WHAT DOES EXACT GLOBAL MASS BALANCE IMPLY IN OPEN-SYSTEM NEURAL PDE SURROGATES?

Anonymous authors
Paper under double-blind review

ABSTRACT

Exact global mass balance constrains one scalar property of a predicted field: its total mass. In an open system, sources and boundary exchanges change the required total mass from one step to the next. We develop a diagnostic framework to examine how specification of this total separates from spatial prediction error in finite-volume neural PDE surrogates. The results show that global mass balance must be assessed through both target specification and the field error that remains. A misspecified target can be satisfied to numerical precision. Even when the target is correct, the mass-changing error fraction varies widely across predictions, and fields satisfying this target can still differ substantially in spatial accuracy. Furthermore, omitting prescribed external mass increments can severely degrade autoregressive rollout stability. These distinctions appear across different mesh geometries and model constructions. Thus, a small absolute constraint residual only establishes agreement with the supplied target mass, without showing that the target was specified correctly or that the predicted field is accurate. Therefore, evaluating conservation-aware surrogates requires reporting how the supplied target mass is specified, the absolute constraint residual, spatial error, and rollout behavior together.

1 INTRODUCTION

Neural PDE surrogates can satisfy global mass balance through conservative updates, hard constraints, or output projection (Horie & Mitsume, 2024; Hansen et al., 2023; Chen et al., 2021; Valente et al., 2025; Baez et al., 2025). While this constraint preserves the total mass of the predicted field, fields with the same total can differ spatially. Therefore, exact balance alone does not determine prediction accuracy. In open systems, sources and boundary exchanges actively change the required total mass. Thus, enforcing a constant total mass in an open system with a non-zero external mass increment supplies the model with a misspecified target mass. Furthermore, regardless of whether this supplied target is correct, a prediction may remain unconstrained by it, be constrained to it during forward evaluation, or be projected to it afterward.

For finite-volume cell-average states, physical mass is the state sum weighted by control-volume measure (LeVeque, 2002; Horie & Mitsume, 2024). On an equal-volume grid, weighted mass is proportional to the unweighted state sum. Keeping that sum constant achieves the correct global balance only when the net external mass increment over the step is zero. Thus, nonuniform grids require volume weighting, and an open-system mass target must also include the prescribed source and boundary mass increments over the step (LeVeque, 2002; Hansen et al., 2023; Baez et al., 2025). Specifying that target and satisfying it exactly are therefore distinct tasks.

A matched reference allows us to measure the share of a prediction's field error associated with total-mass mismatch. When total mass and squared field error use the same control-volume weighting, standard projection geometry splits the error into orthogonal components that change or preserve mass (Panagiotelis et al., 2021; Valente et al., 2025). Projection to the reference mass removes the former while leaving the latter unchanged. The fraction of weighted squared error removed by this projection measures how much correcting only the total mass improves that specific prediction.

However, a reduction in field error after correction does not validate the target specification. A misspecified target mass can still reduce error when it is closer to the reference mass than the prediction's original target mass, although an error floor remains due to the target offset. Even

a target equal to the reference mass can leave spatial error that preserves mass. Furthermore, in autoregressive prediction, corrected states become subsequent inputs, meaning repeated correction produces recursive effects beyond the projection of a single step.

To measure these distinctions and evaluate the prediction performance of existing PDE surrogates in open systems, this study makes the following contributions:

- We construct controlled evaluations that hold the source input fixed while varying whether its mass increment enters the supplied target mass, demonstrating that a zero constraint residual does not guarantee a physically correct specification.
- For held-out predictions from trained surrogates, we apply volume-weighted projection geometry to isolate the exact fraction of weighted squared error associated with total-mass mismatch as opposed to same-mass spatial redistribution.
- We decouple the static removal of mass-changing error from the recursive effects of projecting autoregressive rollouts over time.

These measurements establish an effective diagnostic framework to distinguish the information given by a mass-balance residual from the physical evidence required to assess spatial prediction.

2 Open-System Mass Balance and Error Decomposition

2.1 Specifying the open-system target mass

For the scalar field ϕ whose control-volume integral defines the tracked mass, the continuous open-system balance can be written as

$$\frac{\partial \phi}{\partial \tau} + \nabla \cdot \mathbf{F} = s,$$

where τ is continuous time, $\mathbf{F}$ is the flux, and s is a prescribed volumetric source. Let Ω be the modeled domain and $\partial\Omega$ its boundary. Let $\mathbf{n}$ be the outward unit normal on the boundary, dV denote the differential measure over the domain (length, area, or volume according to dimension), and dS denote the differential measure over the boundary. Integration over Ω gives

$$\frac{d}{d\tau} \int_\Omega \phi \, dV = \int_\Omega s \, dV - \int_{\partial\Omega} \mathbf{F} \cdot \mathbf{n} \, dS.$$

Thus, the domain mass changes only through prescribed source and boundary exchange (LeVeque, 2002).

Let t be the integer index for discrete time steps used for training and evaluation. Let ϕ_i^t be the cell-average intensive state in finite-volume cell i, and let $V_i > 0$ be its control-volume measure. The cell contributes mass $V_i \phi_i^t$, so the domain mass is $\sum_i V_i \phi_i^t$. On a nonuniform mesh, the same internal mass transfer can produce cell-average state changes of different magnitudes, which is why conservation is expressed through this V_i-weighted mass rather than an unweighted state sum. Let Δt be the constant elapsed physical time from step t to step $t+1$, and let $s_i(\tau)$ be the cell-average volumetric source rate. This interval need not equal an internal time step of the reference solver. We define the signed boundary contribution over that same interval as

$$\Delta M_\partial^t := -\int_{t\Delta t}^{(t+1)\Delta t} \int_{\partial\Omega} \mathbf{F} \cdot \mathbf{n} \, dS \, d\tau,$$

so that inflow is positive and outflow is negative. The correctly specified next-step mass constraint is then

$$\sum_i V_i \phi_i^{t+1} = \sum_i V_i \phi_i^t + \sum_i V_i \int_{t\Delta t}^{(t+1)\Delta t} s_i(\tau) \, d\tau + \Delta M_\partial^t. \tag{1}$$

If the source is constant from step t to step $t+1$, its contribution reduces to $\Delta t \sum_i V_i s_i^t$. In a closed, unforced periodic system, both external increments vanish (LeVeque, 2002; Hansen et al., 2023; Baez et al., 2025).

To use the same control-volume weighting for the error geometry, we define

$$\langle x, y \rangle_V := \sum_i V_i x_i y_i$$

for cellwise vectors x and y, and we let $\mathbf{1}$ denote the all-ones vector. We also write $s(\tau) = (s_i(\tau))_i$ for the cellwise source-rate vector. The finite-volume mass is then $\langle \mathbf{1}, \phi^t \rangle_V$, and Eq. (1) can equivalently be written as

$$\langle \mathbf{1}, \phi^{t+1} \rangle_V = \langle \mathbf{1}, \phi^t \rangle_V + \int_{t\Delta t}^{(t+1)\Delta t} \langle \mathbf{1}, s(\tau) \rangle_V \, d\tau + \Delta M_\partial^t.$$

For a linear equality, Hansen et al. (2023) write the constraint as $Gu = b$ and call b the *constraint value*. Here, we use b to indicate the supplied target mass for a prediction step. Under correct specification, b equals the required next-step total mass on the right-hand side of Eq. (1). Thus, omitting either increment can yield a misspecified target mass that can nevertheless be satisfied exactly. For a prediction $\hat{\phi}^{t+1}$, the constraint residual is equivalently $\sum_i V_i \hat{\phi}_i^{t+1} - b$, where $\sum_i V_i \hat{\phi}_i^{t+1} = \langle \mathbf{1}, \hat{\phi}^{t+1} \rangle_V$. Therefore, a zero constraint residual implies exact satisfaction of the supplied target mass b, but does not guarantee that this target was specified correctly. This is because a prescribed source has two distinct roles. Its supplied values inform the predictor's spatial prediction, whereas its integrated mass increment $\sum_i V_i \int_{t\Delta t}^{(t+1)\Delta t} s_i(\tau) \, d\tau$ changes the required next-step total mass. Providing the first does not guarantee the second. That is, the predictor can receive the source while the supplied target mass omits its mass increment.

2.2 Mass-changing and mass-preserving errors

At one fixed evaluation step, let ϕ be the reference field, let $\hat{\phi}$ be a fixed prediction before any optional mass projection, and let $e = \hat{\phi} - \phi$ be their realized error. The reference mass is $\langle \mathbf{1}, \phi \rangle_V$, and the original prediction's total mass is $\langle \mathbf{1}, \hat{\phi} \rangle_V$. Their total-mass mismatch is $\langle \mathbf{1}, \hat{\phi} \rangle_V - \langle \mathbf{1}, \phi \rangle_V = \sum_i V_i e_i = \langle \mathbf{1}, e \rangle_V$. Because total mass is the linear functional defined by the inner product with $\mathbf{1}$, all fields with the same total mass as the reference form an affine hyperplane whose normal direction under this weighted geometry is $\mathbf{1}$.

Using the compact vector notation, the error decomposition is

$$e_\perp = \frac{\langle \mathbf{1}, e \rangle_V}{\langle \mathbf{1}, \mathbf{1} \rangle_V} \mathbf{1}, \qquad e_\parallel = e - e_\perp, \tag{2}$$

with cellwise elements $e_{\perp,i} = \langle \mathbf{1}, e \rangle_V / \langle \mathbf{1}, \mathbf{1} \rangle_V$ and $e_{\parallel,i} = e_i - e_{\perp,i}$. Because $\langle \mathbf{1}, \mathbf{1} \rangle_V = \sum_i V_i$, this matches the cellwise formulation. The normal component $e_\perp$ contains the entire mass mismatch, while $e_\parallel$ is parallel to the constant-mass hyperplane. By construction, $\sum_i V_i e_{\parallel,i} = \langle \mathbf{1}, e_\parallel \rangle_V = 0$ and $\sum_i V_i e_{\perp,i} e_{\parallel,i} = \langle e_\perp, e_\parallel \rangle_V = 0$.

The *weighted squared error* is $\sum_i V_i e_i^2 = \langle e, e \rangle_V$, which is the finite-volume quadrature of the squared L_2 field error. Its Pythagorean decomposition is

$$\langle e, e \rangle_V = \langle e_\perp, e_\perp \rangle_V + \langle e_\parallel, e_\parallel \rangle_V = \frac{\langle \mathbf{1}, e \rangle_V^2}{\langle \mathbf{1}, \mathbf{1} \rangle_V} + \langle e_\parallel, e_\parallel \rangle_V. \tag{3}$$

We call the first term $\langle e_\perp, e_\perp \rangle_V$ the *mass-changing error* and the second term $\langle e_\parallel, e_\parallel \rangle_V$ the *mass-preserving error*. Thus, the weighted squared error is their sum. The mass-changing error is determined entirely by total-mass mismatch, while the mass-preserving error can remain when the prediction and reference have the same total mass but distribute that mass differently across cells.

Volume-weighted orthogonal projection to the reference mass subtracts exactly $e_\perp$ and leaves $e_\parallel$ unchanged (Panagiotelis et al., 2021; Valente et al., 2025). Hence, when the weighted squared error is nonzero, we define the *mass-changing error fraction* as

$$\frac{\langle \mathbf{1}, e \rangle_V^2 / \langle \mathbf{1}, \mathbf{1} \rangle_V}{\langle e, e \rangle_V}.$$

Under this weighted geometry, the mass-changing error fraction is exactly the fraction of weighted squared error removed by projection to the reference mass. Its complement is the mass-preserving error fraction.

2.3 Misspecified target masses and error floors

We now apply the same volume-weighted orthogonal projection to the supplied target mass b, which need not equal the reference mass. The weighted squared error after projection has the form

$$\langle e_{\|}, e_{\|} \rangle_V + \frac{(b - \langle \mathbf{1}, \phi \rangle_V)^2}{\langle \mathbf{1}, \mathbf{1} \rangle_V}.$$

Thus, projection removes the prediction's original mass-changing component, but a mismatch between the supplied target mass b and the reference mass remains as a new mass-changing error. This second term is the target-induced field-error floor for any field satisfying the supplied target mass b.

Before projection, the mass-changing error is $\langle \mathbf{1}, e \rangle_V^2 / \langle \mathbf{1}, \mathbf{1} \rangle_V$. Thus, a misspecified supplied target mass can still reduce field error when $|b - \langle \mathbf{1}, \phi \rangle_V| < |\langle \mathbf{1}, e \rangle_V|$. A prediction can consequently have both lower field error after correction and a zero constraint residual while the supplied target mass remains misspecified. A lower field error after correction therefore does not show that the supplied target mass was specified correctly. When the supplied target mass equals the reference mass, i.e., $b = \langle \mathbf{1}, \phi \rangle_V$, this floor vanishes, but the mass-preserving spatial error in Eq. (3) can remain.

This decomposition applies to a single fixed prediction. Training with a mechanism that enforces the mass constraint can change $e_{\|}$ itself, and in an autoregressive rollout, a corrected state changes the next model input. Therefore, mass-preserving spatial error and recursive rollout effects are evaluated separately in the following sections.

3 Controlled Evaluation Design

We measure the absolute constraint residual, the field-error quantities, and autoregressive accuracy. The comparisons separate changes to the supplied target mass from changes to spatial prediction. Projection of a completed prediction and repeated projection during rollout are evaluated separately because only the latter changes subsequent inputs.

3.1 Measured quantities

A rollout starts at step t from the reference state ϕ^t. Let $h \geq 1$ be the evaluation horizon and write $e_i^{t+h} = \hat{\phi}_i^{t+h} - \phi_i^{t+h}$. We apply the error decomposition at each evaluated step, using the same V_i, weighted inner product, and components $e_{\perp}$ and $e_{\|}$, as explained in Section 2. Specifically, at each horizon, we apply the definitions of the *weighted squared field error*, *mass-changing error*, *mass-preserving error*, and *mass-changing error fraction* to e^{t+h}.

The absolute constraint residual $|\langle \mathbf{1}, \hat{\phi}^{t+h} \rangle_V - b^{t+h}|$ must be distinguished from the absolute difference between the predicted and reference total mass $|\langle \mathbf{1}, e^{t+h} \rangle_V|$, where b^{t+h} is the supplied target mass for the evaluated step $t + h$. The supplied target mass may change at every step and need not equal the reference mass.

For graded-grid spatial and rollout comparisons, we evaluate the volume-weighted relative-L_1 field error. For source-increment comparisons, we also use the cellwise root-mean-square error (RMSE), which weights cells equally. These are given respectively by

$$\text{Relative-}L_1 = \frac{\sum_i V_i |e_i^{t+h}|}{\sum_i V_i |\phi_i^{t+h}|}, \qquad \text{RMSE} = \left[\frac{1}{N} \sum_i (e_i^{t+h})^2 \right]^{1/2},$$

where N is the number of cells. This RMSE weights cells equally.

These quantities separate three evaluation questions that the constraint residual alone does not resolve. They allow us to evaluate the target specification, the mass-changing share of completed-prediction error, and the effect of feeding corrected states back during rollout.

3.2 Output parameterizations

Four parameterizations produce internal updates with zero net mass change under the control-volume weighting. Face-FV uses signed transfers across shared faces (LeVeque, 2002). Antisymmetric forms

pairwise transfers from directed scores using the cancellation principle of locally conservative graph-flux constructions (Horie & Mitsume, 2024). Density-relative distributes bounded donor outflow among neighboring cells in a FluxNet-style construction (Lan et al., 2026). Free projected removes the volume-weighted mean from an unconstrained cellwise update (Chen et al., 2021; Panagiotelis et al., 2021; Valente et al., 2025). The current-state mass and the prescribed external mass increments together determine the supplied target mass through Eq. (1). The four parameterizations satisfy the same supplied target mass while redistributing mass differently. Fixing this scalar target exposes spatial and rollout differences that target satisfaction does not distinguish. Appendix D gives their definitions and relation to the cited architectures.

3.3 PRIMARY COMPARISONS

Target specification with a prescribed source. A periodic 16×16 finite-volume advection–diffusion system compares two conditions with the same source input and MeshGraphNets-style encoder–processor backbone. Only whether the prescribed source mass increment is included in the supplied target mass changes. This separates giving the predictor the source field from using its prescribed mass increment to form the supplied target mass. Per-axis largest-to-smallest cell-width ratios are $1, 2, 4,$ and 6, with eight paired training seeds. Each grade has 24 training, 8 validation, and 8 held-out trajectories, with checkpoints selected by validation loss. A separate source-plus-boundary control includes both external increments.

Completed-prediction error decomposition. On a regular 64-cell grid, validation-selected FluxNet and FluxGNN checkpoints are evaluated on new held-out trajectories using the original-author predictor and decoder implementations (Lan et al., 2026; Horie & Mitsume, 2024). Each family has three training seeds and eight trajectories in each of two forcing conditions. Both conditions include a prescribed source, and one also includes boundary exchange. We apply the Section 2 decomposition to each completed prediction before any correction. At horizons $h = 1, 8, 16, 32, 64,$ and 96, each seed is summarized by the median over its 16 condition–trajectory predictions. We then report the median and minimum–maximum range across the three seeds. The 48 prediction-level observations are nested within these three training seeds. This comparison measures how much realized field error lies in the mass-changing direction under the stated weighted geometry. It does not include feedback from corrected states.

Spatial and rollout errors under the same supplied target mass. A source-driven nonuniform 16×16 system has a per-axis cell-width ratio of 5.07. The four output parameterizations share a periodic CNN backbone, training data, state-prediction objective, and 1,200-update budget, with five independent training seeds per parameterization. The split has 32 training, 8 validation, and 16 held-out trajectories, each with 40 stored states. All four use the same prescribed source and supplied target mass. We measure one-step spatial error and 39-step autoregressive error. Holding the supplied target mass fixed tests field differences that the scalar mass constraint does not distinguish. A pre-training control measures the constraint residual and field error before optimization.

Recursive projection. For the same 64-cell predictors, each decoder output is projected to the supplied target mass determined by its input state and prescribed external mass increments. The projected state becomes the next input. Predictor and decoder weights remain fixed, so later-horizon differences include the effect of earlier corrections on subsequent predictions. We therefore evaluate this separately from the completed-prediction decomposition, where correction does not change later inputs.

3.4 SECONDARY COMPARISONS

The 64-cell post-decoder comparison keeps the trained predictor and decoder fixed while varying six output rules: projection only, a learned free update, a free update followed by projection, Antisymmetric, Density-relative, and Face-FV. This provides a descriptive control of post-decoder output construction with predictor and decoder weights held fixed. The comparison was specified after the initial held-out inspection and remains secondary (Appendix C.2).

The irregular-mesh evaluation uses the 4,299-cell Davis mesh from the competition-format Model 2 data distributed for the January 2026 UrbanFloodBench Kaggle challenge (Lim et al., 2026a;c). Prescribed rainfall and signed reference-solver manhole exchange are supplied to local MeshGraphNets-

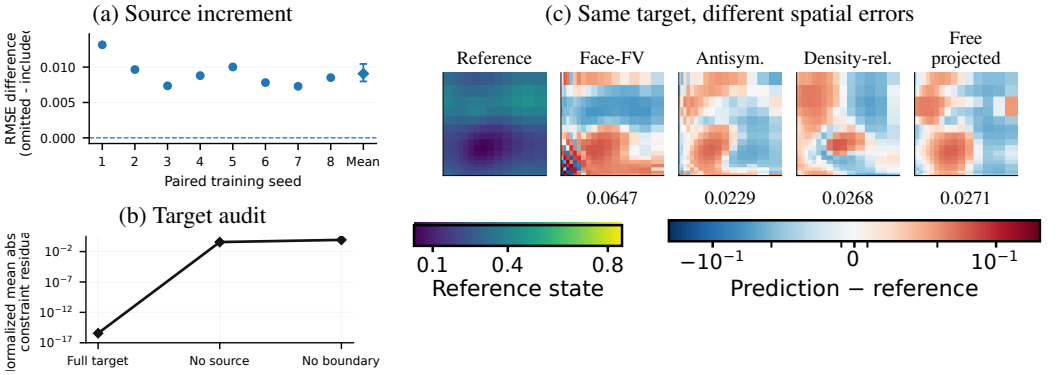


Figure 1: Target specification and same-target fields; values in (c) are volume-weighted relative-L_1 errors.

style, FluxGNN-style, and HydroGraphNet-style predictors, each trained with three training seeds (Pfaff et al., 2021; Horie & Mitsume, 2024; Taghizadeh et al., 2025). These are local benchmark adaptations. The original-author implementations are used in the 64-cell evaluation above. Flood evaluation uses seven internal held-out events from the competition training-side pool and excludes the official 30-event competition test split. The later full UrbanFloodBench release is not the source of this evaluation. Because signed manhole exchange is supplied from the reference solver, this is a conditional irregular-mesh evaluation under prescribed external increments. Appendix C.5 gives the output conditions, split, and adaptation details.

4 Results

4.1 Target specification with a prescribed source

For all eight paired training seeds, including the prescribed source mass increment in the supplied target mass lowers the 32-step horizon-mean cellwise RMSE (Figure 1(a)). The mean omitted-minus-included difference is 0.009074, with a 95% paired-bootstrap interval of $[0.007978, 0.010448]$. The exact sign-flip test gives $p = 0.0391$ after Holm correction over five planned tests (Holm, 1979). With the same weighting over four mesh grades and eight seeds, the RMSE is 0.05588 when the increment is omitted and 0.04680 when it is included. Both matched conditions use the same source field and training setup and differ only in whether the prescribed mass increment enters the supplied target mass, so this comparison directly tests target specification during training and rollout.

Figure 1(b) audits target construction for the same stored rollout from the structural output. The normalized mean absolute constraint residual is near numerical precision when both source and boundary contributions are included in the supplied target mass, but increases by many orders of magnitude when either contribution is omitted from the evaluated target. The constraint residual therefore measures agreement with the supplied target mass and must be distinguished from whether that target was constructed correctly. The two prespecified mesh-grading checks did not support the expected monotonic mechanism: the slope has the opposite sign from the expected direction, and the uniform-grid negative control also differs from expectation. We therefore do not infer a mesh-irregularity mechanism from the paired source result. A pre-training control in Appendix C further shows that an output rule can satisfy the supplied target mass to numerical precision while its weighted squared error remains large.

4.2 Completed-prediction error decomposition

For completed predictions, the mass-changing error fraction varies strongly across predictor and horizon (Figure 2). FluxNet's cross-seed median rises from 5.78% at horizon 1 to 16.7% at horizon 96, whereas the FluxGNN horizon-96 median is 0.00912%. At horizon 96, the 48 FluxNet predictions span 0.108%–76.5%, while the FluxGNN maximum is 0.0366% and most of its weighted squared error remains mass preserving. These prediction-level observations are nested within three training seeds, so family differences remain descriptive.

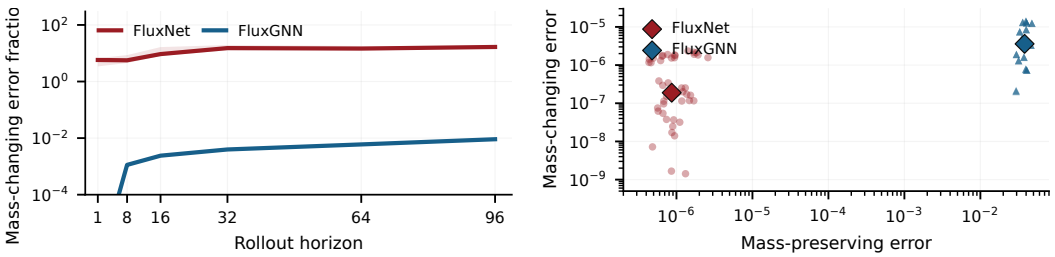


Figure 2: Completed-prediction decomposition: mass-changing fraction (a) and horizon-96 error components (b).

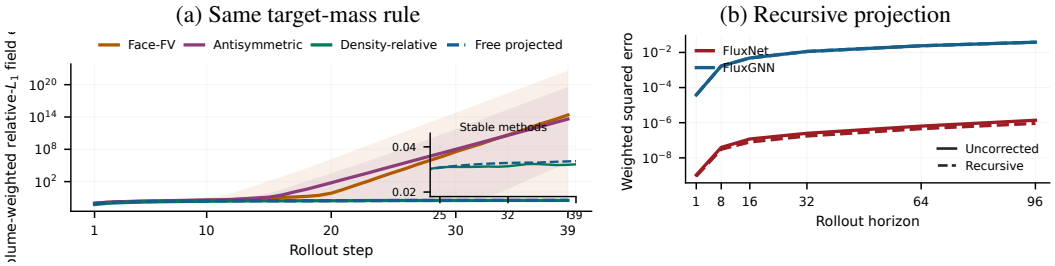


Figure 3: Same-target and recursively projected rollout errors.

Mass-preserving and mass-changing errors sum to the weighted squared error, so their fraction is not a field-accuracy score and depends on the error geometry. In the nonuniform-grid sensitivity check, replacing control-volume weighting with Euclidean weighting changes the seed-level median from 29.1% to 13.1% for Face-FV and from 48.0% to 33.0% for Free projected (Appendix D).

4.3 Spatial and rollout errors under the same supplied target mass

The four graded-grid parameterizations share the same backbone, data, objective, training budget, and supplied target mass while constructing spatial updates differently. Their one-step cross-seed median volume-weighted relative-L_1 errors range from 0.00291 to 0.00457. In the common held-out case, all four predictions match the reference mass to numerical precision but have different spatial error maps (Figure 1(c)); Appendix C shows the same case at horizons 1, 8, and 16.

The difference widens in rollout (Table 1; Figure 3a). All five Antisymmetric seeds and three of five Face-FV seeds become unstable, with step-39 relative-L_1 error reaching 3.772×10^{19} and 3.806×10^{22}, respectively, while the five-seed maxima are 0.0361 for Density-relative and 0.0429 for Free projected. A common target mass therefore does not determine spatial accuracy or autoregressive stability. This is a coexistence result rather than a causal ranking; capacity and optimization also differ, and the runaway constraint residuals were not independently recomputed.

Table 1: Graded-grid relative-L_1 error over five seeds (volume weighted; mean $\pm$ s.d.).

Output	One-step	Rollout mean	Step 39
Face-FV	$0.00438 \pm 5.96 \times 10^{-4}$	$2.28 \times 10^{20} \pm 5.09 \times 10^{20}$	$7.61 \times 10^{21} \pm 1.70 \times 10^{22}$
Antisymmetric	$0.00465 \pm 3.43 \times 10^{-4}$	$3.14 \times 10^{17} \pm 4.90 \times 10^{17}$	$1.06 \times 10^{19} \pm 1.65 \times 10^{19}$
Density-relative	$0.00388 \pm 2.47 \times 10^{-4}$	0.0278 ± 0.00317	0.0315 ± 0.00398
Free projected	$0.00328 \pm 8.36 \times 10^{-4}$	0.0308 ± 0.00771	0.0340 ± 0.00787

4.4 Recursive correction changes later predictions

Figure 3b directly compares weighted squared error for the uncorrected and recursively projected rollouts of the same 64-cell predictors. Recursive projection changes the state passed to later steps. At horizon 96, FluxNet's median weighted squared error decreases from 1.38×10^{-6} in the uncorrected rollout to 9.14×10^{-7} after recursive projection, while both FluxGNN rollouts are about 3.82×10^{-2}.

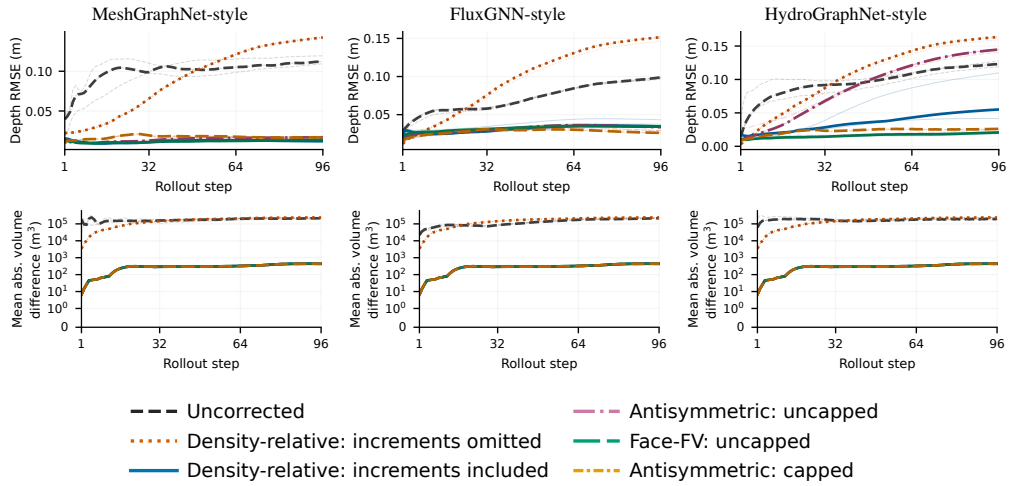


Figure 4: Flood rollout metrics: depth RMSE (top) and absolute prediction-reference volume difference (bottom).

Fixed completed-prediction decomposition and recursive rollout therefore measure different effects. In the secondary comparison with predictor and decoder weights fixed, the FluxGNN horizon-96 error is 7.52×10^{-6} with the Face-FV rule and 3.82×10^{-2} for the uncorrected output (Appendix C.2). This comparison was specified after held-out inspection and is treated as descriptive secondary evidence.

4.5 Irregular-mesh flood secondary check

The secondary irregular-mesh study evaluates a conditional surface-water surrogate supplied with prescribed rainfall and signed reference-solver manhole exchange. Including the prescribed increments in the Density-relative target lowers pooled depth RMSE in all nine backbone-seed pairs. Mass-constrained outputs that include these increments satisfy the supplied target mass to numerical precision, yet their mean absolute prediction-reference total-water-volume difference at the common step 96 is about $438\,\mathrm{m}^3$. In Figure 4, the lower row reports this matched-reference quantity rather than the target-based constraint residual. Target satisfaction, reference-total agreement, and field error therefore remain separate even for the same prediction. Appendix C.5 reports uncertainty and fixed spatial fields for the seven internal held-out events.

5 Conclusion

Exact global mass balance has a narrower meaning than predictive accuracy. It merely establishes agreement between the predicted total mass and the supplied target mass. The controlled source comparison shows why this distinction matters in open systems. The predictor can receive the same source input while the supplied target omits its prescribed mass increment, which can lead to increasing rollout error. Even when the supplied target is correct, satisfying the target does not determine the spatial field. One-step predictions satisfying the same target retain different spatial errors, and the mass-changing error fraction shows that correcting only the total would remove very different shares of weighted squared error across completed predictions. Recursive projection introduces further dynamics because each correction changes the input to the next prediction.

The supplied target mass should first be checked against the prescribed source and boundary mass increments. The residual then shows how closely the predicted total matches that supplied target. With a matched reference, the predicted and reference total masses can be compared directly, and the mass-changing error fraction gives the share of weighted squared error associated with that difference. Spatial error and autoregressive rollout behavior must also be measured directly. Finally, multiple constraints, positivity, and other local admissibility conditions require separate treatment.

AI USE STATEMENT

Under the authors' direction, generative AI tools assisted translation, readability editing, manuscript organization and rearrangement, literature retrieval, and critical checks of derivations, experimental methodology, quantitative claims, and interpretations. They also assisted in writing and refining analysis code, tables, plots, and scientific figures. ChatGPT was used across these tasks, and Claude provided additional cross-checks of selected analyses and manuscript drafts. AI-generated critiques were treated as suggestions, not scientific evidence. The authors made the final methodological and interpretive decisions and take responsibility for the submission, including all AI-assisted content.

REPRODUCIBILITY STATEMENT

Appendix A specifies data partitions, training settings, checkpoint selection, and aggregation. Appendix C reports additional seed summaries and checks. Prediction and reference arrays were retained, and the reported decompositions were recomputed from those arrays with fixed model checkpoints.

REFERENCES

Anthony Baez, Wang Zhang, Ziwen Ma, Lam Nguyen, Subhro Das, and Luca Daniel. Guaranteeing conservation of integrals with projection in physics-informed neural networks. *arXiv preprint arXiv:2511.09048*, 2025. doi: 10.48550/arXiv.2511.09048. URL https://arxiv.org/abs/2511.09048.

Johannes Brandstetter, Daniel Worrall, and Max Welling. Message passing neural PDE solvers. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=vSix3HPYKSU.

Elsa Cardoso-Bihlo and Alex Bihlo. Exactly conservative physics-informed neural networks and deep operator networks for dynamical systems. *Neural Networks*, 181:106826, 2025. doi: 10.1016/j.neunet.2024.106826. URL https://doi.org/10.1016/j.neunet.2024.106826.

William E. Chapman, John Schreck, and Yingkai Sha. Hard conservation correctors can hide a degrading model when training autoregressive emulators. *arXiv preprint arXiv:2607.18416*, 2026. URL https://arxiv.org/abs/2607.18416.

Yuntian Chen, Dou Huang, Dongxiao Zhang, Junsheng Zeng, Nanzhe Wang, Haoran Zhang, and Jinyue Yan. Theory-guided hard constraint projection (HCP): A knowledge-based data-driven scientific machine learning method. *Journal of Computational Physics*, 445:110624, 2021. doi: 10.1016/j.jcp.2021.110624. URL https://doi.org/10.1016/j.jcp.2021.110624.

Derek Hansen, Danielle C. Maddix, Shima Alizadeh, Gaurav Gupta, and Michael W. Mahoney. Learning physical models that can respect conservation laws. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 12469–12510. PMLR, 2023. URL https://proceedings.mlr.press/v202/hansen23b.html.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979. URL https://www.jstor.org/stable/4615733.

Masanobu Horie and Naoto Mitsume. Graph neural PDE solvers with conservation and similarity-equivariance. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 18785–18814. PMLR, 2024. URL https://proceedings.mlr.press/v235/horie24a.html.

Matthias Karlbauer, Timothy Praditia, Sebastian Otte, Sergey Oladyshkin, Wolfgang Nowak, and Martin V. Butz. Composing partial differential equations with physics-aware neural networks. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 10773–10801. PMLR, 2022. URL https://proceedings.mlr.press/v162/karlbauer22a.html.

Zishuo Lan, Junjie Li, Lei Wang, and Jincheng Wang. FluxNet: Learning capacity-constrained local transport operators for conservative and bounded PDE surrogates. *arXiv preprint arXiv:2602.01941*, 2026. doi: 10.48550/arXiv.2602.01941. URL https://arxiv.org/abs/2602.01941.

Randall J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2002. ISBN 9780521009249. doi: 10.1017/CBO9780511791253.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=c8P9NQVtmnO.

Jia Yu Lim, Herath Mudiyanselage Viraj Vidura Herath, Sanka Rasnayaka, Lucy Marshall, Hui Zou, Abhishek Saha, Jobayer Hossain, Jing Yen Tong, Ing Zhen Lee, Yukiya Tsukada, Shrey Gandhi, Katsutoshi Matsumaro, Sajay Raj, Artyom Mazur, Andrey Khlopotnykh, and Dmytro Cheshenko. UrbanFloodBench: Bridging AI and hydrology through benchmarking of coupled 1D–2D urban flood surrogate models. *Environmental Modelling and Software*, 206:107154, 2026a. doi: 10.1016/j.envsoft.2026.107154.

Jia Yu Lim, Viraj Vidura Herath, Sanka Rasnayaka, Lucy Marshall, Hui Zou, and Abhishek Saha. UrbanFloodBench Dataset, 2026b.

Jia Yu Lim, Sanka Rasnayaka, Viraj Vidura Herath Herath Mudiyanselage, Lucy Marshall, Hui Zou, and Abhishek Saha. Urbanfloodbench: Flood modelling. https://www.kaggle.com/competitions/urban-flood-modelling, 2026c. Kaggle.

M. L. Mamud, M. K. Mudunuru, S. Karra, and B. Ahmmed. Quantifying local and global mass balance errors in physics-informed neural networks. *Scientific Reports*, 14:15541, 2024. doi: 10.1038/s41598-024-65472-9. URL https://doi.org/10.1038/s41598-024-65472-9.

Yuan Mi, Pu Ren, Hongteng Xu, Hongsheng Liu, Zidong Wang, Yike Guo, Ji-Rong Wen, Hao Sun, and Yang Liu. Conservation-informed graph learning for spatiotemporal dynamics prediction. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, pp. 1056–1067. Association for Computing Machinery, 2025. doi: 10.1145/3690624.3709244. URL https://doi.org/10.1145/3690624.3709244.

Anastasios Panagiotelis, George Athanasopoulos, Puwasala Gamakumara, and Rob J. Hyndman. Forecast reconciliation: A geometric view with new insights on bias correction. *International Journal of Forecasting*, 37(1):343–359, 2021. doi: 10.1016/j.ijforecast.2020.06.004. URL https://doi.org/10.1016/j.ijforecast.2020.06.004.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W. Battaglia. Learning mesh-based simulation with graph networks. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2010.03409.

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, pp. 6901–6913, 2022. doi: 10.52202/068431-0500.

Mehdi Taghizadeh, Zanko Zandsalimi, Mohammad Amin Nabian, Majid Shafiee-Jood, and Negin Alemazkoor. Interpretable physics-informed graph neural networks for flood forecasting. *Computer-Aided Civil and Infrastructure Engineering*, 40:2629–2649, 2025. doi: 10.1111/mice.13484. URL https://doi.org/10.1111/mice.13484.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Dan MacKinlay, Francesco Alesiani, Dirk Pfl"uger, and Mathias Niepert. PDEBench: An extensive benchmark for scientific machine learning. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL https://openreview.net/forum?id=dh_MkX0QfrK.

Matilde Valente, Tiago C. Dias, Vasco Guerra, and Rodrigo Ventura. Physics-consistent machine learning with output projection onto physical manifolds. *Communications Physics*, 8:433, 2025. doi: 10.1038/s42005-025-02329-1. URL https://doi.org/10.1038/s42005-025-02329-1.

Yihan Wang, Lujun Zhang, N. Benjamin Erichson, and Tiantian Yang. A mass conservation relaxed (MCR) LSTM model for streamflow simulation across CONUS. *Water Resources Research*, 61(8): e2024WR039131, 2025. doi: 10.1029/2024WR039131. URL https://doi.org/10.1029/2024WR039131.

A Reproducibility Details

The experiments use different physical systems and repeat units, so each result is aggregated within its own design. This appendix gives the data splits, training settings, and aggregation rules needed to reproduce those results.

A.1 Graded 16×16 benchmark

On a nonuniform mesh, the same change in an intensive state can correspond to different physical mass changes because cell volumes differ. The main benchmark therefore compares output parameterizations that satisfy the same mass balance in a setting where control-volume weighting matters. The physical system is a periodic finite-volume advection-diffusion equation with a prescribed source. The graded 16×16 mesh has a maximum-to-minimum cell-width ratio of 5.07.

Each trajectory contains 40 stored states, giving 39 consecutive transitions. Reference trajectories are integrated with second-order strong-stability-preserving Runge-Kutta (SSP-RK2), using eight substeps between stored states. All predictors share a 64-bit floating-point periodic CNN backbone with three 3×3 sigmoid linear unit (SiLU) convolutional blocks followed by an output-specific head. Inputs include the current state, prescribed source, local cell widths and control-volume measures, velocity, and diffusivity.

The data consist of 32 training trajectories, 8 validation trajectories, and 16 held-out trajectories. Face-FV, Antisymmetric, Density-relative, and Free projected are each trained with five independent training seeds, giving 20 trained checkpoints. The held-out trajectories are generated and evaluated only after these checkpoints and their validation records are fixed.

Each model uses AdamW for exactly 1,200 updates with batch size 16, without a scheduler or early stopping, and the step-1,200 checkpoint is evaluated directly. The training seed is the repeat unit; trajectories, transitions, cells, and horizons are nested observations within a seed.

Spatial accuracy in the graded-grid results is measured with the volume-weighted relative-L_1 field error. One-step values are computed from teacher-forced predictions, and the same definition is used at each of the 39 autoregressive rollout steps. Agreement with the supplied target mass is checked separately with the absolute constraint residual defined in Section 2.

A.2 Eight-seed source-increment comparison

Providing a prescribed source to the predictor and including the mass created by that source in the required next-step total are different operations. The source-increment comparison isolates this distinction by giving the same source field to both conditions while changing only whether the integrated source contribution is included in the supplied target mass.

For mesh grades 1, 2, 4, and 6, six trainable output conditions are run for each of eight seeds, giving 192 trained models. Each grade uses 24 training, 8 validation, and 8 held-out trajectories. The first three conditions are Free decoder, FluxNet-L-style graph readout, and a FluxNet head divided by control-volume measure. The other three are Antisymmetric with the source increment omitted from the supplied target mass, the same Antisymmetric parameterization with the source increment included, and an MC-LSTM port. The persistence baseline is separate from these 192 trained models.

The source-increment comparison uses the two Antisymmetric conditions. Their architecture, limiter, data split, optimizer search, training seed, initialization seed, and batch-permutation stream are matched. The only intended difference is whether the prescribed source mass increment is included in the required next-step total.

Network calculations use 32-bit floating point (fp32), while domain-mass and mass-balance calculations use 64-bit floating point (fp64). Training uses a rolling-window unroll loss with the additional budget-penalty weight set to zero. AdamW uses weight decay 10^{-2} and learning-rate candidates 10^{-3} and 3×10^{-3}. Runs continue for at most 10,000 epochs, with validation every five epochs and patience 1,500; the checkpoint with the lowest validation loss is retained. Held-out evaluation begins only after all 192 training runs and checkpoint choices are complete. The five confirmatory tests and their aggregation rules are fixed before this evaluation, and the corresponding run configuration and analysis records are included with the supplementary reproduction materials.

The trajectory generator samples horizontal velocity uniformly from $[0.3, 0.8]$, vertical velocity from $[-0.4, 0.4]$, and diffusivity from $[4 \times 10^{-3}, 1.2 \times 10^{-2}]$. The initial state is $0.2 + 0.8$ times a localized blob with center sampled in $[0.2, 0.8]^2$ and width sampled from $[0.06, 0.14]$. The prescribed source is a signed difference of two independently located blobs, with injection amplitude sampled from $[0.4, 1.0]$ and withdrawal amplitude from $[0.2, 0.6]$; their center and width ranges are the same as for the initial-state blob. The same source array is generated for both conditions, so the paired comparison changes only whether the prescribed source mass increment is included in the required next-step total, not source availability.

For each checkpoint, the score is the arithmetic mean of the cellwise RMSE over 32 rollout steps, followed by the mean over eight held-out events. The included-omitted contrast is computed first within the same seed and mesh grade. The four grade-level differences are then averaged to give one paired difference per training seed. These eight seed-level differences are summarized with a paired bootstrap and tested with an exact two-sided sign-flip test, with Holm correction across the five prespecified tests. Trajectories and mesh grades within a training seed are nested observations, not additional independent repeats.

A.3 Matched 64-cell predictor evaluation

The 64-cell study separates a one-time correction of a completed prediction from repeated correction during rollout. Projecting one completed prediction changes only its current field, whereas recursive projection also changes the states seen by later model steps. We evaluate both cases with fixed trained FluxNet and FluxGNN predictors.

The physical system is a regular one-dimensional finite-volume advection-diffusion equation on a domain of length 1, with velocity 0.2, diffusivity 0.001, and time step 0.002. Both forcing conditions include a prescribed source, and one also includes boundary exchange. Each predictor family uses the same three training-seed labels across the two forcing conditions, with one condition-specific checkpoint for each seed and condition, giving six checkpoints per family. The three matched seed labels are the repeat units.

The final held-out split contains eight trajectories per condition, with trajectory seeds 2000-2007, and is distinct from the validation trajectories. Before opening this split, we fixed the checkpoints, the uncorrected and recursively projected output rules, horizons $1, 8, 16, 32, 64, 96$, the error definitions, and the aggregation rules. Checkpoint hashes and validation replays were verified before final evaluation. The resulting evaluation data contain 576 scalar records and 9,216 field records per predictor family.

At each horizon, one training-seed summary is the median over its 16 condition-trajectory predictions. We then report the median and minimum-maximum across the three seed summaries. The 48 prediction-level values at one horizon are nested within three training seeds; they are not 48 independent repeats.

The uncorrected rollout feeds the completed decoded state directly into the next step. The recursively projected rollout instead projects each decoded state to the supplied target mass determined by the current state and prescribed external increments, then feeds that projected state forward. Later-horizon differences therefore include the effect of changing the input trajectory.

The secondary post-decoder output-rule comparison was added after the held-out results had been inspected. Predictor and decoder weights remain fixed while only the post-decoder output rule changes, and the learned updates use four-step rollout exposure with gradients detached between steps. Implementation details, metrics, and aggregation rules were fixed before this secondary evaluation.

A.4 Source-plus-boundary control

In an open system, both volumetric sources and external boundary exchange can change the next-step mass. A separate graded 16×16 control includes both contributions. The domain is periodic in y and open in x, with a time-varying prescribed source and prescribed external-boundary flux. The source is constant within one stored interval but may change between intervals.

The learned comparison uses three output rules, each with three training seeds and eight held-out trajectories. Free uses the original decoder with prediction loss. Penalty uses the same decoder with an added gradient-balanced budget penalty. Structural uses a mass-balanced structural readout with prediction loss.

The prediction is held fixed while the supplied target mass is changed. For the same rollout, we evaluate the correctly specified supplied target mass with both source and boundary contributions, one that omits the source contribution, and one that omits the boundary contribution. On the held-out graded-grid evaluation, the periodic reference solver closes the correctly specified one-step mass constraint to a worst absolute residual of 1.11×10^{-16}. In the source-plus-boundary control, including both external contributions gives a normalized mean absolute constraint residual between 3.57×10^{-16} and 3.82×10^{-16} across the three reported seeds. The numerator is the mean absolute constraint residual over the scored transitions. The denominator is fixed from the training split as the mean of

$$\left| \sum_i V_i \left(\phi_i^{t+1} - \phi_i^t \right) \right| .$$

We use this single training-derived scale, which avoids division by zero on scored steps with vanishing external forcing.

B Related Work

Learned PDE surrogates include neural operators that learn solution maps between function spaces, autoregressive message-passing PDE solvers, and benchmark suites spanning multiple time-dependent PDEs (Li et al., 2021; Brandstetter et al., 2022; Takamoto et al., 2022). Physics-aware models such as FINN instead build finite-volume transport structure directly into the learned model (Karlbauer et al., 2022). We focus on a narrower evaluation question: what exact global mass balance certifies once target specification and field error are separated.

One way to impose conservation in a learned PDE surrogate is to make the internal transport itself conservative. In finite-volume methods, fluxes across an internal face enter the two adjacent control volumes with opposite signs, so internal transport does not change the total extensive quantity (LeVeque, 2002). FluxGNN carries this structure into graph-neural message passing so that local updates follow a finite-volume conservation law (Horie & Mitsume, 2024). Other learned simulators impose conservation through antisymmetric interactions for particle momentum or symmetry-based graph constructions for spatiotemporal dynamics (Prantl et al., 2022; Mi et al., 2025). Our Face-FV and Antisymmetric experiment conditions use the same pairwise-cancellation principle. They allow us to compare spatial fields and rollouts while keeping the supplied target mass fixed.

A second family corrects a completed prediction so that it satisfies a known constraint. Hard-constraint projection maps network outputs to a constraint set, ProbConserv incorporates linear conservation laws into a probabilistic correction, and output-projection methods define constraint-preserving corrections under positive-definite metrics (Chen et al., 2021; Hansen et al., 2023; Valente et al., 2025). Forecast reconciliation uses the same linear projection geometry (Panagiotelis et al., 2021). Projection has also been used to enforce first-integral conservation in PINN and DeepONet models for dynamical systems (Cardoso-Bihlo & Bihlo, 2025). We apply this geometry to finite-volume mass and measure the mass-changing error fraction defined in Section 2.

In an open system, the conserved total may change over time. Conservation laws with prescribed boundary flux can be written as time-varying linear constraints (Hansen et al., 2023). Projection methods have also been developed to guarantee specified integral quantities (Baez et al., 2025). For the systems studied here, the required next-step total is the current mass plus the increments created by prescribed sources and signed boundary exchange. Our source-increment comparison separates supplying source information to the predictor from including the prescribed source mass increment in that total.

The weights used to define physical mass and the metric used to correct or score predictions are not generally the same object. Finite-volume mass uses control-volume measures, whereas ProbConserv can use predictive covariance for its correction, and forecast reconciliation and output projection allow generalized distances or positive-definite metrics (Hansen et al., 2023; Panagiotelis et al., 2021; Valente et al., 2025). Here the same control-volume measures V_i define both physical mass and the

weighted squared error. Under this choice, the mass-changing component measures mismatch with the reference mass and the mass-preserving component is spatial error within the same total.

Conservation-aware studies report mass-balance error and prediction error as separate quantities (Valente et al., 2025; Baez et al., 2025). Groundwater PINN evaluations likewise measure local and global mass-balance errors alongside solution accuracy, with the two assessments not necessarily agreeing (Mamud et al., 2024). A recent autoregressive atmosphere-emulator study showed that a hard budget corrector can deliver machine-precision closure while the raw prediction degrades (Chapman et al., 2026). Open-system hydrology also provides a setting in which unobserved incoming water flux can conflict with a strict mass-conservation specification (Wang et al., 2025). These lines of work motivate the three measurements used here. We examine the rollout effect of omitting a known source increment from the supplied target mass, the prediction-dependent mass-changing error fraction, and the mass-preserving spatial and rollout error under exact agreement with that mass.

C Additional Empirical Evidence

C.1 Spatial fields and rollouts can differ under the same supplied target mass

A single global mass equality does not determine how mass is arranged across cells. Before training, random nonzero transport weights already satisfy the structural mass balance to numerical precision while producing substantial weighted squared error (Figure 5). After training, all four output parameterizations satisfy the same supplied target mass to numerical precision while producing different spatial errors. Main Figure 1(c) shows the horizon-16 example; Figure 6 shows the reference field and signed prediction error for the same held-out case at horizons 1, 8, and 16.

The display uses a median-case selection rule. Among stored Face-FV seed-trajectory pairs, we choose the pair whose terminal volume-weighted relative-L_1 field error is closest to the evaluation-population median, then use that same training seed and held-out trajectory for all four output parameterizations. Table 2 gives the Section 2 weighted squared error, mass-changing error, and mass-preserving error for these predictions. Its columns are aggregated separately, so ratios of displayed medians need not equal the median mass-changing error fraction.

Table 2: Graded-grid weighted squared error and its Section 2 decomposition at selected horizons.

Output	h	Total weighted squared field error	Mass-changing error	Mass-preserving error	Mass-changing error fraction
Face-FV	1	1.604e-05	8.912e-36	1.604e-05	3.34e-31
	8	1.465e-04	7.909e-35	1.465e-04	4.71e-31
	16	1.091e-02	1.223e-34	1.091e-02	1.02e-32
Antisymmetric	1	2.335e-05	7.684e-36	2.335e-05	2.72e-31
	8	1.309e-04	7.244e-35	1.309e-04	4.43e-31
	16	4.371e-01	3.130e-34	4.371e-01	1.24e-33
Density-relative	1	7.097e-06	8.505e-36	7.097e-06	5.50e-31
	8	4.534e-05	8.796e-35	4.534e-05	9.84e-31
	16	7.802e-05	1.236e-34	7.802e-05	1.14e-30
Free projected	1	1.621e-05	7.952e-36	1.621e-05	3.76e-31
	8	6.622e-05	6.442e-35	6.622e-05	6.13e-31
	16	9.733e-05	1.255e-34	9.733e-05	1.08e-30

The 39-step autoregressive rollout uses the same volume-weighted relative-L_1 field error. Thin curves in Main Figure 3 are medians over 16 held-out trajectories within each training seed, and the thick curves are medians across the five seed summaries. One-step errors are of order 10^{-3} for all four parameterizations, but their rollouts diverge sharply. All five Antisymmetric seeds and three of the five Face-FV seeds become unstable; step-39 error reaches 3.772×10^{19} and 3.806×10^{22}, respectively. By contrast, all five Density-relative and Free projected seeds remain below 0.043. Main Table 1 summarizes the five training seeds as mean $\pm$ sample standard deviation, while the thin curves retain the seed-level trajectories. The raw prediction-reference total-mass difference is not plotted because it is not the constraint residual to the supplied target mass and grows with the field scale in these runaway trajectories.

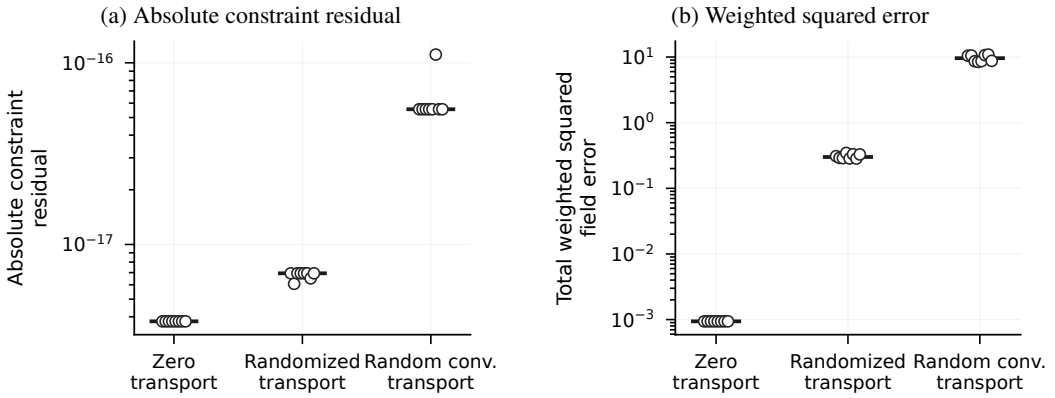


Figure 5: Pre-training absolute constraint residual and weighted squared error.

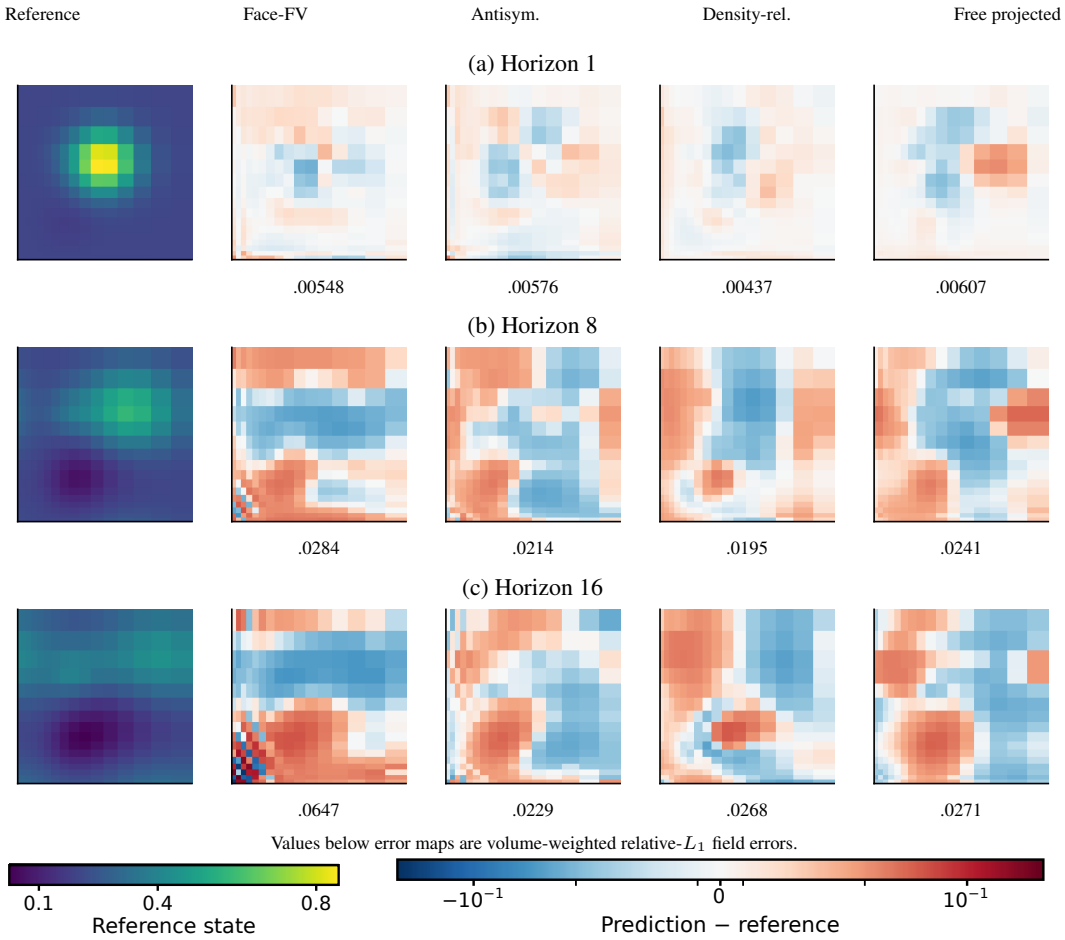


Figure 6: Reference fields and signed prediction errors for the fixed graded-grid case at horizons 1, 8, and 16.

C.2 Mass-changing error fraction varies for individual predictions

For completed predictions, the Section 2 mass-changing error fraction varies widely. In the final held-out 64-cell evaluation, its cross-seed median rises from 5.78% at horizon 1 to 16.7% for FluxNet. For FluxGNN, the horizon-96 median is 0.00912%. The 48 FluxNet prediction-level values at horizon 96 range from 0.108% to 76.5%.

Each training-seed summary is first computed as the median over its 16 condition-trajectory observations, after which we report the median and range across the three training-seed summaries. The 48 prediction-level values are nested within three training seeds; they are not 48 independent training replicates. The uncorrected results measure the realized error geometry of completed predictions against matched references.

Table 3: Uncorrected held-out weighted squared error and its decomposition.

Predictor	Horizon	Total weighted squared field error	Mass-changing error	Mass-preserving error	Mass-changing error fraction (%)
FluxNet	1	1.05×10^{-9}	5.12×10^{-11}	9.07×10^{-10}	$5.78\,[3.42, 6.71]$
FluxNet	8	3.94×10^{-8}	2.21×10^{-9}	3.09×10^{-8}	$5.62\,[4.52, 8.59]$
FluxNet	16	1.18×10^{-7}	1.26×10^{-8}	7.97×10^{-8}	$9.33\,[8.04, 16.5]$
FluxNet	32	2.50×10^{-7}	3.74×10^{-8}	1.74×10^{-7}	$15.2\,[12.7, 19.5]$
FluxNet	64	6.42×10^{-7}	8.12×10^{-8}	4.38×10^{-7}	$14.6\,[13.7, 18.1]$
FluxNet	96	1.38×10^{-6}	1.89×10^{-7}	8.66×10^{-7}	$16.7\,[16.1, 17.3]$
FluxGNN	1	3.79×10^{-5}	9.51×10^{-14}	3.79×10^{-5}	$2.33\,[1.37, 3.37]\times10^{-7}$
FluxGNN	8	1.70×10^{-3}	1.96×10^{-8}	1.70×10^{-3}	$0.00113\,[0.00112, 0.00114]$
FluxGNN	16	4.81×10^{-3}	1.17×10^{-7}	4.81×10^{-3}	$0.00239\,[0.00239, 0.0024]$
FluxGNN	32	1.13×10^{-2}	4.50×10^{-7}	1.13×10^{-2}	$0.00396\,[0.00395, 0.00398]$
FluxGNN	64	2.39×10^{-2}	1.52×10^{-6}	2.39×10^{-2}	$0.00598\,[0.00596, 0.00601]$
FluxGNN	96	3.82×10^{-2}	3.60×10^{-6}	3.82×10^{-2}	$0.00912\,[0.00908, 0.00916]$

Table 4: Recursively projected held-out weighted squared error and its decomposition.

Predictor	Horizon	Total weighted squared field error	Mass-changing error	Mass-preserving error	Mass-changing error fraction (%)
FluxNet	1	9.07×10^{-10}	1.14×10^{-33}	9.07×10^{-10}	8.16×10^{-23}
FluxNet	8	3.09×10^{-8}	9.05×10^{-33}	3.09×10^{-8}	2.21×10^{-23}
FluxNet	16	7.96×10^{-8}	3.93×10^{-32}	7.96×10^{-8}	5.53×10^{-23}
FluxNet	32	1.75×10^{-7}	1.89×10^{-31}	1.75×10^{-7}	1.33×10^{-22}
FluxNet	64	4.57×10^{-7}	6.08×10^{-31}	4.57×10^{-7}	1.28×10^{-22}
FluxNet	96	9.14×10^{-7}	1.07×10^{-30}	9.14×10^{-7}	1.03×10^{-22}
FluxGNN	1	3.79×10^{-5}	1.14×10^{-33}	3.79×10^{-5}	3.10×10^{-27}
FluxGNN	8	1.70×10^{-3}	9.05×10^{-33}	1.70×10^{-3}	5.04×10^{-28}
FluxGNN	16	4.81×10^{-3}	3.93×10^{-32}	4.81×10^{-3}	8.26×10^{-28}
FluxGNN	32	1.13×10^{-2}	1.89×10^{-31}	1.13×10^{-2}	1.74×10^{-27}
FluxGNN	64	2.39×10^{-2}	6.78×10^{-31}	2.39×10^{-2}	2.68×10^{-27}
FluxGNN	96	3.82×10^{-2}	1.15×10^{-30}	3.82×10^{-2}	2.93×10^{-27}

Once a projected state is fed back into the model, later predictions also change. At horizon 96, FluxNet's median weighted squared error decreases from 1.38×10^{-6} in the uncorrected rollout to 9.14×10^{-7} in the recursively projected rollout. For FluxGNN, both rollouts are about 3.82×10^{-2} at the same horizon. The later-horizon difference therefore combines the immediate mass correction with the effect of changing the subsequent input trajectory.

The same distinction appears in the secondary post-decoder output-rule comparison. Predictor and decoder weights remain fixed while the post-decoder rule changes among projection only, a learned free update, a free update followed by projection, and Antisymmetric, Density-relative, and Face-FV mass-neutral learned updates. Weighted squared error and per-step absolute constraint residual do not induce the same ranking. This comparison was planned after the initial held-out results had been inspected, so we treat it as a secondary analysis on the same held-out population.

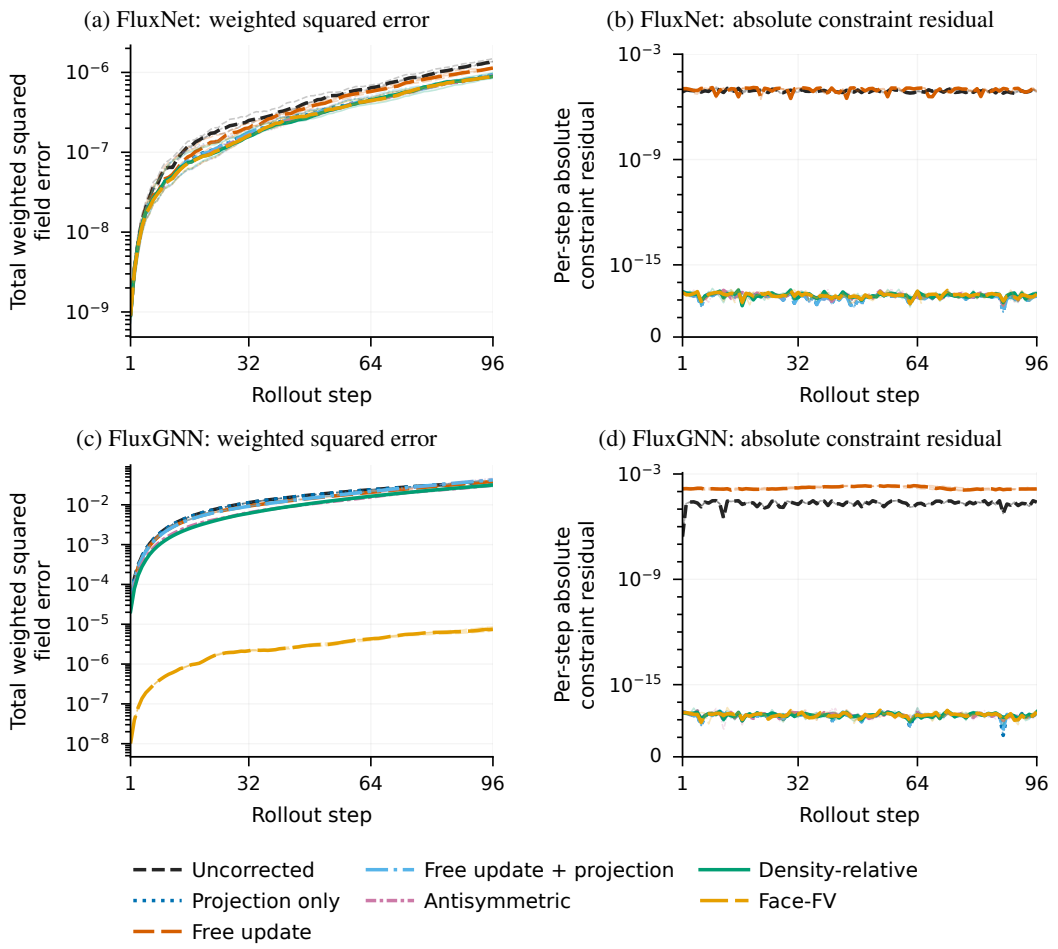


Figure 7: Secondary post-decoder output-rule rollouts: weighted squared error and per-step absolute constraint residual.

Table 5: Secondary post-decoder output-rule weighted squared error at the first and final reported horizons.

Predictor	Output rule	$h = 1$	$h = 96$
FluxNet	Uncorrected	1.05e-09	1.38e-06
	Projection only	9.07e-10	9.14e-07
	Free update	9.71e-10	1.14e-06
	Free update + projection	9.16e-10	9.36e-07
	Antisymmetric	8.87e-10	9.10e-07
	Density-relative	9.06e-10	8.91e-07
	Face-FV	8.98e-10	8.95e-07
FluxGNN	Uncorrected	3.79e-05	3.82e-02
	Projection only	3.79e-05	3.82e-02
	Free update	3.40e-05	3.84e-02
	Free update + projection	3.38e-05	4.23e-02
	Antisymmetric	2.49e-05	3.12e-02
	Density-relative	2.08e-05	3.14e-02
	Face-FV	1.01e-08	7.52e-06

C.3 External mass increments must be included in the required next-step total

A predictor can receive the prescribed source field while still being constrained to the wrong next-step total mass. The eight-seed source-increment comparison gives the prescribed source field to both conditions and changes only whether its prescribed source mass increment is included in the required next-step total mass. Table 6 reports the seed-level omitted-minus-included 32-step horizon-mean cellwise RMSE differences; all eight are positive.

The mean paired difference is 0.009074, with 95% paired-bootstrap interval $[0.007978, 0.010448]$. The exact two-sided sign-flip test gives $p = 0.0078125$; applying Holm correction across the five prespecified confirmatory tests gives 0.0390625 (Holm, 1979). The paired result shows that supplying the source field and including the source mass increment in the required next-step total can have different consequences for rollout accuracy.

The two prespecified mesh-grading diagnostics move in the opposite direction from the proposed monotonic mechanism. The prespecified slope across mesh grades has mean -0.208 with 95% interval $[-0.319, -0.089]$, and the uniform-grid negative control has mean misspecified-minus-correctly-specified difference 0.00311 with interval $[0.000897, 0.00544]$. These diagnostics do not support an effect that grows monotonically with mesh grading. We therefore interpret the paired source-increment result as an effect of required next-step total specification and do not infer a mesh-grading mechanism.

Boundary exchange enters the supplied target mass in the same way as a volumetric source. A separate graded 16×16 control is periodic in y, open in x, and contains both a prescribed source and signed external-boundary exchange. When both contributions are included in the correctly specified supplied target mass, the dimensionless mean absolute constraint residual is between 3.57×10^{-16} and 3.82×10^{-16} across the three reported seeds. This residual is normalized by the mean absolute reference mass increment on the training split, fixed before training. Removing either contribution from the evaluated budget for the same stored rollout increases this normalized residual by many orders of magnitude. This fixed-output audit therefore tests how the residual is defined; it does not enforce two alternative targets.

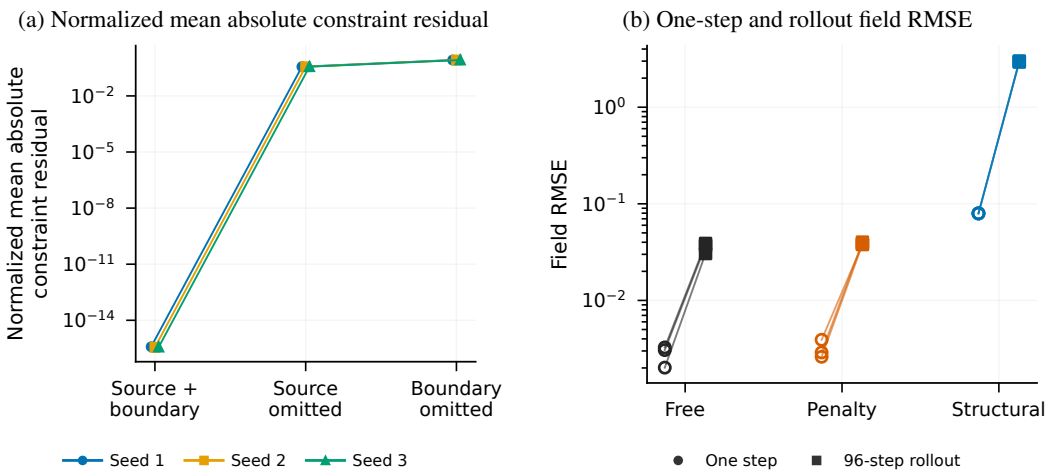


Figure 8: Source-plus-boundary control: normalized mean absolute constraint residuals for the same stored rollout under the complete budget and budgets omitting either source or boundary contribution.

C.4 Mass correction and field error remain distinct under other conditions

Inference-time exact projection often lowers point RMSE under the prespecified distribution shifts. At horizon h, point RMSE is the mean across evaluation trajectories of the cellwise RMSE at step h; it is not the horizon-mean RMSE averaged from steps 1 through h. With trained weights fixed, projection is toggled on or off for one unshifted condition and 12 prespecified shift blocks. Projected predictions have lower point RMSE in 13/13 conditions at horizon 8, 10/13 at horizon 16, and 12/13 at horizon 300. Median Free-minus-projected RMSE differences are 4.6923×10^{-4},

Table 6: Seed-level omitted-minus-included 32-step horizon-mean cellwise RMSE differences for the prescribed source mass increment.

Seed	ΔRMSE	Seed	ΔRMSE
1	0.013151	5	0.010021
2	0.009653	6	0.007814
3	0.007356	7	0.007289
4	0.008797	8	0.008513

1.0546×10^{-3}, and 8.4030×10^{-2}, respectively. Because autoregressive projection changes later inputs, this within-weight rollout evidence is distinct from the completed-prediction mass-changing error fraction.

In the independent Face-FV extension, all 32 trained models satisfy the structural mass-conservation check, with a maximum absolute component residual of 8.47×10^{-16}. This residual is the maximum, over scored steps and conserved components, of the absolute volume-weighted sum of the internal state increment. The rollout contrasts use the same 32-step horizon-mean cellwise RMSE as Section 3, with mesh grades averaged within each training seed before the contrast is formed. The prespecified mean contrasts are -0.00104 for Free minus Face-FV, -0.00177 for Density-relative minus Face-FV, and $+0.00526$ for uncapped Antisymmetric minus Face-FV; all three have Holm-adjusted $p = 1.0$. None of the prespecified differences is detected in this test; the result does not establish equivalence.

A closed-periodic traffic setting removes both source and external-boundary exchange. The held-out one-step field RMSE pools squared error over all cells, admissible start indices, and held-out trajectories before taking one square root. For the five paired seeds, mean field RMSE is 7.013×10^{-4} for Uncorrected, 2.551×10^{-4} for Face-FV, and 1.230×10^{-4} for Density-relative; the table reports the corresponding sample standard deviations. Both mass-conserving output parameterizations are below Uncorrected in all five pairs.

Table 7: Closed-periodic traffic one-step field RMSE across five paired training seeds (mean ± sample standard deviation).

Output rule	Field RMSE (mean ± SD)
Uncorrected	7.013e-04 ± 1.101e-04
Face-FV	2.551e-04 ± 0.058e-04
Density-relative	1.230e-04 ± 0.123e-04

The same checkpoints have a 25-step autoregressive evaluation on 50 validation trajectories. This population is selection-exposed and is separate from the held-out one-step test. At each rollout step, the field RMSE is obtained by computing the volume-weighted mean squared field error within each trajectory, averaging those values across trajectories, and then taking the square root. With no external forcing, the per-step absolute constraint residual is $|\sum_i V_i(\hat{\phi}_i^{t+1} - \hat{\phi}_i^t)|$. Face-FV and Density-relative keep this residual much smaller than Uncorrected while field RMSE evolves separately. In the one-step comparison, output parameterization and parameter count also change, so the RMSE difference is not interpreted as a causal effect of conservation alone.

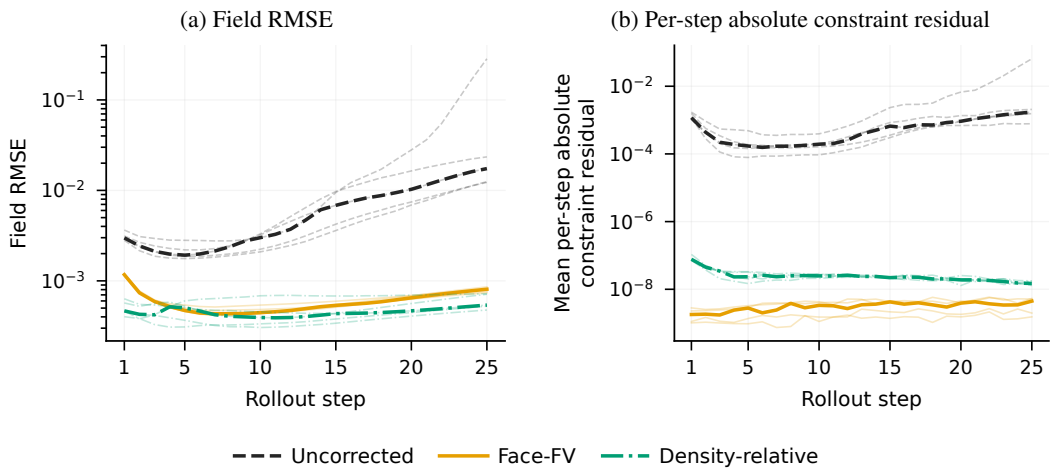


Figure 9: Closed-periodic traffic 25-step validation rollouts.

C.5 The same separation appears on an irregular urban-flood mesh

The UrbanFloodBench Model 2 evaluation provides an irregular surface mesh on which a small prediction-reference total-water-volume difference still does not determine spatial accuracy. This experiment uses the competition-format Model 2 data distributed for the January 2026 UrbanFloodBench Kaggle challenge (Lim et al., 2026a;c). Model 2 is the Davis, California HEC-RAS-derived system with 198 1D nodes and 4,299 2D surface cells.

The competition-format copy separates 69 training-side events from 30 competition-test events. Training and evaluation in this paper remain entirely within the 69-event training-side pool: 55 events are used for training, 7 for validation, and the internal-heldout events are 16, 48, 57, 63, 81, 89, and 97. None of the 30 Kaggle competition-test events is used in the reported Flood results.

The later full UrbanFloodBench release is not the same file set as the competition-format data used here (Lim et al., 2026b). It provides complete HEC-RAS simulation output for Models 1-4 and full trajectories for all 99 Model 2 events, without the competition-specific temporal truncation. Direct comparison of the preserved files shows the practical difference. Competition-format Model 2 training event 16 begins at 04:00, whereas the corresponding full-release trajectory begins at 00:00 and ends at the same time. A competition-test example likewise lacks four hours at the beginning and three hours at the end that are present in the full release. The Flood results in this paper therefore characterize the earlier competition-format Model 2 trajectories; the later full benchmark release was not used for this evaluation.

The evaluated state is volume-equivalent water depth, obtained by dividing 2D cell water volume by cell area. Surface-cell area is denoted by A_i, internal-heldout events by r, and recorded time steps by t; the predicted and reference depths are $\hat{h}_{r,i}(t)$ and $h^{\mathrm{ref}}_{r,i}(t)$. Rainfall and signed reference-solver 1D-2D exchange are prescribed inputs. The surface surrogate therefore predicts the 2D state conditional on these forcings. Three backbones, six output conditions, and three training seeds give 54 fixed checkpoints.

The primary depth RMSE uses surface cells that are valid under the dataset mask and exceed $0.01\,\mathrm{m}$ in the reference trajectory at least once. Squared depth errors are pooled over all selected cell-time observations from the full recorded trajectories of the seven internal-heldout events before taking the square root:

$$\left[\frac{\sum_r \sum_t \sum_{\substack{i\ \text{valid} \\ \max_{t'} h^{\mathrm{ref}}_{r,i}(t') > 0.01\,\mathrm{m}}} \left(\hat{h}_{r,i}(t) - h^{\mathrm{ref}}_{r,i}(t)\right)^2}{\sum_r \sum_t \sum_{\substack{i\ \text{valid} \\ \max_{t'} h^{\mathrm{ref}}_{r,i}(t') > 0.01\,\mathrm{m}}} 1}\right]^{1/2}.$$

This metric pools cell-time observations across events; events are not given equal weight. Each training seed contributes one RMSE summary, and Table 8 reports the mean and sample standard deviation across the same three seeds.

Main Figure 4 uses the first 96 steps shared by all seven internal-heldout events. At each step t, its depth RMSE pools squared errors over the same reference-defined ever-wet cells across events:

$$\left[\frac{\sum_r \sum_{\substack{i \text{ valid} \\ \max_{t'} h_{r,i}^{\mathrm{ref}}(t')>0.01\,\mathrm{m}}} \left(\hat{h}_{r,i}(t) - h_{r,i}^{\mathrm{ref}}(t)\right)^2}{\sum_r \sum_{\substack{i \text{ valid} \\ \max_{t'} h_{r,i}^{\mathrm{ref}}(t')>0.01\,\mathrm{m}}} 1}\right]^{1/2}.$$

The prediction-reference total-water-volume difference at the same step is

$$\frac{1}{7}\sum_r \left|\sum_i A_i \hat{h}_{r,i}(t) - \sum_i A_i h_{r,i}^{\mathrm{ref}}(t)\right|,$$

reported in m^3. This is not the constraint residual against the supplied target mass. A mass-balanced output rule can still have a small nonzero prediction-reference volume difference because the prescribed rainfall-plus-manhole budget and the stored reference-volume trajectory do not close identically. We therefore keep this diagnostic in absolute physical units because reference volume can be nearly zero in early dry states.

For output rules that include the prescribed external increments, predicted total water volume satisfies the supplied target mass to numerical precision. At the common step 96, the mean absolute prediction-reference difference is about $438\,\mathrm{m}^3$. For the checked MeshGraphNets-style seed 0, this corresponds to about 0.005%-0.798% of event reference volume, with mean 0.216% and median 0.184%. By contrast, the Uncorrected rollout reaches about 2.0-2.2 $\times\ 10^5\,\mathrm{m}^3$ depending on the backbone, and Density-relative with the prescribed increment omitted reaches about $2.38 \times 10^5\,\mathrm{m}^3$.

Including the prescribed rainfall and 1D-2D exchange increments in the required next-step total lowers pooled depth RMSE in all nine paired backbone-seed comparisons. Spatial error and seed variability remain across output rules and backbones, so supplied-target-mass satisfaction, agreement with the reference total, and spatial field accuracy remain distinct on this irregular mesh.

The temporal metrics do not show where these errors occur on the 4,299-cell surface mesh, so we also retain fixed spatial views. These views use event 16 and training seed 0. Each backbone figure shows one 3×5 matrix: subfigure rows are steps 1, 222, and 444, and columns are the reference plus the four comparison outputs.

The same five outputs appear in every backbone figure: Reference, Uncorrected, Density-relative with prescribed external increments omitted, Density-relative with those increments included, and Face-FV. All three figures use one reference-depth scale and one symmetric-log signed-error scale, linear within ± 0.01 m, without clipping. Snapshot RMSE is ordinary cellwise depth RMSE over all 4,299 surface cells at that time step and is distinct from the primary RMSE pooled over reference ever-wet cells. Event, seed, timesteps, field values, and normalizations remain fixed. Main Figure 4 and Table 8 retain the full six-condition time evolution.

Table 8: Flood depth RMSE (m) on seven internal-heldout events (mean $\pm$ sample standard deviation across three training seeds).

Output rule	MeshGraphNets-style	FluxGNN-style	HydroGraphNet-style
Uncorrected	0.1204 ± 0.0014	0.1045 ± 0.0031	0.1379 ± 0.0061
Density-relative, increments omitted	0.1690 ± 0.0002	0.1726 ± 0.0017	0.1797 ± 0.0014
Density-relative, increments included	0.0134 ± 0.0011	0.0390 ± 0.0044	0.0912 ± 0.0559
Antisymmetric, uncapped	0.0194 ± 0.0006	0.0345 ± 0.0026	0.1213 ± 0.0687
Face-FV, uncapped	0.0171 ± 0.0007	0.0372 ± 0.0022	0.0342 ± 0.0085
Antisymmetric, capped	0.0210 ± 0.0002	0.0308 ± 0.0017	0.0793 ± 0.0650

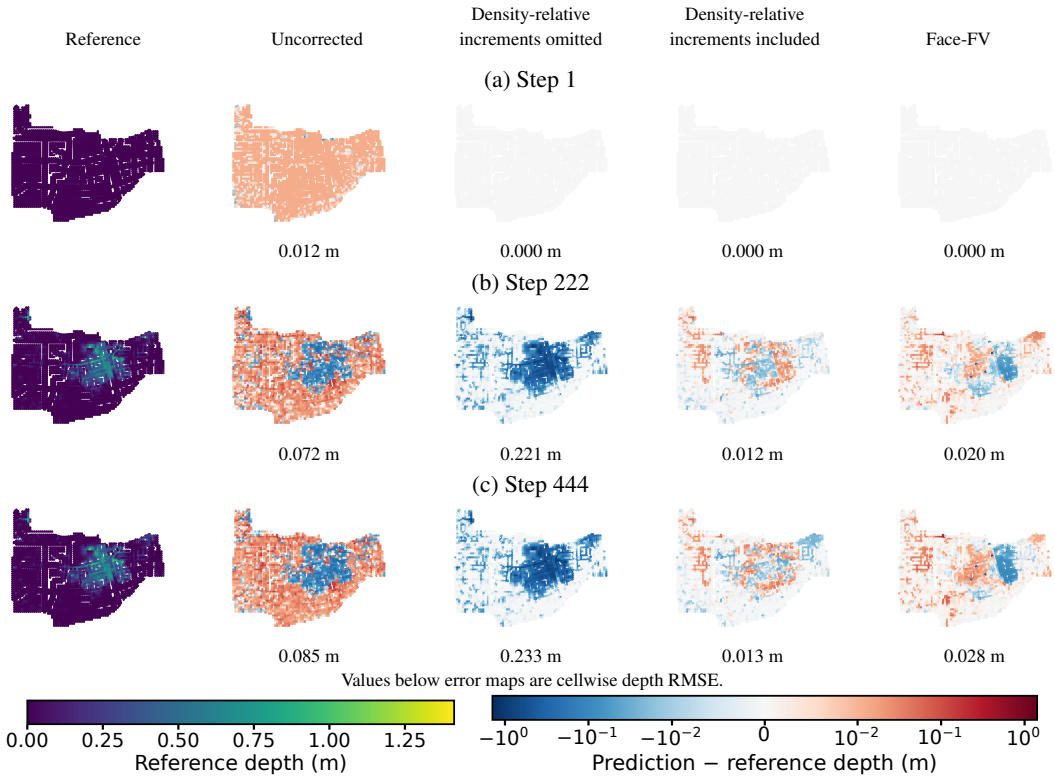


Figure 10: MeshGraphNets-style fields for event 16 and seed 0 at steps 1, 222, and 444.

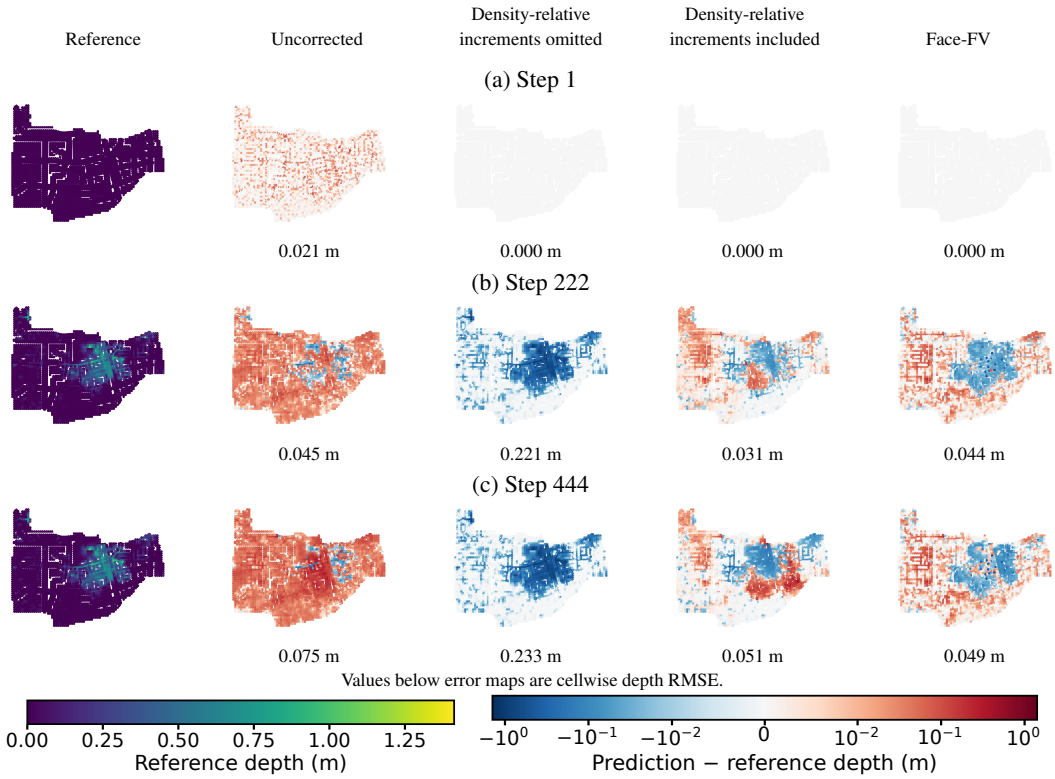


Figure 11: FluxGNN-style fields for the same event and seed at steps 1, 222, and 444.

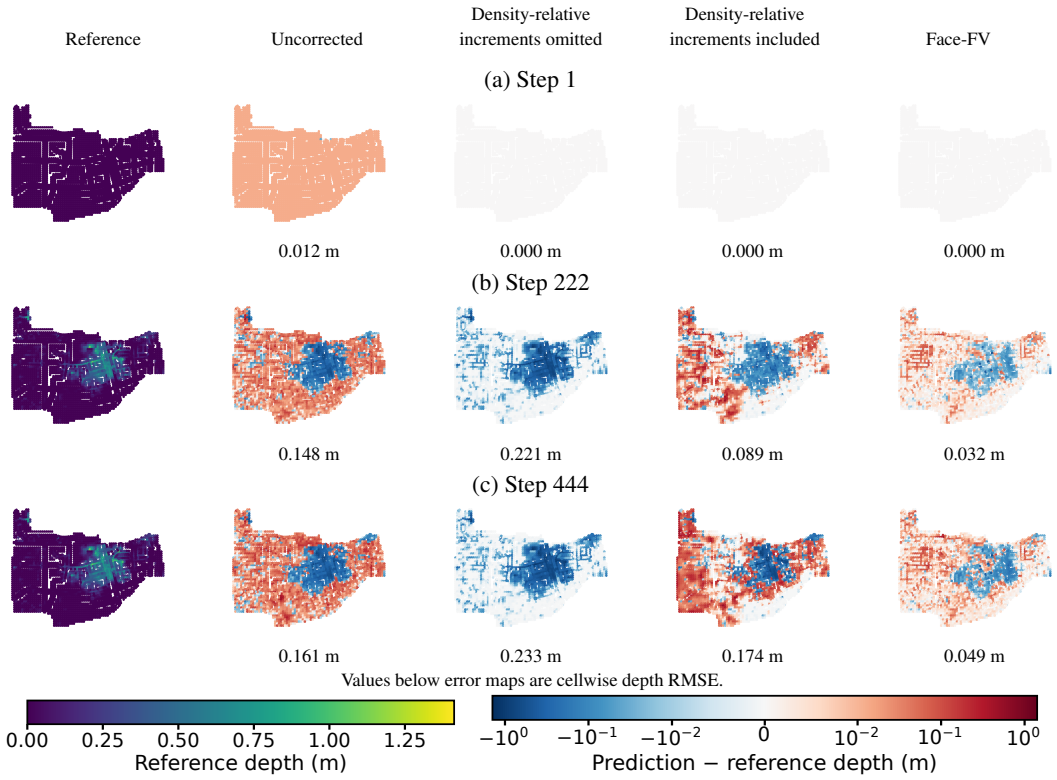


Figure 12: HydroGraphNet-style fields for the same event and seed at steps 1, 222, and 444.

D Technical Checks and Implementation Details

D.1 Post-decoder output rules for the 64-cell predictors

The 64-cell study keeps the trained FluxNet and FluxGNN predictor paths unchanged so that mass correction does not alter their internal representations (Lan et al., 2026; Horie & Mitsume, 2024). Encoder, processor, decoder, and trained weights are all fixed, and each model first produces the completed state prediction from its original decoder.

The uncorrected condition reports this decoded state directly and feeds it to the next step. The exact-projection condition applies the same open-system mass projection after decoding. In the recursively projected rollout, the projected state then becomes the next model input. The completed-prediction comparison uses the Section 2 mass-changing and mass-preserving errors, whereas the recursive-rollout comparison also includes the effect of changing subsequent model inputs.

Projection to the supplied target mass does not guarantee nonnegative cell states. We therefore check the final held-out 64-cell rollouts directly. The check covers two predictor families, two conditions, three training seeds, eight trajectories per condition, and 96 steps; each output rule contains 9,216 states and 589,824 cell values. No cell has $\phi_i < 0$ in either the uncorrected or recursively projected outputs. The smallest projected cell value is 0.05877 for FluxNet and 0.30068 for FluxGNN. These values describe the stored held-out predictions; they are an empirical check, not a positivity guarantee of the projection operator.