$ denote particle position and velocity, $\rho_{i,t}^c$ and $p_{i,t}^c$ are density and pressure, m_i^c is mass, and $\tau_i^c \in \{0, 1\}$ is a binary type indicator (1 for fluid, 0 for non-fluid). The reference state is

$$\mathcal{S}_t^r = \left\{ \left(\mathbf{x}_{j,t}^r, \mathbf{v}_{j,t}^r, m_j^r, \tau_j^r \right) \right\}_{j=1}^{N_r}, \qquad N_c \ll N_r. \tag{2}$$

Let $\mathcal{F}^c = \{i : \tau_i^c = 1\}$ and $\mathcal{F}^r = \{j : \tau_j^r = 1\}$ denote the fluid-particle index sets in the coarse and reference states. When wall or obstacle particles are present, they remain in $\mathcal{S}_t^c$ and are available to the model through their attributes and type labels, but supervision is applied only to fluid particles.

The reference state is used only to construct supervision signals and evaluation targets; our method does not roll out the reference simulation or require a differentiable solver.

Rather than generating a denser particle set, we keep the coarse particle set fixed and learn a current-frame closure that corrects the coarse state toward the high-resolution reference. Given $\mathcal{S}_t^c$, the model predicts residual corrections $(\Delta\mathbf{x}_{i,t}, \Delta\mathbf{v}_{i,t})$ for each fluid particle $i \in \mathcal{F}^c$, yielding

$$\widehat{\mathbf{x}}_{i,t} = \mathbf{x}_{i,t}^c + \Delta\mathbf{x}_{i,t}, \qquad \widehat{\mathbf{v}}_{i,t} = \mathbf{v}_{i,t}^c + \Delta\mathbf{v}_{i,t}. \tag{3}$$

The model also predicts an anisotropic covariance $\mathbf{C}_{i,t} \in \mathbb{S}_{++}^d$, which provides each coarse particle with a local geometric descriptor of unresolved reference structure. Unlike next-step prediction, these residuals are applied only to the current observed frame and are not fed back into the solver. The coarse trajectory is generated by a stable solver, while the learned closure improves coarse-state expressiveness at the same temporal resolution as the reference data.

3.2 Paired Trajectory Generation

For each benchmark scene, we construct paired low- and high-resolution simulation sequences using the same underlying particle solver. The coarse sequence and the high-resolution reference sequence are generated independently at different spatial resolutions, but are sampled at identical physical times. Specifically, both sequences share the same simulation duration t_{end}, output interval, and number of saved frames T:

$$t_k = k\Delta t_{\text{out}}, \qquad k = 0, \dots, T-1, \tag{4}$$

where

$$\Delta t_{\text{out}} = \Delta t_{\text{solver}} \cdot n_{\text{write}}, \qquad T = \frac{t_{\text{end}}}{\Delta t_{\text{out}}} + 1. \tag{5}$$

Here Δt_{solver} is the solver integration step and n_{write} is the number of solver steps between two saved frames. This construction ensures that $\mathcal{S}_{t_k}^c$ and $\mathcal{S}_{t_k}^r$ are temporally aligned for every frame t_k.

The coarse sequence $\{\mathcal{S}_{t_k}^c\}_{k=0}^{T-1}$ is produced by a stable solver rollout at a reduced spatial resolution. It serves as the transport skeleton on which our model operates. The corresponding high-resolution reference sequence $\{\mathcal{S}_{t_k}^r\}_{k=0}^{T-1}$ provides supervision for learning current-frame corrections and local geometric structure. Since the two simulations use different spatial resolutions, their particle counts generally satisfy $N_c \ll N_r$ and their particles do not have one-to-one correspondence. This construction assumes that the coarse trajectory remains close enough to the corresponding reference flow so that the local neighborhood around each coarse particle contains physically relevant reference particles. When the coarse solver drifts substantially, the neighborhood may become empty or may switch rapidly across frames, making the target less reliable. In implementation, empty neighborhoods are handled by numerical safeguards, but the method is not intended to recover global transport modes completely missing from the coarse trajectory.

After generating paired sequences, the solver is no longer invoked during training. Each training sample is a saved frame $\left(\mathcal{S}_{t_k}^c, \mathcal{S}_{t_k}^r\right)$ from the dataset, rather than a state produced by recursively applying learned corrections through a differentiable simulator. This decoupled setting has two practical advantages. First, the learned residuals cannot destabilize the coarse solver rollout, because they are not fed back into the simulation process. Second, the learning problem focuses on improving the representation of each coarse frame under the same temporal resolution as the reference data.

3.3 Coarse-Supported Reference Alignment

Since the coarse and reference simulations use different spatial resolutions, their particles do not admit one-to-one correspondence. Directly comparing $\mathbf{x}_{i,t}^c$ with an arbitrary reference particle $\mathbf{x}_{j,t}^r$ is therefore ill-defined. We instead construct coarse-supported reference targets by locally aggregating the high-resolution reference state around each coarse particle, as shown in Fig 4 in supplements.

For each saved frame t_k and coarse fluid particle $i \in \mathcal{F}^c$, we define a local reference neighborhood:

$$\mathcal{N}_{i,t_k}^r = \left\{ j \in \mathcal{F}^r \;\middle|\; \left\|\mathbf{x}_{j,t_k}^r - \mathbf{x}_{i,t_k}^c\right\|_2 \le R_i \right\}, \tag{6}$$

where R_i is the support radius associated with the coarse particle. In practice, R_i is chosen from the coarse particle spacing and solver smoothing scale.

We assign each reference neighbor a distance-based weight

$$\omega_{ij,t_k} = K\left(\frac{\left\|\mathbf{x}^r_{j,t_k} - \mathbf{x}^c_{i,t_k}\right\|_2}{R_i}\right), \qquad j \in \mathcal{N}^r_{i,t_k}, \tag{7}$$

where $K(\cdot)$ is a compactly supported radial kernel. The normalized weights are:

$$\alpha_{ij,t_k} = \frac{\omega_{ij,t_k}}{\sum_{\ell \in \mathcal{N}^r_{i,t_k}} \omega_{i\ell,t_k}}. \tag{8}$$

In practice, a small constant ϵ is added to the denominator for numerical robustness against vanishing weight sums. The aligned reference position and velocity for coarse particle i are then computed as:

$$\mathbf{x}^*_{i,t_k} = \sum_{j \in \mathcal{N}^r_{i,t_k}} \alpha_{ij,t_k} \mathbf{x}^r_{j,t_k}, \qquad \mathbf{v}^*_{i,t_k} = \sum_{j \in \mathcal{N}^r_{i,t_k}} \alpha_{ij,t_k} \mathbf{v}^r_{j,t_k}. \tag{9}$$

These targets provide frame-wise supervision for the corrected coarse state $\widehat{\mathbf{x}}_{i,t_k}$ and $\widehat{\mathbf{v}}_{i,t_k}$.

We extract a local second-order structure from the same reference neighborhood. The mean target captures the local reference center, while the covariance target preserves the spatial extent and anisotropy that would be lost by a pointwise target alone. The covariance target is:

$$\mathbf{C}^*_{i,t_k} = \sum_{j \in \mathcal{N}^r_{i,t_k}} \alpha_{ij,t_k} \left(\mathbf{x}^r_{j,t_k} - \mathbf{x}^*_{i,t_k}\right)\left(\mathbf{x}^r_{j,t_k} - \mathbf{x}^*_{i,t_k}\right)^\top + \epsilon_{\text{geo}}\mathbf{I}. \tag{10}$$

Here $\epsilon_{\text{geo}} > 0$ is a small constant. The diagonal jitter $\epsilon_{\text{geo}}\mathbf{I}$ ensures that $\mathbf{C}^*_{i,t_k}$ is positive definite. This covariance represents the anisotropic spatial extent of the high-resolution reference region associated with coarse particle i. The result of this step is a set of coarse-indexed supervision signals:

$$\mathcal{Y}^*_{t_k} = \left\{\left(\mathbf{x}^*_{i,t_k}, \mathbf{v}^*_{i,t_k}, \mathbf{C}^*_{i,t_k}\right)\right\}_{i \in \mathcal{F}^c}, \tag{11}$$

which is indexed by the coarse fluid set $\mathcal{F}^c$. This makes it possible to train a fixed-size coarse-particle model even when the reference simulation contains many more particles.

3.4 Structured Latent Closure with Ellipsoids

Given the coarse state $\mathcal{S}^c_{t_k}$ at a frame t_k, our model predicts a structured closure for each coarse particle. The closure has two coupled components: a current-frame state residual and an anisotropic ellipsoidal latent representation. The residual corrects the coarse particle center, while the ellipsoid represents the local spatial extent of the reference region associated with that coarse particle.

For a coarse particle $i \in \mathcal{F}^c$, we construct a feature vector:

$$\mathbf{h}_{i,t_k} = \phi_\theta\left(\mathcal{S}^c_{t_k}, i\right), \tag{12}$$

where ϕ_θ denotes the learnable feature extractor operating on the current coarse state. The input features are derived from the coarse particle attributes defined in Problem Formulation section, including position, velocity, density, pressure, mass, and particle type.

The residual head predicts whitened corrections:

$$\left(\bar{\Delta}\mathbf{x}_{i,t_k}, \bar{\Delta}\mathbf{v}_{i,t_k}\right) = g^{\text{res}}_\theta\left(\mathbf{h}_{i,t_k}\right), \tag{13}$$

where the per-component normalization statistics $(\boldsymbol{\mu}_{\Delta x}, \boldsymbol{\sigma}_{\Delta x})$ and $(\boldsymbol{\mu}_{\Delta v}, \boldsymbol{\sigma}_{\Delta v})$ are estimated once from the training-split residual targets $\Delta\mathbf{x}^*_{i,t_k} = \mathbf{x}^*_{i,t_k} - \mathbf{x}^c_{i,t_k}$ and $\Delta\mathbf{v}^*_{i,t_k} = \mathbf{v}^*_{i,t_k} - \mathbf{v}^c_{i,t_k}$, induced by the coarse-supported reference targets. The physical residuals are recovered by

$$\Delta\mathbf{x}_{i,t_k} = \bar{\Delta}\mathbf{x}_{i,t_k} \odot \boldsymbol{\sigma}_{\Delta x} + \boldsymbol{\mu}_{\Delta x}, \quad \Delta\mathbf{v}_{i,t_k} = \bar{\Delta}\mathbf{v}_{i,t_k} \odot \boldsymbol{\sigma}_{\Delta v} + \boldsymbol{\mu}_{\Delta v}. \tag{14}$$

where $\odot$ denotes the Hadamard product. The corrected particle state is then defined by Equation 3 . This is a current-frame closure: the predicted residual is used to improve the state at t_k and is not integrated forward by the particle solver.

To enrich the representation of each coarse particle, we also predict an anisotropic covariance matrix:

$$\mathbf{C}_{i,t_k} = g^{\text{geo}}_\theta\left(\mathbf{h}_{i,t_k}\right), \qquad \mathbf{C}_{i,t_k} \in \mathbb{S}^d_{++}, \tag{15}$$

where $\mathbb{S}^d_{++}$ denotes the space of $d \times d$ symmetric positive definite matrices. In practice, positive definiteness is enforced by predicting a matrix factor and forming:

$$\mathbf{C}_{i,t_k} = \mathbf{L}_{i,t_k}\mathbf{L}_{i,t_k}^\top + \epsilon_{\text{geo}}\mathbf{I}, \tag{16}$$

where $\epsilon_{\text{geo}} > 0$ is the same small constant as in Equation 10.

The pair $(\widehat{\mathbf{x}}_{i,t_k}, \mathbf{C}_{i,t_k})$ defines an anisotropic ellipsoid centered at the corrected coarse particle position; its level set can be written as

$$\mathcal{E}_{i,t_k} = \left\{ \mathbf{y} \in \mathbb{R}^d \;\middle|\; (\mathbf{y} - \widehat{\mathbf{x}}_{i,t_k})^\top \mathbf{C}_{i,t_k}^{-1} (\mathbf{y} - \widehat{\mathbf{x}}_{i,t_k}) \le 1 \right\}. \tag{17}$$

Thus, a coarse particle is no longer treated as an isotropic point sample. Instead, it acts as a structured latent element that summarizes an anisotropic region of the reference flow.

This representation is useful when $N_c < N_r$. The corrected center $\widehat{\mathbf{x}}_{i,t_k}$ and velocity $\widehat{\mathbf{v}}_{i,t_k}$ capture the local dynamics, while $\mathbf{C}_{i,t_k}$ captures the local geometry encoded by the coarse-supported reference covariance $\mathbf{C}^*_{i,t_k}$. The resulting model preserves the computationally light coarse particle set while providing a richer description of the fluid structure.

3.5 Training Objective

The model is trained with supervision constructed from the coarse-supported reference targets $\mathcal{Y}^*_{t_k}$ defined in Section 3.3. The objective contains two groups of constraints: dynamics constraints on the corrected coarse state and geometry constraints on the anisotropic ellipsoidal representation.

The corrected state $\widehat{\mathbf{x}}_{i,t_k}$ and $\widehat{\mathbf{v}}_{i,t_k}$ is defined as in Equation 3. We supervise it with position and velocity correction losses:

$$\mathcal{L}_x = \frac{1}{TN_f^c} \sum_{k=0}^{T-1} \sum_{i \in \mathcal{F}^c} \left\| \widehat{\mathbf{x}}_{i,t_k} - \mathbf{x}^*_{i,t_k} \right\|_2^2, \tag{18}$$

$$\mathcal{L}_v = \frac{1}{TN_f^c} \sum_{k=0}^{T-1} \sum_{i \in \mathcal{F}^c} \left\| \widehat{\mathbf{v}}_{i,t_k} - \mathbf{v}^*_{i,t_k} \right\|_2^2, \tag{19}$$

where $N_f^c = |\mathcal{F}^c|$ is the number of coarse fluid particles.

To compare kinetic behavior across different particle resolutions, we use a fluid-only mass-normalized kinetic energy. The corrected coarse specific kinetic energy is:

$$\widehat{e}^c_{\text{kin}}(t_k) = \frac{\sum_{i \in \mathcal{F}^c} \frac{1}{2} m_i^c \left\| \widehat{\mathbf{v}}_{i,t_k} \right\|_2^2}{\sum_{i \in \mathcal{F}^c} m_i^c}, \tag{20}$$

and the reference specific kinetic energy is:

$$e^r_{\text{kin}}(t_k) = \frac{\sum_{j \in \mathcal{F}^r} \frac{1}{2} m_j^r \left\| \mathbf{v}^r_{j,t_k} \right\|_2^2}{\sum_{j \in \mathcal{F}^r} m_j^r}. \tag{21}$$

In practice, a small constant is added to each denominator for numerical robustness against vanishing total mass. The kinetic-energy auxiliary loss is then:

$$\mathcal{L}_{\text{ekin}} = \frac{1}{T} \sum_{k=0}^{T-1} \left(\widehat{e}^c_{\text{kin}}(t_k) - e^r_{\text{kin}}(t_k) \right)^2. \tag{22}$$

For the anisotropic ellipsoidal representation, we compare the predicted covariance $\mathbf{C}_{i,t_k}$ with the coarse-supported reference covariance $\mathbf{C}^*_{i,t_k}$. Since both matrices are symmetric positive definite, we use a log-covariance geometry loss:

$$\mathcal{L}_{\text{geo}} = \frac{1}{TN_f^c} \sum_{k=0}^{T-1} \sum_{i \in \mathcal{F}^c} \left\| \log \mathbf{C}_{i,t_k} - \log \mathbf{C}^*_{i,t_k} \right\|_F^2. \tag{23}$$

Here $\log(\cdot)$ is the matrix logarithm and $\|\cdot\|_F$ is the Frobenius norm. The final training objective is:

$$\mathcal{L} = \lambda_x \mathcal{L}_x + \lambda_v \mathcal{L}_v + \lambda_{\text{ekin}} \mathcal{L}_{\text{ekin}} + \lambda_{\text{geo}} \mathcal{L}_{\text{geo}}, \tag{24}$$

where the λ coefficients control the relative strength of the dynamics and geometry constraints. For clarity, we write the loss for a single trajectory, in practice it is averaged over all training trajectories. Implementation details are summarized in Algorithm 1 and 2 in Appendix B.1.

4 Experiments

4.1 Experimental Setup and Implementation

We evaluate on paired coarse and high-resolution reference sequences. Models are trained on the training split, selected by the validation composite score, and evaluated on the held-out test split on 2 NVIDIA H200 GPUs. Detailed hyperparameters are reported in Table 9 in Appendix B.3. Evaluation metrics are introduced in Appendix A.2.

We use a two-layer MLP with hidden dimension 64 to predict current-frame residuals $(\Delta \mathbf{x}i, t, \Delta \mathbf{v}i, t)$ and anisotropic covariance $\mathbf{C}i, t$ from coarse particle features. Unless otherwise specified, we use Adam with learning rate 3×10^{-4}, batch size 32, gradient clipping norm 1.0, and loss weights $\lambda \Delta x = \lambda_{\Delta v} = 2.0$, $\lambda_{\text{geo}} = 1.0$, and $\lambda_{\text{ekin}} = 0.5$. For 3D TGV, we use learning rate $1e-4$ and clipping norm 0.5.

4.2 Results

We report mean squared error (MSE) in position, velocity and the mass-normalized specific kinetic-energy error. Each sample is a saved frame $\left(\mathcal{S}^c_{t_k}, \mathcal{S}^r_{t_k}\right)$. The model takes the coarse state $\mathcal{S}^c_{t_k}$ as input and predicts current-frame residual corrections for the coarse particles. The reference $\mathcal{S}^r_{t_k}$ is used only to construct coarse-supported supervision targets $\mathcal{Y}^*_{t_k}$ and to compute evaluation metrics, which are introduced in supplements. We answer three questions: (i) whether AnisoLift improves coarse particle fidelity under a fixed particle budget; (ii) whether anisotropic geometry provides useful local structural information beyond state residuals; and (iii) whether the additional inference stage preserves the efficiency advantage. We show quantitative results in the Table 1, where we can see our method improve the fidelity of coarse particles generated by SPH and MPS. In Figure 3 in supplements, we show our learned structure in the middle row on 2D DB, ellipsoidal shape can be observed in each particle, and our method shows cavities appearing when water waves recede in $t = 0.50T$, while coarse particles failed to construct such structures. In addition to particle-state metrics, we provide qualitative visualizations in Figure 1 and Figures 3-6 in Appendix A.3. We also provide runtime analysis in Appendix B.2.

4.3 Datasets

We evaluate on multiple particle-based liquid simulation scenarios in both 2D and 3D, including Taylor–Green vortex (TGV), dam break (DAM), lid-driven cavity (LDC), and reverse Poiseuille flow (RPF). For each scenario, we construct paired coarse and high-resolution reference sequences with matched physical duration, output interval, and number of saved frames. The coarse sequence uses fewer particles and provides the low-resolution input, while the high-resolution sequence provides supervision and evaluation targets. More details are shown in Appendix A.4.

Paired trajectory generation. We generate paired coarse and high-resolution reference trajectories with the same solver family, physical setup, and saved temporal resolution. The coarse trajectory is a stable precomputed solver rollout and is used directly during training, so no differentiable or closed-loop solver rollout is required. We evaluate both SPH and MPS trajectories, generated with JAX-SPH and MPS-Basic, respectively, and apply quality checks to reject trajectories with non-finite states. Detailed particle counts, spatial resolutions, time steps, split sizes, boundary conditions, and solver-level numerical parameters are provided in Supplementary Tables 5, 6, and 7.

Particle solvers. We use two classical particle solvers to generate paired simulation data: SPH and MPS. SPH represents the fluid as Lagrangian particles and estimates field quantities through kernel

Table 1: Quantitative results on 2D and 3D paired particle simulation datasets. Ekin denotes fluid-only mass-normalized specific kinetic energy. The scale in each dataset header applies to all three metrics. Smaller value reflects better performance.

Solver	2D TGV (1e-5)			2D DAM (1e-3)			2D LDC (1e-3)			2D RPF (1e-3)		
	MSE_x	MSE_v	$\mathrm{MSE}_{\mathrm{Ekin}}$	MSE_x	MSE_v	$\mathrm{MSE}_{\mathrm{Ekin}}$	MSE_x	MSE_v	$\mathrm{MSE}_{\mathrm{Ekin}}$	MSE_x	MSE_v	$\mathrm{MSE}_{\mathrm{Ekin}}$
SPH_c	7.750	256.46	2.711	2.560	35.92	0.250	0.6420	21.79	0.013	0.0800	8.200	2.350
$\mathrm{SPH}_{\mathrm{Ours}}$	**2.010**	**221.60**	**2.350**	**1.100**	**22.34**	**0.110**	**0.0020**	**2.690**	**0.003**	**0.0080**	**1.690**	**0.170**
MPS_c	12.70	460.78	387.20	7.330	710.27	32.40	7.050	203.70	1.480	0.1200	7.950	61.51
$\mathrm{MPS}_{\mathrm{Ours}}$	**7.600**	**367.84**	**298.66**	**3.710**	**368.99**	**28.19**	**1.510**	**91.70**	**0.480**	**0.0300**	**1.580**	**8.300**

Solver	3D TGV (1e-3)			3D LDC (1e-3)			3D RPF (1e-3)		
	MSE_x	MSE_v	$\mathrm{MSE}_{\mathrm{Ekin}}$	MSE_x	MSE_v	$\mathrm{MSE}_{\mathrm{Ekin}}$	MSE_x	MSE_v	$\mathrm{MSE}_{\mathrm{Ekin}}$
SPH_c	0.0519	21.26	0.0397	0.32	3.16	0.009	4.54	8.99	2.90
$\mathrm{SPH}_{\mathrm{Ours}}$	**0.0517**	**12.54**	**0.0012**	**0.03**	**0.49**	**0.008**	**2.87**	**2.15**	**1.88**
MPS_c	15.84	79.64	48.15	0.50	9.92	0.110	1.88	9.00	3.11
$\mathrm{MPS}_{\mathrm{Ours}}$	**5.43**	**39.82**	**32.53**	**0.41**	**4.89**	**0.004**	**0.08**	**2.85**	**1.22**

interpolation over neighboring particles. A generic SPH approximation of a scalar field a at particle i can be written as

$$a_i \approx \sum_j \frac{m_j}{\rho_j} a_j W\left(\|\mathbf{x}_i - \mathbf{x}_j\|, h\right), \tag{25}$$

where $W(\cdot, h)$ is a smoothing kernel with support radius controlled by h. In our experiments, SPH provides stable low- and high-resolution particle rollouts with matched times. MPS is also a fully Lagrangian particle method, but does not rely on the standard SPH density-kernel formulation. It computes particle number density from neighbor weights and solves pressure implicitly to enforce near-incompressibility. A typical MPS number density estimate is

$$n_i = \sum_{j \neq i} w\left(\|\mathbf{x}_j - \mathbf{x}_i\|\right), \tag{26}$$

where $w(\cdot)$ is the MPS interaction weight. We include MPS-generated datasets to test whether the proposed closure is tied to SPH-specific discretization artifacts.

4.4 Ablation Study

Loss-component ablation on 2D LDC. We ablate the main loss terms on the 2D lid-driven cavity dataset using SPH-generated paired trajectories. In Table 2, State denotes $\lambda_x \mathcal{L}_x + \lambda_v \mathcal{L}_v$, and residual denotes $\lambda_{\Delta x} \mathcal{L}_{\Delta x} + \lambda_{\Delta v} \mathcal{L}_{\Delta v}$. The coarse baseline has no learned correction. For learned variants, we remove one loss group at a time: corrected-state losses, residual-consistency losses, log-covariance geometry loss, and specific kinetic-energy loss. Table 2 shows that full model achieves comprehensive balanced performance on position, velocity and kinetic energy, and there is a tradeoff between kinetic energy and other two metrics. Removing corrected-state losses degrades position and velocity accuracy, while removing residual-consistency losses reduces correction quality. The geometry loss is essential for learning meaningful anisotropic structure, and the kinetic-energy term regularizes global behavior, preventing locally accurate but physically inconsistent updates. Overall, the results support combining state supervision, residual consistency, geometric supervision, and a fluid-only kinetic-energy regularizer. These components are complementary: state and residual terms drive local accuracy, geometry shapes anisotropic representation, and the kinetic term improves global consistency.

Table 2: Loss terms ablation on 2D LDC. C/O omitted; w/o Geometry constrains $\mathbf{C}_{i,t} = \mathbf{I}$.

Variant	MSE_x	MSE_v	$\mathrm{MSE}_{\mathrm{Ekin}}$
Coarse	6.420e−4	2.179e−2	1.339e−5
w/o Ekin	2.802e−4	1.104e−2	8.524e−5
w/o Geometry	6.895e−6	2.930e−3	2.147e−6
w/o Residual	3.210e−6	2.984e−3	1.964e−6
w/o State	5.552e−6	3.398e−3	3.348e−6
AnisoLift	2.102e−6	2.659e−3	3.131e−6

Table 3: Resolution scaling on 2D TGV. The high-resolution reference is fixed with $N_r = 2500$. Gain denotes the relative error reduction of our method over the corresponding coarse baseline.

N_c	MSE_x			MSE_v			$\mathrm{MSE}_{\mathrm{Ekin}}$		
	Coarse	Ours	Gain	Coarse	Ours	Gain	Coarse	Ours	Gain
484	7.754e−5	2.009e−5	74.1%	2.565e−3	2.216e−3	13.6%	2.711e−5	2.357e−5	15.0%
1225	2.029e−5	9.203e−6	54.7%	8.130e−4	7.054e−4	13.2%	3.473e−6	2.973e−6	14.4%

Anisotropic vs. isotropic representation on 2D LDC. We ablate the anisotropic footprint by replacing the full covariance with an isotropic covariance parameterized by a single scale, while keeping the same current-frame correction pipeline. Lines in Table 2 with coarse, w/o geometry, and AnisoLift in the variant column compare the coarse baseline, isotropic variant, and full AnisoLift on SPH-generated 2D LDC trajectories. The variant already improves over the coarse baseline through residual state correction, but the anisotropic model better captures direction-dependent local support in shear-dominated and wall-driven regions. This shows that anisotropic geometry provides additional representational capacity beyond state correction alone.

Table 4: Surface-level diagnosis. 2D dam-break surface diagnostics under SPH and MPS. Higher IoU/F1 and lower H-MAE are better.

Solver	Variant	IoU ↑	F1 ↑	H-MAE ↓
SPH	Coarse	0.6433	0.7811	7.077e−1
	AnisoLift	0.8055	0.8911	1.121e−1
MPS	Coarse	0.3161	0.4739	3.468e−1
	AnisoLift	0.4709	0.6355	2.673e−1

Resolution scaling on 2D TGV. We evaluate the proposed correction under different coarse particle budgets on the 2D Taylor–Green vortex dataset. The high-resolution reference is fixed at $N_r = 2500$, while coarse trajectories are generated with varying N_c. This tests whether the structured latent closure remains effective when the coarse simulation is either highly sparse or moderately resolved. Table 3 reports results for $N_c = 484$ and $N_c = 1225$. In both cases, our method reduces the position error relative to the coarse baseline, indicating that the learned current-frame residual correction consistently improves spatial fidelity across resolutions. The gain is larger at lower resolution, where the coarse transport skeleton misses more local structure, and smaller but still positive at higher resolution, where the baseline is already closer to the reference. Improvements in velocity and kinetic energy are more moderate. This is because the correction is applied frame-wise without differentiable solver rollout, while the coarse solver already provides a stable velocity field at the same temporal resolution. Overall, the method is most beneficial under strong coarse compression, while still yielding measurable gains at higher coarse resolutions.

Surface-level diagnostics. To complement particle-state errors with visually relevant free-surface evidence, we evaluate occupancy IoU, F1, and height-profile MAE on the 2D dam-break benchmark. AnisoLift improves IoU/F1 from 0.643/0.781 to 0.806/0.891 on SPH trajectories and from 0.316/0.474 to 0.471/0.636 on MPS trajectories, while reducing height-profile MAE from 0.707 to 0.112 on SPH and from 0.347 to 0.267 on MPS. The supplementary Figure 5 in Appendix A.3 further shows that the corrected fluid-region contour and free-surface height profile are consistently closer to the high-resolution reference at both early and late stages of the dam-break evolution. These results indicate that the learned anisotropic footprints improve visible fluid support and surface geometry, not only particle-wise state errors, as summarized in Table 4.

5 Conclusion and Discussion

We presented AnisoLift, a fixed-budget framework that enhances coarse particle simulations by predicting current-frame residual corrections and anisotropic ellipsoidal representations for existing particles. Experiments on paired 2D and 3D benchmarks show improved fidelity without particle densification or differentiable solver rollout. Limitations include dependence on solver-generated supervision, frame-wise rather than closed-loop correction, and limited ability to recover missing global transport. We will explore temporally consistent and tightly coupled extensions in future. Further detailed limitation and future works are discussed in Appendix D.

AI USE STATEMENT

In this work, Generative-AI assistants were used for language editing and manuscript polishing. The authors reviewed all suggested edits and take full responsibility for the manuscript and its claims.

REPRODUCIBILITY STATEMENT

The model architecture and all training hyperparameters are specified in Section 4.1 and Appendix B.3. Data generation parameters for all benchmarks and solvers and the train/validation/test split protocol are detailed in Appendix A.4. The coarse-supported reference alignment procedure is fully specified in Section 3.3. Simulations use the open-source solvers JAX-SPH and MPS-Basic, code and trained checkpoints will be released upon acceptance.

REFERENCES

Markus Becker and Matthias Teschner. Weakly compressible SPH for free surface flows. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2007. doi: 10.1145/1272690.1272719. URL https://dl.acm.org/doi/10.5555/1272690.1272719.

Keigo Enomoto, Takato Ishida, Yuya Doi, Takashi Uneyama, and Yuichi Masubuchi. Extension of moving particle simulation by introducing rotational degrees of freedom for dilute fiber suspensions. *International Journal for Numerical Methods in Fluids*, 96(2):125–137, 2024. doi: https://doi.org/10.1002/fld.5235. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.5235.

R. A. Gingold and J. J. Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181(3):375–389, 12 1977. ISSN 0035-8711. doi: 10.1093/mnras/181.3.375. URL https://doi.org/10.1093/mnras/181.3.375.

Robert D. Groot and Patrick B. Warren. Dissipative particle dynamics: Bridging the gap between atomistic and mesoscopic simulation. *The Journal of Chemical Physics*, 107(11):4423–4435, 09 1997. ISSN 0021-9606. doi: 10.1063/1.474784. URL https://doi.org/10.1063/1.474784.

Yuanming Hu, Yu Fang, Ziheng Ge, Ziyin Qu, Yixin Zhu, Andre Pradhana, and Chenfanfu Jiang. A moving least squares material point method with displacement discontinuity and two-way rigid body coupling. *ACM Trans. Graph.*, 37(4), July 2018. ISSN 0730-0301. doi: 10.1145/3197517.3201293. URL https://doi.org/10.1145/3197517.3201293.

Chenfanfu Jiang, Craig Schroeder, Andrew Selle, Joseph Teran, and Alexey Stomakhin. The affine particle-in-cell method. *ACM Trans. Graph.*, 34(4), July 2015. ISSN 0730-0301. doi: 10.1145/2766996. URL https://doi.org/10.1145/2766996.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023. doi: 10.1145/3592433. URL https://dl.acm.org/doi/10.1145/3592433.

Dan Koschier, Jan Bender, Barbara Solenthaler, and Matthias Teschner. A Survey on SPH Methods in Computer Graphics. *Computer Graphics Forum*, 41(2), 2022. ISSN 1467-8659. doi: 10.1111/cgf.14508.

S. Koshizuka and Y. Oka. Moving-particle semi-implicit method for fragmentation of incompressible fluid. *Nuclear Science and Engineering*, 123(3):421–434, 1996. doi: 10.13182/NSE96-A24205. URL https://doi.org/10.13182/NSE96-A24205.

L'ubor Ladický, SoHyeon Jeong, Barbara Solenthaler, Marc Pollefeys, and Markus Gross. Data-driven fluid simulations using regression forests. *ACM Transactions on Graphics*, 34(6), 2015. doi: 10.1145/2816795.2818129.

Yuqiu Liu, Jialin Song, Marissa Ramirez de Chanlatte, Rochishnu Chowdhury, Rushil Paresh Desai, Wuyang Chen, Daniel Martin, and Michael Mahoney. Fluidgaussian: Propagating simulation-based uncertainty toward functionally-intelligent 3d reconstruction. *CVPR*, 2026.

J. J. Monaghan. Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics*, 30:543–574, 1992. doi: 10.1146/annurev.aa.30.090192.002551.

J. J. Monaghan. Smoothed particle hydrodynamics. *Reports on Progress in Physics*, 68(8):1703–1759, 2005. doi: 10.1088/0034-4885/68/8/R01.

Matthias Müller, David Charypar, and Markus Gross. Particle-based fluid simulation for interactive applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '03, pp. 154–159, Goslar, DEU, 2003. Eurographics Association. ISBN 1581136595.

Mohammad Sina Nabizadeh, Ritoban Roy-Chowdhury, Hang Yin, Ravi Ramamoorthi, and Albert Chern. Fluid implicit particles on coadjoint orbits. *ACM Trans. Graph.*, 43(6), November 2024. ISSN 0730-0301. doi: 10.1145/3687970. URL https://doi.org/10.1145/3687970.

Michael B. Nielsen and Robert Bridson. Guide shapes for high resolution naturalistic liquid simulation. *ACM Trans. Graph.*, 30(4), July 2011. ISSN 0730-0301. doi: 10.1145/2010324.1964978. URL https://doi.org/10.1145/2010324.1964978.

Ruben Ohana, Michael McCabe, Lucas Meyer, Rudy Morel, Fruzsina Agocs, Miguel Beneitez, Marsha Berger, Blakesly Burkhart, Stuart Dalziel, Drummond Fielding, et al. The well: a large-scale collection of diverse physics simulations for machine learning. *Advances in Neural Information Processing Systems*, 37:44989–45037, 2024.

J. Michael Owen, Jens V. Villumsen, Paul R. Shapiro, and Hugo Martel. Adaptive smoothed particle hydrodynamics: Methodology. II. *The Astrophysical Journal Supplement Series*, 116(2):155–209, 1998. doi: 10.1086/313100.

Lukas Prantl, Benjamin Ummenhofer, Vladlen Koltun, and Nils Thuerey. Guaranteed conservation of momentum for learning particle-based fluid dynamics. In *Advances in Neural Information Processing Systems*, volume 35. Neural Information Processing Systems Foundation, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/2dd7f33ffbb59b4ff987be5442a13016-Abstract-Conference.html.

Bruno Roy, Pierre Poulin, and Eric Paquette. Neural upflow: A scene flow learning approach to increase the apparent resolution of particle-based liquids. *Proc. ACM Comput. Graph. Interact. Tech.*, 4(3), September 2021. doi: 10.1145/3480147. URL https://doi.org/10.1145/3480147.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8459–8468. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/sanchez-gonzalez20a.html.

Paul R. Shapiro, Hugo Martel, and Jens V. Villumsen. Adaptive smoothed particle hydrodynamics, with application to cosmology: Methodology. *The Astrophysical Journal Supplement Series*, 103: 269–330, 1996. doi: 10.1086/192279.

B. Solenthaler and R. Pajarola. Predictive-corrective incompressible sph. In *ACM SIGGRAPH 2009 Papers*, SIGGRAPH '09, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605587264. doi: 10.1145/1576246.1531346. URL https://doi.org/10.1145/1576246.1531346.

Deborah Sulsky, Shi-Jian Zhou, and Howard L. Schreyer. Application of a particle-in-cell method to solid mechanics. *Computer Physics Communications*, 87(1):236–252, 1995. ISSN 0010-4655. doi: https://doi.org/10.1016/0010-4655(94)00170-7. URL https://www.sciencedirect.com/science/article/pii/0010465594001707. Particle Simulation Methods.

Artur Toshev, Jonas A. Erbesdobler, Nikolaus A. Adams, and Johannes Brandstetter. Neural SPH: Improved neural modeling of lagrangian fluid dynamics. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 48428–48452. PMLR, 21–27 Jul 2024a. URL https://proceedings.mlr.press/v235/toshev24a.html.

Artur P. Toshev, Gianluca Galletti, Fabian Fritz, Stefan Adami, and Nikolaus A. Adams. Lagrangebench: A lagrangian fluid mechanics benchmarking suite. *arXiv preprint arXiv:2309.16342*, 2023. doi: 10.48550/arXiv.2309.16342. URL https://arxiv.org/abs/2309.16342.

Artur P Toshev, Harish Ramachandran, Jonas A Erbesdobler, Gianluca Galletti, Johannes Brandstetter, and Nikolaus A Adams. Jax-sph: A differentiable smoothed particle hydrodynamics framework. *arXiv preprint arXiv:2403.04750*, 2024b.

Kiwon Um, Xiangyu Hu, and Nils Thuerey. Liquid splash modeling with neural networks. *Computer Graphics Forum*, 37(8):171–182, 2018. doi: 10.1111/cgf.13522. URL https://doi.org/10.1111/cgf.13522.

Benjamin Ummenhofer, Lukas Prantl, Nils Thuerey, and Vladlen Koltun. Lagrangian fluid simulation with continuous convolutions. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://openreview.net/forum?id=B1lDoJSYDH.

Haoxiang Wang, Tao Yu, Hui Qiao, and Qionghai Dai. Neural fluid simulation on geometric surfaces. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 99091–99118, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/f58c24798220ba724fe05c0fa786227d-Paper-Conference.pdf.

Jingrui Xing, Bin Wang, Mengyu Chu, and Baoquan Chen. Gaussian fluids: A grid-free fluid solver based on gaussian spatial representation. In *ACM SIGGRAPH 2025 Conference Papers*, SIGGRAPH '25, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400715402. doi: 10.1145/3721238.3730620. URL https://doi.org/10.1145/3721238.3730620.

Jingxuan Xu, Hong Huang, Chuhang Zou, Manolis Savva, Yunchao Wei, and Wuyang Chen. Hybrid neural-mpm for interactive fluid simulations in real-time. *arXiv preprint arXiv:2505.18926*, 2025.

Jihun Yu and Greg Turk. Reconstructing surfaces of particle-based fluids using anisotropic kernels. *ACM Transactions on Graphics*, 32(1), 2013. doi: 10.1145/2421636.2421641. URL https://dl.acm.org/doi/10.1145/2421636.2421641.

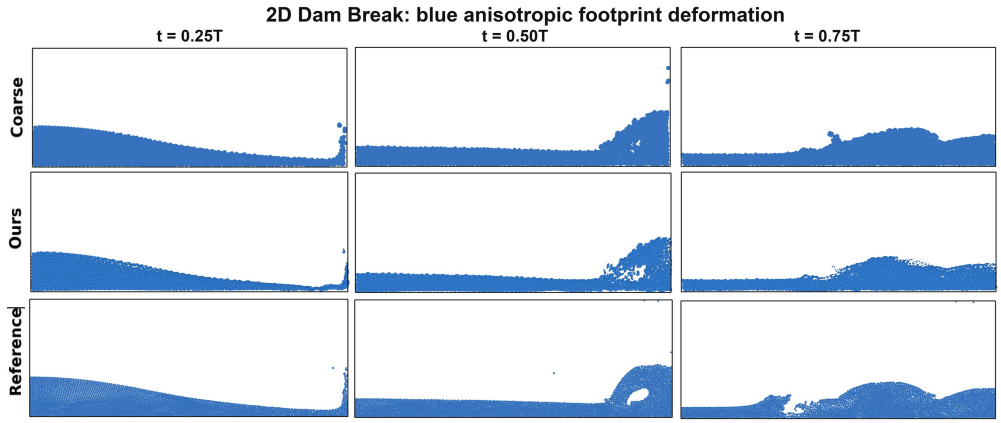

Figure 3: Qualitative visualization on 2D dam break with uniform rendering. AnisoLift keeps the same coarse particles but learns anisotropic ellipsoidal footprints, producing more coherent fallback structures and avoiding abnormal coarse wave crests without particle densification.

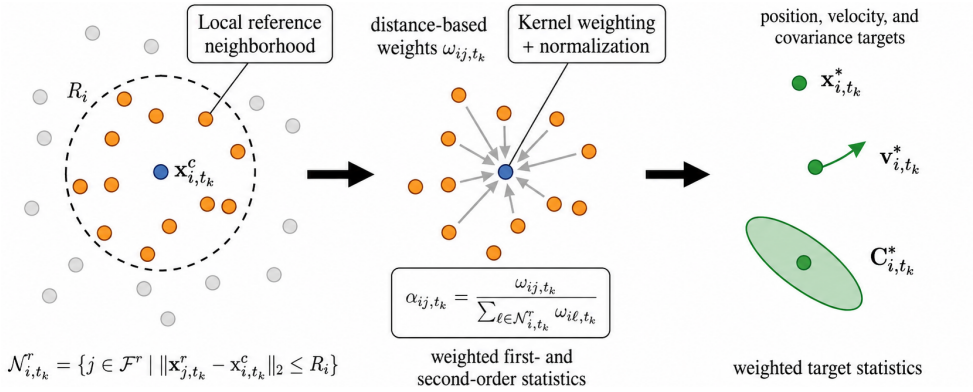

Figure 4: Coarse-supported target construction. Nearby reference particles are locally aggregated around each coarse particle to construct aligned supervision targets $(\mathbf{x}^*_{i,t_k}, \mathbf{v}^*_{i,t_k}, \mathbf{C}^*_{i,t_k})$.

A Appendix

This supplementary material provides implementation details, metric definitions, dataset-generation settings, and additional visual evidence for AnisoLift. The supplementary Figure 4 illustrates coarse-supported target construction. Algorithm 1 and 2 summarize training and inference. Sec. A.2 defines the evaluation metrics. Tables 6–9 report training hyperparameters, dataset statistics, and solver-level numerical settings. Figs. 3, 5 and 6 provide additional qualitative visualizations.

A.1 Coarse-Supported Target Construction

Since coarse and high-resolution particles do not have one-to-one correspondence, supervision targets are constructed on the fixed coarse support. For each coarse particle, nearby reference particles are selected, weighted, and aggregated into aligned position, velocity, and covariance targets. As shown in Figure 4.

A.2 Evaluation Metrics

We evaluate corrected coarse states against coarse-supported reference targets and high-resolution reference sequences. Position and velocity errors are computed on the aligned coarse fluid support:

$$\mathrm{MSE}_x = \frac{1}{TN_f^c} \sum_{k=0}^{T-1} \sum_{i \in \mathcal{F}^c} \left\| \widehat{\mathbf{x}}_{i,t_k} - \mathbf{x}^*_{i,t_k} \right\|_2^2, \tag{27}$$

$$\mathrm{MSE}_v = \frac{1}{TN_f^c} \sum_{k=0}^{T-1} \sum_{i \in \mathcal{F}^c} \left\| \widehat{\mathbf{v}}_{i,t_k} - \mathbf{v}^*_{i,t_k} \right\|_2^2 . \tag{28}$$

For the uncorrected coarse baseline, $\widehat{\mathbf{x}}_{i,t_k}$ and $\widehat{\mathbf{v}}_{i,t_k}$ are replaced by $\mathbf{x}^c_{i,t_k}$ and $\mathbf{v}^c_{i,t_k}$.

For global kinetic behavior, we use fluid-only mass-normalized specific kinetic energy,

$$e_{\mathrm{kin}}(t_k) = \frac{\sum_{i \in \mathcal{F}} \frac{1}{2} m_i \|\mathbf{v}_{i,t_k}\|_2^2}{\sum_{i \in \mathcal{F}} m_i + \epsilon}. \tag{29}$$

The reported kinetic-energy error is

$$\mathrm{MSE}_{e_{\mathrm{kin}}} = \frac{1}{T} \sum_{k=0}^{T-1} \left(\widehat{e}^c_{\mathrm{kin}}(t_k) - e^r_{\mathrm{kin}}(t_k) \right)^2 . \tag{30}$$

This avoids directly comparing raw total kinetic energy across particle sets with different resolutions.

A.3 Additional Visual Results

Supplementary Figure 3 and 6 show that AnisoLift enriches coarse particles with anisotropic footprints while preserving the fixed particle count. These visualizations complement the quantitative metrics by highlighting local geometry and velocity structure.

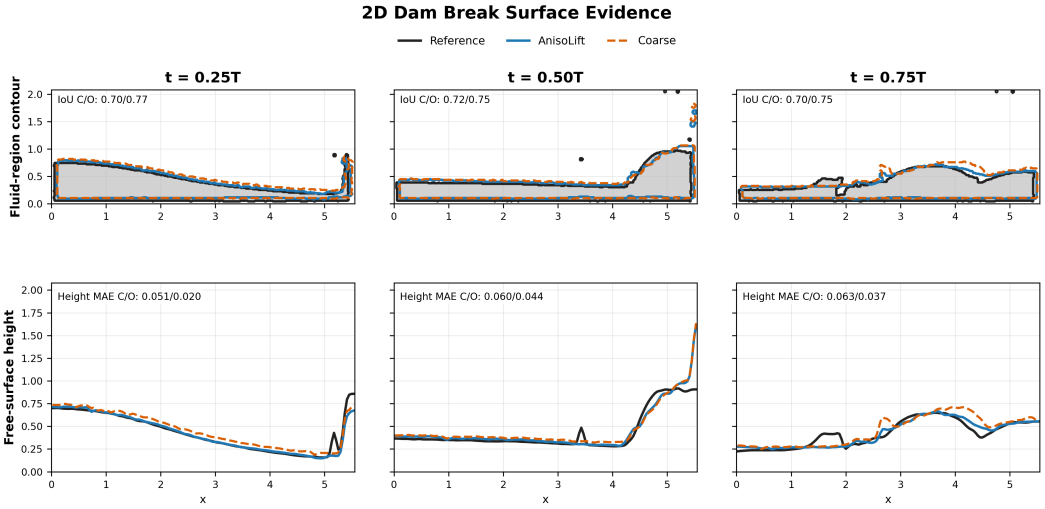

Figure 5: Visualization at surface level. The corrected fluid-region contour and free-surface height profile are consistently closer to the high-resolution reference at both early and late stages.

A.4 Data Generation and Split

Data generation settings, parameters and split are shown in supplementary Table 5 and 6 The data splits are disjoint at the trajectory-file level and vary by benchmark. For Taylor–Green vortex (TGV) and dam break (DAM), we generate multiple independent paired trajectories with different random seeds; the training and test sets consist of entirely separate trajectories, so the model is evaluated on genuinely unseen initial conditions drawn from the same physical regime. Specifically, 2D TGV uses 100/50/50 and 2D DAM uses 50/25/25 trajectories for train/validation/test splits (Table S1). For lid-driven cavity (LDC) and reverse Poiseuille flow (RPF), each benchmark uses a single long solver rollout split temporally into training and test segments; evaluation thus measures generalization to held-out temporal windows within the same boundary conditions and physical parameters. We note that the current evaluation assesses whether a model trained on representative trajectories from a given benchmark regime can improve unseen coarse frames from the same regime. Zero-shot transfer to new boundary conditions, domain geometries, or physical parameters (e.g., different

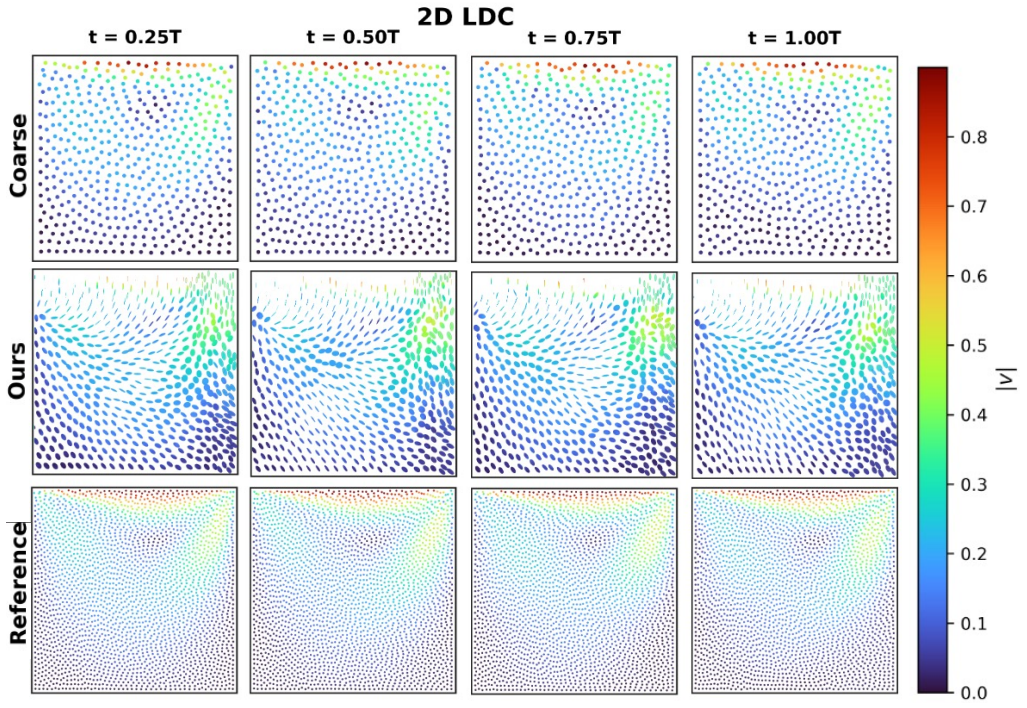

Figure 6: Qualitative correction on 2D lid-driven cavity. Rows compare coarse, AnisoLift, and reference states across normalized time. Particles are colored by velocity magnitude; AnisoLift replaces coarse points with learned anisotropic footprints under the same particle count.

Reynolds numbers) is not evaluated in this work and remains an important direction for future investigation. MPS-generated datasets test whether the same enhancement pipeline remains effective on trajectories from a different particle solver family, with models trained for each solver–scene setting.

Table 5: Dataset-generation settings for paired coarse and high-resolution trajectories. N_c and N_r are reported as fluid/total particles. Splits are train/validation/test trajectory counts.

Dataset	Solver	N_c	N_r	$\Delta x_c / \Delta x_r$	Δt	Write every	Frames	Splits
2D TGV	SPH	484/484	2500/2500	0.04545 / 0.02000	4.0e−4	100	126	100/50/50
2D TGV	MPS	484/484	2500/2500	0.04545 / 0.02000	4.0e−4	100	126	100/50/50
2D DAM	SPH	1250/1622	5000/5740	0.04000 / 0.02000	3.0e−4	100	401	50/25/25
2D DAM	MPS	1250/1622	5000/5740	0.04000 / 0.02000	1.5e−4	200	401	50/25/25
2D LDC	SPH	484/580	2500/2708	0.04545 / 0.02000	4.0e−4	100	10001	1/1/1
2D LDC	MPS	500/580	2500/2708	0.04545 / 0.02000	4.0e−4	100	10001	1/1/1
2D RPF	SPH	800/800	3200/3200	0.05000 / 0.02500	5.0e−4	100	20001	1/1/1
2D RPF	MPS	800/800	3200/3200	0.05000 / 0.02500	5.0e−4	100	10001	1/1/1
3D TGV	SPH	1000/1000	8000/8000	0.62832 / 0.31416	5.0e−3	100	61	200/100/100
3D TGV	MPS	1000/1000	8000/8000	0.62832 / 0.31416	5.0e−3	100	61	200/100/100
3D LDC	SPH	2048/2608	6912/8160	0.06250 / 0.04167	9.0e−4	100	5001	1/1/1
3D LDC	MPS	2048/2608	6912/8160	0.06250 / 0.04167	9.0e−4	100	9999	1/1/1
3D RPF	SPH	1000/1000	8000/8000	0.10000 / 0.05000	1.0e−3	100	5001	1/1/1
3D RPF	MPS	1000/1000	8000/8000	0.10000 / 0.05000	1.0e−3	100	5001	1/1/1

B Analytical comparison

To contextualize AnisoLift relative to these existing approaches, we highlight several key distinctions. First, learned simulators such as GNS (Sanchez-Gonzalez et al., 2020) and Neural SPH (Toshev et al., 2024a) replace or augment the solver's time-stepping rule, producing full rollout trajectories; errors in learned dynamics can compound over long horizons. In contrast, AnisoLift does

not perform rollout: it applies independent corrections to each frame of a pre-existing coarse trajectory, so it cannot introduce rollout instability. Second, coarse-to-high methods like MLFLIP (Um et al., 2018) and Neural UpFlow (Roy et al., 2021) generate additional particles or fine-scale detail, increasing the effective particle count. AnisoLift maintains a fixed particle budget, enriching existing particles with anisotropic latent components rather than introducing new ones. These differences mean that direct numerical comparison requires careful interpretation, as each method class operates under different assumptions about what is fixed (solver, particle count, rollout horizon) and what is learned. We provide quantitative comparisons with Neural SPH in Table 10 to situate our method within the broader landscape.

Table 6: MPS parameters for trajectory generation. All runs use implicit pressure calculation, compressibility $4.5\text{e}{-10}$, CFL 0.3, pressure relaxation coefficient 0.2, collision distance/restitution 0.5/0.2, and radius ratios 3.1/2.1/3.1 for number density, gradient, and Laplacian operators.

Dataset	Boundary / forcing	ν	Gravity / forcing	Δt	Shared init std.
2D TGV	periodic	$1.0\text{e}{-2}$	$(0, 0, 0)$	$4.0\text{e}{-4}$	$5.0\text{e}{-3}$
2D DAM	walls/free surface	$1.0\text{e}{-4}$	$(0, -1, 0)$	$1.5\text{e}{-4}$	$1.0\text{e}{-3}$
2D LDC	lid-driven cavity	$1.0\text{e}{-2}$	$(0, 0, 0)$	$4.0\text{e}{-4}$	$1.0\text{e}{-3}$
2D RPF	periodic reverse-Poiseuille forcing	$1.0\text{e}{-1}$	$(0, 0, 0)$	$5.0\text{e}{-4}$	$1.25\text{e}{-3}$
3D TGV	periodic	$2.0\text{e}{-2}$	$(0, 0, 0)$	$5.0\text{e}{-3}$	$1.5708\text{e}{-2}$
3D LDC	lid-driven cavity	$1.0\text{e}{-2}$	$(0, 0, 0)$	$9.0\text{e}{-4}$	$2.0833\text{e}{-3}$
3D RPF	periodic reverse-Poiseuille forcing	$1.0\text{e}{-1}$	$(0, 0, 0)$	$1.0\text{e}{-3}$	$2.5\text{e}{-3}$

B.1 Training and Inference Algorithms

Algorithm 1 Training Structured Latent Closure

Input: Paired sequences $\{(\mathcal{S}^c_{t_k}, \mathcal{S}^r_{t_k})\}_{k=0}^{T-1}$, epochs E, parameters θ
Output: Best parameters $\theta^\star$ and residual normalization statistics

Preprocess: align $\mathcal{S}^r$ to the coarse time grid. For each coarse fluid particle $i \in \mathcal{F}^c$, aggregate nearby reference particles to obtain $(\mathbf{x}^*_{i,t_k}, \mathbf{v}^*_{i,t_k}, \mathbf{C}^*_{i,t_k})$ and residual targets $\Delta\mathbf{x}^*_{i,t_k}$, $\Delta\mathbf{v}^*_{i,t_k}$ Estimate residual normalization statistics from the training split Initialize θ, optimizer state, and best validation score $s^\star \leftarrow +\infty$ **for** $e = 1$ **to** E **do**
  foreach *mini-batch* $\mathcal{B}$ **do**
    Extract features $\mathbf{h}_{i,t_k} = \phi_\theta(\mathcal{S}^c_{t_k}, i)$ for $i \in \mathcal{F}^c$ Predict geometry and normalized residuals: $\mathbf{C}_{i,t_k} = g^{\text{geo}}_\theta(\mathbf{h}_{i,t_k})$, $(\bar{\Delta}\mathbf{x}_{i,t_k}, \bar{\Delta}\mathbf{v}_{i,t_k}) = g^{\text{res}}_\theta(\mathbf{h}_{i,t_k})$ Denormalize residuals and form corrected states $(\widehat{\mathbf{x}}_{i,t_k}, \widehat{\mathbf{v}}_{i,t_k})$ Compute $\mathcal{L}_x$, $\mathcal{L}_v$, $\mathcal{L}_{\text{ekin}}$, and $\mathcal{L}_{\text{geo}}$; update θ by minimizing $\mathcal{L}$
  Evaluate validation score s_e and update $\theta^\star$ if $s_e < s^\star$
return $\theta^\star$ *and normalization statistics*

Algorithm 2 Inference with Structured Latent Closure

Input: Coarse sequence $\{\mathcal{S}^c_{t_k}\}_{k=0}^{T-1}$, trained parameters $\theta^\star$, residual normalization statistics
Output: Corrected sequence $\{\widehat{\mathcal{S}}_{t_k}\}_{k=0}^{T-1}$ and anisotropic geometry $\{\mathbf{C}_{i,t_k}\}$

foreach *coarse frame* t_k **do**
  foreach *coarse fluid particle* $i \in \mathcal{F}^c$ **do**
    Extract $\mathbf{h}_{i,t_k} = \phi_{\theta^\star}(\mathcal{S}^c_{t_k}, i)$ Predict $\mathbf{C}_{i,t_k}$ and normalized residuals $(\bar{\Delta}\mathbf{x}_{i,t_k}, \bar{\Delta}\mathbf{v}_{i,t_k})$ Denormalize residuals and compute $(\widehat{\mathbf{x}}_{i,t_k}, \widehat{\mathbf{v}}_{i,t_k})$
  Pass non-fluid particles through unchanged and form $\widehat{\mathcal{S}}_{t_k}$
return $\{\widehat{\mathcal{S}}_{t_k}\}_{k=0}^{T-1}$ *and* $\{\mathbf{C}_{i,t_k}\}$

B.2 Runtime analysis

Since AnisoLift applies a lightweight two-layer MLP (hidden dimension 64) as a post-processing step on precomputed coarse trajectories, its inference overhead is minimal. Table 8 reports the

Table 7: SPH numerical parameters for synthetic trajectory generation. SPH uses the QSK kernel with h-factor 1.0 and CFL 0.25.

Dataset	Boundary / forcing	ν	$\|\mathbf{g}\|$ or forcing	Density evolution	Artificial α	p_{bg} factor	Kernel
2D TGV	periodic	1.0e−2	0	false	0.0	0.0	QSK
2D DAM	walls/free surface	5.0e−5	1.0	true	0.1	0.0	QSK
2D LDC	lid-driven cavity	1.0e−2	0	false	0.0	0.01	QSK
2D RPF	periodic reverse-Poiseuille forcing	1.0e−1	1.0	false	0.0	0.05	QSK
3D TGV	periodic	2.0e−2	0	false	0.0	0.0	QSK
3D LDC	lid-driven cavity	1.0e−2	0	false	0.0	0.01	QSK
3D RPF	periodic reverse-Poiseuille forcing	1.0e−1	1.0	false	0.0	0.02	QSK

wall-clock time per frame for the coarse solver, the high-resolution reference solver, and AnisoLift inference on a single NVIDIA H200 GPU. The inference cost of AnisoLift is dominated by

Table 8: Wall-clock time per frame. AnisoLift inference adds negligible overhead ($< 10\,\mu$s per frame) compared to the coarse solver, while avoiding the cost of high-resolution simulation. Solver times are measured with JAX-SPH on a single NVIDIA H200 GPU; AnisoLift inference time is the per-frame MLP forward pass averaged over 200 repetitions with batch size 128.

Dataset	N_c / N_h (particles)	Solver time / frame (s) Coarse	High-res	AnisoLift inference	Speedup vs. High-res
3D TGV (SPH)	1000 / 8000	0.593	2.482	6.3 μs	4.18×

the feature extraction step (computing per-particle input features from the coarse state), while the MLP forward pass itself is negligible. Importantly, the coarse solver runs independently and is not modified; AnisoLift inference can be applied selectively to frames of interest without re-running the simulation. This decoupled design means the additional inference stage preserves the efficiency advantage of coarse simulation: the total cost of coarse simulation plus AnisoLift post-processing remains substantially lower than running the high-resolution solver directly.

B.3 Training Hyperparameters

We list detailed hyperprameters applied in the training process of the main experiments in Table 9.

Table 9: Training hyperparameters used in the main experiments.

Solver	Dataset	Epochs	Batch	LR	Clip
SPH	2D TGV / DAM / LDC / RPF	30	32	3.0e−4	1.0
SPH	3D TGV	40	32	1.0e−4	0.5
SPH	3D LDC	30	32	3.0e−4	1.0
SPH	3D RPF	10	32	2.0e−4	0.5
MPS	2D DAM / LDC / RPF	30	32	3.0e−4	1.0
MPS	2D TGV	40	32	3.0e−4	1.0
MPS	3D TGV	40	32	1.0e−4	0.5
MPS	3D LDC / RPF	30	32	3.0e−4	1.0

C Comparison with Learned Particle Simulators

AnisoLift is a resolution enhancement method that post-processes coarse trajectories, placing it in a different methodological category from learned simulators such as GNS (Sanchez-Gonzalez et al., 2020) that replace the solver and produce rollouts at the original resolution. To contextualize the improvements of AnisoLift, we compare against GNS (Sanchez-Gonzalez et al., 2020), a graph-network-based learned particle simulator trained on the LagrangeBench benchmark (Toshev et al., 2023) in the Table 10. Unlike AnisoLift, which post-processes precomputed coarse trajectories, GNS produces its own rollout by learning the dynamics directly at the original particle resolution. We also evaluate Neural SPH (Toshev et al., 2024a), which augments the GNS architecture with

physics-informed relaxation steps. However, with publicly available configurations, it did not improve over the plain GNS baseline on our test benchmarks. This fundamental difference in methodology means the two method classes are not drop-in replacements for each other.The following comparison reveals complementary strengths. AnisoLift achieved comparable performance with a significantly lower particle counts as shown in Table 10

Table 10: Quantitative results for GNS and AnisoLift on overlapping benchmarks. AnisoLift corrects precomputed coarse solver trajectories at the coarse particle count (N_c). MSE values are not directly comparable to GNS due to the protocol differences. MSE_v for GNS is computed post-hoc via finite differencing of predicted positions.

Panel B: AnisoLift — Current-Frame Correction on Paired Data					
Benchmark	Method	N	$\text{MSE}_x \downarrow$	$\text{MSE}_v \downarrow$	E_{kin} MSE $\downarrow$
2D DAM	Coarse SPH	1622	2.560	35.92	0.250
	GNS	5740	0.264	84.70	0.077
	AnisoLift	1622	1.100	22.34	0.110
	AnisoLift	2800	0.280	23.49	0.090
3D TGV	Coarse SPH	1000	0.0519	21.26	0.0397
	GNS	8000	0.0476	6.01	0.0349
	AnisoLift	1000	0.0517	12.54	0.0012
	AnisoLift	4000	0.0483	6.51	0.0112

D LIMITATION AND FUTURE WORK

Limitations. The main limitations of the current work are frame-wise independence without temporal regularisation, evaluation metrics tied to the training targets, and memorisation risk from absolute-position input on quasi-steady benchmarks.

Future works In our future work, temporal consistency can be enforced by a sliding-window input or a lightweight recurrent head, both compatible with the current architecture at minimal additional cost. Target-independent evaluation can be achieved by reconstructing density and velocity fields from the corrected particles on a shared regular grid, computing Chamfer distance between corrected and reference positions, or extending the dam-break surface diagnostics to all benchmarks in 3D. Cross-regime transfer can be pursued via conditioning on physical parameters or few-shot fine-tuning, neither of which requires architectural changes. These extensions can be pursued incrementally without redesigning the core framework.