GUIDED OBSERVATION-SPACE PROJECTIONS FOR CONSTRAINED FLOW MATCHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative models of physical fields are often required at inference time to satisfy observations or physical constraints that were not enforced during training. Projection-based samplers meet them by correcting the predicted final sample onto the constraint set at every step, but a gradient-guidance step taken beforehand is largely lost in that correction and in the interpolation that follows. We introduce guided observation-space projection (GOSP), which keeps a guided displacement across repeated projection by updating the source sample with the current state, and characterise the guidance that survives a linear projection. Because the projection runs at every step, we solve it in observation space without forming the constraint Jacobian, with a matrix-free and a structured kernel-Gram solver. On Navier–Stokes benchmarks GOSP improves reconstruction from sparse and moving observations while keeping the constraint residuals of a projection sampler.

1 INTRODUCTION

Flow matching learns a time-dependent vector field whose flow transports a simple source distribution to a data distribution (Lipman et al., 2022; Liu et al., 2022; Albergo et al., 2025). It has been used for generative modelling of images (Esser et al., 2024), speech (Le et al., 2023) and protein conformational ensembles (Jing et al., 2024), and its extension to functions (Kerrigan et al., 2023; Lim et al., 2025) lets a sample be a whole spatial or spatiotemporal field.

In standard flow matching, an interpolation between source and data samples defines a target velocity field, and sampling follows the learned velocity from the source distribution to the data distribution. This by itself does not enforce additional conditions at inference time. In physical systems, however, generated samples often need to satisfy observations, initial or boundary conditions, conservation laws, or other task-specific constraints. For example, a flow model trained on Navier–Stokes trajectories may be asked to generate a trajectory that agrees with a few velocity sensors, starts from a given vorticity, or conserves a physical quantity. Matching the observed values alone is not sufficient; the generative model should also produce a plausible completion of the unobserved field. The challenge is therefore to enforce such constraints at inference time while retaining what the pretrained model has learned.

Two families of inference-time methods address this. Projection methods move the model's prediction of the final sample onto the constraint set at every step (Utkarsh et al., 2025; Cheng et al., 2025). They enforce linear constraints exactly, but the correction lies in the row space of the constraint Jacobian, so with sparse sensors it changes the field where it is measured and leaves the rest to later network evaluations. Guidance methods add to each step the gradient of the constraint misfit taken through the network (Chung et al., 2022a), which carries the misfit into the unobserved field but leaves a residual. The obvious combination is to guide and then project, and it provides little benefit. A projection sampler does not step forward from the corrected prediction. It rebuilds the next state by interpolating from the source sample to that prediction. A guided change of the current state therefore reaches the next state only through the prediction, and the projection removes the part of it that lies along the constraint rows. Little of the guided displacement survives.

If the source sample moves together with the guided state, the interpolation reproduces the guided state instead of undoing it, and the projection then acts on a prediction that already contains the guidance. We call this *charging the guidance step to the source sample*: at each step, we take a

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.

guidance step, shift the source sample accordingly, and project the prediction. The projection itself is unchanged from the projection sampler of Utkarsh et al. (2025).

Projecting at every step, however, has a cost. Each projection solves a linear system with the Gram matrix JJ^T of the constraint Jacobian, and in the released implementation of projection-based flow matching[1], assembling J takes almost all of the sampling time. For fields with hundreds of thousands of unknowns and thousands of constraints, the projection rather than the network becomes the cost of the method.

To address both issues, we propose *guided observation-space projection* (GOSP), a simple inference-time method for constrained generation with pretrained flow models. GOSP combines source charging, which preserves guidance across repeated projections, with an observation-space formulation that reduces the cost of those projections. It modifies only inference and requires no retraining of the flow model. Our contributions are:

- We introduce a source-charging update that keeps a guidance step alive under repeated projection; before a late projection start it reduces to guidance alone. For linear constraints we characterise the part of a guidance step that survives the projection analytically, and ablations confirm it.
- We develop an observation-space projection that avoids explicit Jacobian construction and substantially reduces projection cost.
- We evaluate GOSP on sparse, moving, nonlinear and noisy constraints in Navier–Stokes and shallow-water systems, including held-out references and sensor layouts without re-tuning.

2 Flow matching and constrained sampling

2.1 Flow-matching sampling

Flow matching transports a source measure π_0 to the data distribution over a flow time $\tau \in [0, 1]$. Training draws a source sample $u_0 \sim \pi_0$, the prior draw, and a data sample u_1, and forms the linear path $u_\tau = (1-\tau)u_0 + \tau u_1$. Along it the model regresses $v_\theta(u_\tau, \tau)$ onto the constant velocity $u_1 - u_0$ (Lipman et al., 2022; Liu et al., 2022; Albergo et al., 2025). In Functional Flow Matching (Kerrigan et al., 2023) a sample is a discretised solution field, π_0 a Gaussian process and v_θ a neural operator (Li et al., 2020). Sampling draws $u_0 \sim \pi_0$ and takes K Euler steps, $\tau' = \tau + 1/K$.

Because the velocity is learned along that straight path, the current state gives a prediction of the final sample. This look-ahead,

$$\xi(u_\tau) = u_\tau + (1-\tau)\, v_\theta(u_\tau, \tau), \tag{1}$$

is what the constrained samplers below act on. Constraints on the final sample are a differentiable residual $h : \mathbb{R}^n \to \mathbb{R}^m$, with $h(u) = 0$ when they hold.

2.2 Projection-based constraint enforcement

Explicit methods move the prediction onto the constraint set at every step, using only the constraints. Physics-Constrained Flow Matching (PCFM) (Utkarsh et al., 2025) takes one damped Gauss–Newton step,

$$\hat{\xi} = \xi - J^\mathsf{T}\lambda, \qquad \left(JJ^\mathsf{T} + \varepsilon I\right)\lambda = h(\xi), \qquad J = Dh(\xi) \in \mathbb{R}^{m\times n}, \tag{2}$$

which is exact for linear constraints and linearised for nonlinear ones. PCFM does not then step forward from the corrected endpoint. It rebuilds the next state by interpolating again, from the stored prior draw u_0 to the projected endpoint, $u_{\tau'} = (1-\tau')u_0 + \tau'\hat{\xi}$, and projects once more at $\tau = 1$. The prior draw thus stays part of the state for the whole run, which is what §3 builds on. Extrapolation–Correction–Interpolation (ECI) sampling (Cheng et al., 2025) also corrects the prediction, with closed-form maps for simple constraints such as prescribed values, but continues from a fresh prior draw.

[1] https://github.com/cpfpengfei/PCFM

2.3 GRADIENT GUIDANCE

Implicit methods keep the flow's own update and add a step that reduces the constraint misfit of the prediction. The gradient is taken through the model, and nothing forces the misfit to vanish. In diffusion posterior sampling (DPS) (Chung et al., 2022a) each Euler step is followed by a step along

$$g = \nabla_u \big\| h\big(\xi(u)\big) \big\| \Big|_{u=u_\tau}, \qquad u_{\tau'} = u_\tau + \Delta\tau\, v_\theta(u_\tau, \tau) - \zeta g, \tag{3}$$

with the gradient taken through the network. This is equivalent to the normalized DPS guidance step up to a factor of two, which can be absorbed into ζ; it is not the Gaussian log-likelihood gradient. DiffusionPDE (Huang et al., 2024) holds v_θ fixed in the gradient. D-Flow (Ben-Hamu et al., 2024) differentiates $\|h\|^2$ through the whole sampler to optimise the prior draw, at the cost of tens of sampling runs per sample. Figure 1 sketches both next to GOSP.

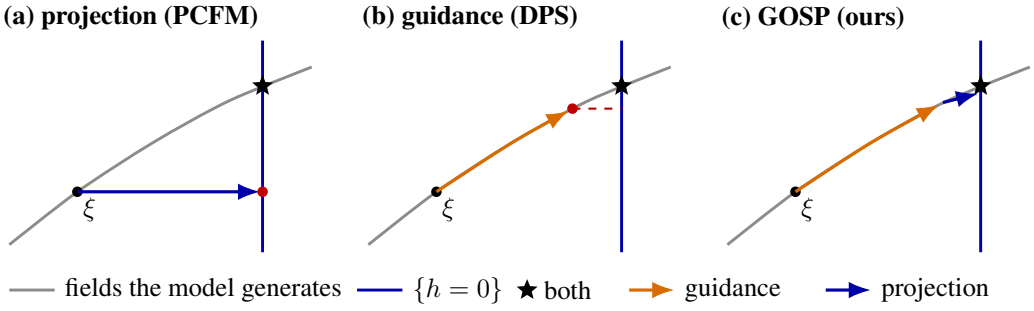


Figure 1: Projection, gradient guidance and GOSP, starting from the same prediction ξ. (a) Projection uses only the geometry of the constraints, $J = Dh$, and moves to the nearest point of $\{h = 0\}$ (exactly for linear h, to first order otherwise), which need not be a field the model would generate. (b) Guidance follows the model through $\partial v_\theta / \partial u$ and ends off $\{h = 0\}$. (c) GOSP guides, then projects, which assumes the projection is usable at the guided prediction; §3.3 treats the case where it is not.

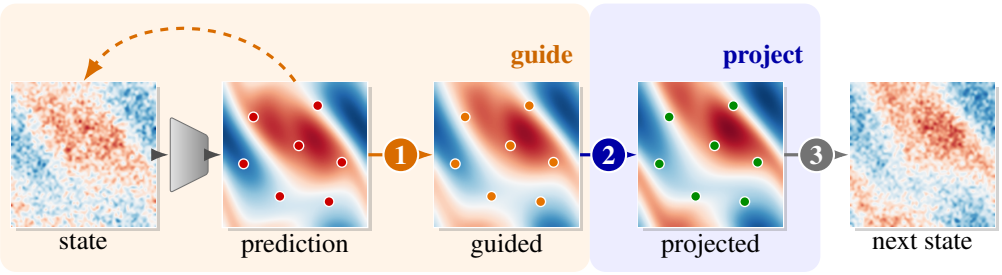


Figure 2: One GOSP step on a vorticity field with sparse sensors (dots). The sensor misfit is passed back through the flow model (dashed) to move the state and its prior draw (1), the guided prediction is projected (2), and the next state is interpolated (3).

3 GUIDED OBSERVATION-SPACE PROJECTION

A natural way to combine guidance and projection is to take a guidance step before projecting the predicted endpoint onto the constraint set. The difficulty is that a projection sampler does not continue directly from this corrected endpoint. Instead, it constructs the next state by interpolating between the stored prior draw and the projected prediction. If guidance moves only the current state, while the prior draw remains fixed, this interpolation partially undoes the guided displacement. The projection then removes the component of the displacement along the constrained directions, leaving little of the original guidance to affect the next network evaluation.

GOSP avoids this loss by moving the prior draw together with the guided state. We charge the guidance step to the source sample so that the subsequent interpolation reproduces the guided state

rather than pulling it back toward the original source. The projection then acts on a prediction that already carries the guidance, allowing its unconstrained component to persist across repeated projection steps.

3.1 THE GOSP UPDATE

A PCFM state is an interpolation between the prior draw and the most recent projected endpoint, $u_\tau = (1-\tau)u_0 + \tau\hat{\xi}$, and the next state depends on it only through the look-ahead ξ. GOSP therefore moves the prior draw with the state (Figure 2); we call this charging the step to the prior draw. At flow time $\tau = k/K$, with $\tau' = \tau + 1/K$ and $v = v_\theta(u_\tau, \tau)$,

$$g = \nabla_u \big\| h\big(u + (1-\tau)\, v_\theta(u,\tau)\big) \big\| \Big|_{u=u_\tau}, \tag{4}$$

$$\tilde{u} = u_\tau - \zeta g, \qquad \tilde{u}_0 = u_0 - \frac{\zeta}{1-\tau}\, g, \tag{5}$$

$$\tilde{\xi} = \tilde{u} + (1-\tau)\, v, \qquad \hat{\xi} = \tilde{\xi} - J^\mathsf{T}\big(JJ^\mathsf{T} + \varepsilon_\tau I\big)^{-1} h(\tilde{\xi}), \quad J = Dh(\tilde{\xi}), \tag{6}$$

$$u_{\tau'} = (1-\tau')\, \tilde{u}_0 + \tau'\, \hat{\xi}, \qquad u_0 \leftarrow \tilde{u}_0. \tag{7}$$

A tilde marks a quantity after the guidance step and a hat one after the projection: $\tilde{u}$ and $\tilde{u}_0$ are the guided state and prior draw, $\tilde{\xi}$ is their look-ahead and $\hat{\xi}$ its projection. As in DPS, the gradient (4) is taken through the network and the norm is not squared, so that ζ is a step length in the state; at $h = 0$ we take the zero subgradient. The network is evaluated once per step, at u_τ, and guidance, projection and interpolation use that velocity. Evaluating it at the moved state instead orders the substeps differently, and changes the look-ahead by $-(1-\tau)\zeta\, \partial_u v_\theta\, g + O(\zeta^2)$. The steps we select ($\zeta$ up to 10) are not small in this sense. The ablations compare the two orderings.

When the projection starts only at $\tau_0 > 0$ (§3.3), charging a step before τ_0 is not needed. It would move the stored prior draw away from the one the state and the velocity imply,

$$\hat{\epsilon} = \tilde{u} - \tau v. \tag{8}$$

On such steps GOSP interpolates from that draw instead, $u_{\tau'} = (1-\tau')\hat{\epsilon} + \tau'\tilde{\xi}$ with $u_0 \leftarrow \hat{\epsilon}$, which is the Euler step $\tilde{u} + \Delta\tau\, v$. Before τ_0 GOSP therefore takes DPS's steps and, like DPS, holds the restricted initial condition. On sets projected at every step the rule never applies.

With $\zeta = 0$ and a projection at every step, GOSP reproduces PCFM sample for sample; with a later start it does not, since its unprojected steps are Euler steps. Every state is an interpolation $\tilde{u} = (1-\tau)\tilde{u}_0 + \tau\hat{\xi}_{\text{prev}}$ between a moved prior draw and a projected endpoint. This has the form of the training path, but $\tilde{u}_0$ is no longer a draw from π_0. The charged source is bookkeeping for the interpolation, and GOSP is an inference-time construction rather than posterior sampling. Through the prior draw, a guidance step at τ enters later states with weight at most $(1-\tau')/(1-\tau) \le 1$, and the final sample with weight zero. The last step is a projection, as in PCFM. Algorithm 1 and the derivations are in Appendix C.

3.2 EFFECT OF PROJECTION ON THE GUIDANCE STEP

For a linear residual $h(\xi) = A\xi - y$ the projection in (6) is affine, $\hat{\xi} = M\tilde{\xi} + A^\mathsf{T}(AA^\mathsf{T} + \varepsilon I)^{-1}y$ with $M = I - A^\mathsf{T}(AA^\mathsf{T} + \varepsilon I)^{-1}A$, and the guidance moves the projected endpoint by exactly

$$\hat{\xi}(\zeta) - \hat{\xi}(0) = -\zeta\, Mg. \tag{9}$$

M tends to the orthogonal projector onto the null space of A as $\varepsilon \to 0$, and $MA^\mathsf{T} = \varepsilon A^\mathsf{T}(AA^\mathsf{T} + \varepsilon I)^{-1}$. With $r = h(\xi)/\|h(\xi)\|$ and $\partial_u \xi = I + (1-\tau)\, \partial_u v_\theta$, the gradient (4) is $g = A^\mathsf{T} r + (1-\tau)(\partial_u v_\theta)^\mathsf{T} A^\mathsf{T} r$, so

$$Mg = (1-\tau)\, M\, (\partial_u v_\theta)^\mathsf{T} A^\mathsf{T} r + \varepsilon A^\mathsf{T}(AA^\mathsf{T} + \varepsilon I)^{-1} r. \tag{10}$$

The last term has norm at most $\varepsilon\|r\|/s_{\min}$ if the nonzero singular values of A are at least $s_{\min}$, and at most $\sqrt{\varepsilon}\|r\|/2$ without that assumption; on the Grams of our experiments, whose condition numbers reach 3×10^7, the second bound is the relevant one. The direct term $A^\mathsf{T} r$ lies in the row space of A, and the projection removes it up to that damping term. Up to damping, the surviving displacement

is therefore $(1-\tau)M(\partial_u v_\theta)^\mathsf{T} A^\mathsf{T} r$. It acts only in directions the linear constraint does not fix, so it can change the next network prediction while leaving $A\hat{\xi}$ unchanged as $\varepsilon \to 0$. The minimum-norm correction (2) ($B = I$ in data-assimilation terms) does not move these directions, and neither does a gradient that holds v_θ fixed, as in DiffusionPDE. Because of the charge (5), the next state moves by $-\zeta\big[(1-\tau')g/(1-\tau) + \tau' Mg\big]$, so the direct term also reaches the next network evaluation and is pulled back through the network before the next projection.

For nonlinear h the projection is the map $P(\xi) = \xi - J^\mathsf{T}\lambda$ with $\lambda = (JJ^\mathsf{T} + \varepsilon I)^{-1} h(\xi)$, and the guidance moves its output by $-\zeta\, DP(\xi)\, g + O(\zeta^2)$ with

$$DP = M\,(I - H_\lambda) + J^\mathsf{T}(JJ^\mathsf{T} + \varepsilon I)^{-1} K_\lambda, \qquad H_\lambda = \textstyle\sum_i \lambda_i \nabla^2 h_i, \quad (K_\lambda)_i = (J^\mathsf{T}\lambda)^\mathsf{T} \nabla^2 h_i. \tag{11}$$

The linear result is the first term with $H_\lambda = 0$. The other terms scale with the multipliers and the constraint curvature, so the linear picture holds near the constraint set or for mildly curved constraints; far from it on strongly curved constraints the projection rotates and rescales the surviving guidance by an amount (10) does not control. Kinetic energy, a quadratic observable projected late on residuals of order one, is such a case.

3.3 Noisy observations and schedules

When the observation errors are independent with variance σ^2, expressed in the units of the residual, we set $\varepsilon_\tau = \sigma^2$. The projection (6) then has the form of the analysis update $\tilde{\xi} - J^\mathsf{T}(JJ^\mathsf{T} + R)^{-1} h$ with $R = \sigma^2 I$ and background covariance $B = I$. It does not interpolate the noisy observations exactly, and it does not sample the posterior, since $B = I$ has no physical scale and the guidance term is not a likelihood gradient. GOSP guides and then projects, so it needs a projection that is usable at the guided prediction (Figure 1c). For strongly nonlinear constraints this fails early in sampling: a Gauss–Newton step linearised at an early prediction can overshoot. For kinetic energy the first step raises the residual from 7.0 to 142 and moves the state by 124% of its norm. We found no schedule that keeps these early corrections stable, and GOSP does not change this. The projection therefore starts at τ_0, with guidance alone before it. Its damping ε_τ may decrease during sampling. For kinetic energy $\tau_0 = 0.95$ with three Gauss–Newton steps; for streamline crossings ε_τ decreases from 10^{-1} to 10^{-6} after $\tau = 0.5$ while the root search window shrinks from 16 to 4 cells. These are the schedules we use for PCFM on these sets, fixed before GOSP; Appendix F gives the traces behind them.

3.4 The projection in observation space

The released implementation of PCFM builds J in (2) by reverse-mode differentiation, with $O(mn)$ memory and $O(m^2 n)$ time per correction. The step needs only $J^\mathsf{T} z$, a vector–Jacobian product, and $J(J^\mathsf{T} z)$, a Jacobian–vector product of it. With these, batched over samples, the conjugate-gradient (CG) method solves (2) for any residual in $O(n+m)$ memory per sample, and makes complete runs faster (Table 15). Its iteration count grows with the conditioning of the Gram, and simple row scaling does not reduce it on our sets. Throughout, the initial condition is imposed by restricting the variables to the slices $t \geq 1$, which removes its identity block from J for every solver. For a narrower class of constraints the Gram itself has closed form.

Consider, at one time slice, a field w on a periodic grid with N cells a side and r translation-invariant linear operators $L_1, \dots, L_r$, each acting on one channel $\mathrm{ch}(a)$ with Fourier multiplier μ_a: $\widehat{L_a w} = \mu_a\, \widehat{w}$. Derivatives, the inverse Laplacian and their products give velocity, strain and the divergent part of a vector field. A *kernel-Gram constraint* samples $z_a = L_a w$ at points $x_1, \dots, x_m$ and applies a pointwise nonlinearity,

$$h_i(w) = f\big(z_1(x_i), \dots, z_r(x_i)\big) - y_i, \qquad i = 1, \dots, m, \tag{12}$$

for example kinetic energy $|u|^2$ with $u = \nabla^\perp \psi$ and $-\Delta\psi = w$. Let K_a be the kernel of L_a and $g_a(x_i) = \partial f / \partial z_a$ at $z(x_i)$. The rows of the Jacobian are $J_i = \sum_a g_a(x_i)\, K_a(x_i - \cdot)$, and Parseval's identity turns the inner product of two kernel translates into a kernel evaluated at their offset:

$$(JJ^\mathsf{T})_{ij} = \sum_{a,b\,:\,\mathrm{ch}(a)=\mathrm{ch}(b)} g_a(x_i)\, g_b(x_j)\, G_{ab}(x_i - x_j), \qquad G_{ab} = \mathcal{F}^{-1}\big[\mu_a\, \overline{\mu_b}\big]. \tag{13}$$

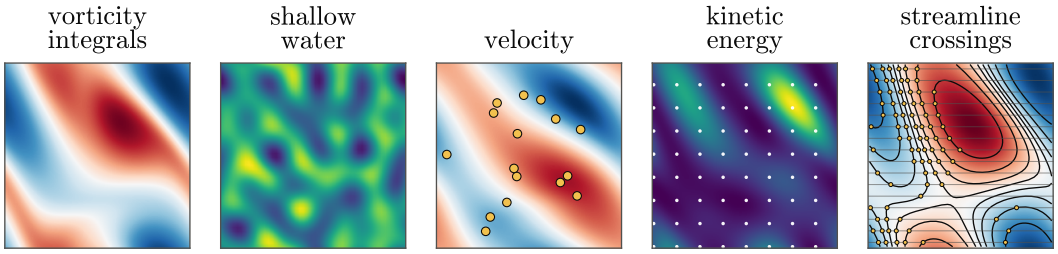


Figure 3: The five constraint sets on held-out reference fields, with their sampling locations: every eighth grid point for kinetic energy, the 16 velocity sensors, and the crossings of 8 contours of ψ with 16 lines. The vorticity-integral and shallow-water sets constrain integrals over each level.

Identity (13) is the observation-space system $(HBH^{\mathsf{T}} + R)\lambda = d$ of data assimilation with $H = J$, $B = I$, $R = \varepsilon I$ and misfit $d = h$, written for kernel-Gram constraints. We solve the projection (6) in the m-dimensional constraint space, one block per time level, with pointwise weights in closed form and no automatic differentiation. G_{ab} is computed once by one inverse fast Fourier transform (FFT) per pair, each entry of (13) is a look-up, and $J^{\mathsf{T}}\lambda$ is a scatter and one FFT per operator. The solve reproduces the correction assembled by automatic differentiation to 2.3×10^{-12} in double precision (Table 16). Rows at different time levels touch different slices, so the Gram is block diagonal in time. With m_t constraints per level, forming the block is $O(m_t^2 r^2)$ and factoring it $O(m_t^3)$; everything else is $O(rn \log n)$ time and $O(n)$ memory. For linear constraints the block does not depend on the state and is factored once for the whole run.

The same construction handles sensors off the grid and sampling points that move with the state, such as the positions where streamlines cross fixed lines, and an image construction extends it to Dirichlet rectangles. What it needs is pointwise functions of translation-invariant operators. Residuals on a general domain, or defined through a solver, fall back to the matrix-free projection (Appendix C.2).

4 Experiments

All runs use the frozen checkpoint of PCFM's Navier–Stokes benchmark ($64 \times 64 \times 50$, $\nu = 10^{-3}$) and a shallow-water model ($48^2 \times 25$) that we trained with the same architecture on data we generated. Sampling is in float32 on one NVIDIA H100, with 100 Euler steps, $\varepsilon = 10^{-6}$ and one Gauss–Newton step per correction unless stated. Every method draws 100 samples from the same prior draws.

We use five constraint sets (Figure 3). The vorticity-integral set is PCFM's Navier–Stokes benchmark: the initial vorticity and the total vorticity of each level. Shallow water imposes the initial state and the mass, momenta and energy of each level. *Velocity* reads both components of u at 16 off-grid sensors on five levels. *Kinetic energy* $|u|^2$ is read every eighth grid point at $t = 49$. *Streamline crossings* are where 8 contours of ψ cross 16 lines on five levels, so the points move with the state. For the last four sets the targets come from one held-out reference trajectory.

The baselines are PCFM and ECI with the kernel-Gram projection, DPS and DiffusionPDE with step ζ, and D-Flow with twenty L-BFGS iterations and learning rate η; every method holds the initial condition by restriction. ζ (GOSP: 0 to 30; DPS: 0.01 to 30) and $\eta \in \{0.01, 0.1, 1\}$ are chosen on 10 samples from a different prior seed with the same reference. The criterion is the median trajectory error over all samples (MSE for the vorticity integrals), and is the same for every method. Because this choice sees the reference, the tuned results are an upper bound. A reference-free choice of ζ and 15 held-out settings without re-tuning give the deployable picture. GOSP's projection uses PCFM's schedule of each set. DiffusionPDE and all tuning runs are in Appendix D.1.

MSE is the mean squared error of the ensemble mean against the ground-truth mean, over grid points; for the vorticity integrals the ground truth is the 100 reference trajectories that share the initial vorticity, for the other sets the pinned reference. A sample is *in range* if it is finite and below 10^3 in magnitude, and metrics are over samples in range. The trajectory error of a sample is $\|u - u_{\text{ref}}\|^2 / \|u_{\text{ref}}\|^2$; a sample is *close to the reference* if this error is below 0.1. The residual is

Table 1: Mean squared error (MSE) of the ensemble mean of 100 samples against the ground truth, on the five sets, same prior draws; in brackets, below a value, the samples in range where not all are. PCFM: projection at every step. ECI: PCFM with a fresh prior draw at every step. D-Flow: learning rate tuned against the reference. DPS and GOSP: step ζ tuned against the reference (Appendix D.1) and chosen by the reference-free rule of §4.3 (no ref.). MSE over the samples every method keeps in range: Table 7. The text quotes the tuned runs.

				DPS		GOSP	
set	PCFM	ECI	D-Flow	tuned	no ref.	tuned	no ref.
vorticity integrals	1.7e−1	2.1e−1	1.4e−1	1.9e−1	2.0e−1	1.7e−1	1.5e−1
shallow water	2.0e−3	4.5e−4	3.4e−3	6.8e−3	7.8e−3 (98)	2.0e−3	2.1e−3
velocity, $P = 16$	1.1e−3	1.0e−3	4.8e−3 (88)	9.9e−5	2.2e−2	4.4e−5	6.7e−5
kinetic energy	4.3e−1	6.0e−1	4.6e−2 (88)	2.7e−3	2.7e−3	2.6e−3	2.6e−3
streamline crossings	2.0e−4 (54)	2.9e−4 (49)	2.4e−1 (80)	2.2e−2 (96)	1.2e−1	4.7e−5 (95)	4.7e−5 (95)

the norm of h at the final sample. Streamline residuals are recomputed on the final sample by an independent nearest-root search on the whole line, a missing root counting as a failure. The sampler's own windowed residual sets the residual of crossings it misses to zero, and is used only inside the projection.

4.1 Constrained sampling

On the sparse sets the projection fixes the measured quantities but does not determine the unobserved field, and this is where the charge matters. Guidance recovers much more of the unobserved field than projection does, and GOSP recovers more still while keeping the projection's residual: on the 16 velocity sensors the MSE of the ensemble mean falls from 1.1×10^{-3} under PCFM to 4.4×10^{-5}. Its guidance changes the directions the sensors leave free, and the charge keeps that change through every projection. A final projection of DPS's samples removes their sensor residual without changing their MSE. Streamline crossings add sampling points that move with the state. There 95 GOSP samples are close to the reference, against 62 for DPS, and an independent root search places every crossing within half a cell of its target in 95 GOSP samples and in no DPS sample.

The integral sets leave less to complete. Tuning finds little for guidance to add and selects $\zeta = 0$ on the vorticity integrals and 0.1 on shallow water, where GOSP matches PCFM; guidance alone is less accurate on both, and on shallow water ECI's resampling of the prior draw gives the lowest MSE.

For kinetic energy the projection starts only at $\tau_0 = 0.95$ (§3.3), and no PCFM sample is close to the reference. GOSP does not remove the instability. Before τ_0 it takes DPS's steps and ends as accurate as DPS, with MSE 2.6×10^{-3} against 2.7×10^{-3}; its late projection then meets the energy to 2.0×10^{-6}, where DPS leaves 0.4. Charging the stored prior draw on the early steps as well raised the MSE to 0.014, with the error on the unconstrained levels (Appendix E.3).

4.2 Sparse, noisy sensors with a free initial state

The experiments above treat the observations as exact. Real sensors are noisy, and interpolating their readings exactly is then undesirable. In this setting the initial slice is left to the prior, and the reference is a held-out trajectory whose initial vorticity the model never saw. P off-grid sensors read both velocity components at five levels with Gaussian noise of standard deviation σ times the root mean square of the clean readings. We evaluate 20 held-out references, which differ in initial vorticity, forcing, sensor layout and noise, and each method draws 32 samples from matched prior draws. Because none of the methods is designed to sample $p(u \mid y)$, we score the ensembles as probabilistic forecasts of the true trajectory. The continuous ranked probability score (CRPS) (Hersbach, 2000), in its fair ensemble form (Ferro, 2014) averaged over grid values, is lower for better ensembles (formulas

Table 2: Sparse noisy velocity sensors, free initial state: median continuous ranked probability score (CRPS; lower is better) over 20 held-out references, 32 samples each, and median spread–skill ratio (below one: under-dispersed). *Some samples out of range. All methods and metrics: Table 8.

	CRPS						spread–skill ratio	
	PCFM							
sensors, noise	10^{-6}	σ^2	ECI	D-Flow	DPS	GOSP	DPS	GOSP
$P = 16, \sigma = 0.1$	0.032^*	0.051	0.034	0.165^*	0.021	0.026	0.69	1.12
$P = 16, \sigma = 0.3$	0.073^*	0.199	0.140	0.145^*	0.045	0.044	0.62	1.20
$P = 4, \sigma = 0.1$	0.094^*	0.187^*	0.134^*	0.056^*	0.038	0.035	0.48	0.97
$P = 4, \sigma = 0.3$	0.095^*	0.282	0.236	0.065^*	0.076	0.078	0.34	0.71

Table 3: Samples close to the reference (trajectory error below 0.1), of 100; in brackets those in range where not all are. Main reference: the runs of Table 1, one change at a time. 15 settings: held-out references and seeds, ζ not re-tuned, median [min, max]; — not run. Full metrics: Tables 9 and 11.

	main reference			15 settings		
sampler	velocity	energy	streamline	velocity	energy	streamline
PCFM	100	0	54 (54)	100 [99, 100]	0 [0, 1]	61 [45, 77]
DPS	100	92	62 (96)	100 [92, 100]	80 [30, 94]	61 [31, 82]
DPS, likelihood step	66 (73)	61	28	—	—	—
GOSP	100	93	95 (95)	100 [100, 100]	79 [29, 94]	90 [73, 95]
GOSP, no charge to u_0	100	92	55 (55)	—	—	—
GOSP, $v_\theta(\tilde{u})$ recomputed	100	83 (94)	90 (90)	—	—	—
GOSP, likelihood step	100	62	89 (89)	—	—	—
Euler-form combination	100	92	94	—	—	—

in Appendix E.1). The spread–skill ratio, the ensemble spread divided by the root-mean-square error (RMSE) of the ensemble mean (Fortin et al., 2014), is below one for an under-dispersed ensemble and above one for an over-dispersed one; a ratio near one is necessary for calibration but not sufficient. PCFM is run with its own $\varepsilon = 10^{-6}$ and with $\varepsilon = \sigma^2$; ECI and GOSP use $\varepsilon = \sigma^2$. ζ and η are chosen by CRPS on four other references.

PCFM with $\varepsilon = 10^{-6}$ interpolates the noise and loses samples, as few as 173 of 640 in range; with $\varepsilon = \sigma^2$ it keeps them but is over-dispersed (Table 2). DPS is under-dispersed in every configuration (spread–skill ratio 0.34–0.69). GOSP's CRPS is within a factor 0.93–1.23 of DPS's, lower in two of the four configurations, with a spread–skill ratio of 0.71–1.20. GOSP's damped projection stops short of the readings and keeps a spread closer to its error. Neither ingredient does that alone: PCFM at the same $\varepsilon = \sigma^2$ and without guidance is over-dispersed (spread–skill ratio 1.24–1.90), and GOSP with the guidance but $\varepsilon = 10^{-6}$ ranges over 0.52–1.80 and loses samples at $\sigma = 0.3$ (Table 8). D-Flow is best with 4 sensors and $\sigma = 0.3$, at 350 s per reference against 14 s.

4.3 Ablations and robustness

The gains above rest on charging the source sample and on a few sampling choices; Table 3 changes one at a time. Without the charge the guided runs fall back to PCFM's numbers, as the linear analysis of §3.2 predicts: velocity MSE is 8.7×10^{-4}, and 55 samples are close to the reference on streamline crossings. On kinetic energy, where the charge acts only in the last steps, removing it changes little (92 against 93).

The charge leaves the network's input close to what it saw in training: the draw implied by the state at which the network is evaluated, $u_\tau - \tau v_\theta(u_\tau, \tau)$, which is (8) taken before the guidance step and equals u_0 on the training path, keeps an energy per entry within 0.99–1.01 of the prior's up to $\tau = 0.9$ on every set, as under DPS, which never moves the prior draw. Recomputing v_θ after the guidance step costs an extra network pass and does not improve the results; on kinetic energy it loses samples.

Table 4: Velocity at P sensors on $128^2 \times 50$ fields ($n = 802{,}816$) with the 128^2 model, 100 samples (10 for CG). Close: trajectory error below 0.1. Projection: time per sample. CG: matrix-free conjugate gradients with the stated cap on iterations per correction (mean used in brackets). Residual: norm of the sensor residual. Out of memory: on an 80 GB GPU.

P	m	solver	close	projection	peak	residual
2048	20529	kernel Gram	100/100	526 ms	15.2 GB	1.7e−5
		CG, 32 it.	10/10	7.21 s	14.5 GB	0.025
		CG, 128 it.	10/10	27 s	14.5 GB	4.2e−4
		CG, $\leq$ 2048 it. (mean 398)	10/10	89.2 s	14.5 GB	1.7e−5
		dense, reassembled or factored once		out of memory		
4096	41009	kernel Gram	100/100	1.04 s	15.6 GB	2.6e−5
		CG, 32 it.	10/10	7.23 s	14.5 GB	0.11
		CG, 128 it.	10/10	27.6 s	14.5 GB	2.1e−3
		CG, $\leq$ 2048 it. (mean 709)	10/10	159 s	14.5 GB	2.6e−5
		dense, reassembled or factored once		out of memory		

Replacing the normalised residual step by the likelihood gradient $\nabla\|h\|^2/2$ is worse on every set. Full metrics, the implied-noise diagnostics and the recomputed-velocity sweep are in Appendix E.2.

The advantage persists without re-tuning, across held-out references and sensor layouts. Over the 15 held-out settings, with ζ fixed, an independent root search finds every streamline crossing within half a cell in a median of 90 [73, 95] GOSP samples, against 60 for PCFM and 0 for DPS. This is stricter than being close to the reference, which 61 DPS samples are (Table 3): a sample can be close and still have crossings displaced or missing. GOSP's velocity MSE and kinetic-energy residuals are the lower of the two as well (Table 11). On a prior seed used for neither tuning nor evaluation, a wide range of ζ gives the same result (Figure 4), and a reference-free rule, the largest ζ whose share of samples in range is within 5 points of the best, selects the best ζ of the sweep on all three sets.

4.4 Scaling the projection to 128^2 fields

We fine-tuned PCFM's 64^2 neural operator on 128^2 Navier–Stokes data and projected onto $P = 2048$ and 4096 velocity sensors ($m = 20{,}529$ and $41{,}009$). Dense projection, whether reassembled at every step or factored once, runs out of memory (Table 4). Matrix-free CG fits in memory. With 32 or 128 iterations per correction it leaves a residual of 4.2×10^{-4}–0.11; with a cap of 2048 it reaches the kernel Gram's residual at 89.2 s and 159 s per sample, with or without row scaling. The kernel Gram takes 526 ms and 1.04 s per sample, 153–170 times less.

5 Conclusion

We propose GOSP, an inference-time method for constrained generation with pretrained flow models. Unlike existing approaches that use either gradient guidance or projection alone, GOSP combines the two by charging the guided displacement to the source sample, allowing the effect of guidance to persist across repeated projections while maintaining constraint satisfaction. We further formulate the projection in observation space, avoiding explicit construction of the constraint Jacobian through matrix-free and structured kernel-Gram solvers. Across Navier–Stokes and shallow-water benchmarks, GOSP substantially improves reconstruction from sparse and moving observations over projection- and guidance-based baselines while retaining small constraint residuals, and the kernel-Gram solver enables projection at scales where dense implementations run out of memory. These results suggest that GOSP can provide a practical framework for incorporating task-specific observations and physical constraints into pretrained generative models, with potential applications in sparse-sensor reconstruction, data assimilation, inverse problems, and physics-constrained generation.

AI Assistance Disclosure. In this work, we used generative AI tools to assist with editing the manuscript for clarity and readability and with editing, debugging, and refactoring software code written by the authors. We did not use generative AI tools to generate synthetic datasets, develop

the theoretical or conceptual framework, formulate or prove mathematical claims, implement the proposed methods, or design or interpret the experiments. All AI-assisted text was reviewed and revised by the authors, and all AI-assisted code was manually inspected and tested for correctness. Numerical implementations were additionally checked against independent implementations where applicable, and all reported quantitative results were verified against stored experimental outputs. We take responsibility for the final content of this work, including text, claims, code, and artifacts produced with the aid of generative AI.

REFERENCES

Michael Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

Heli Ben-Hamu, Omri Puny, Itai Gat, Brian Karrer, Uriel Singer, and Yaron Lipman. D-flow: Differentiating through flows for controlled generation. *arXiv preprint arXiv:2402.14017*, 2024.

Andrew F Bennett. *Inverse methods in physical oceanography*. Cambridge university press, 1992.

Chaoran Cheng, Boran Han, Danielle Maddix, Abdul Fatir Ansari, Andrew Stuart, Michael W Mahoney, and Bernie Wang. Gradient-free generation for hard-constrained systems. In *International Conference on Learning Representations*, volume 2025, pp. 100510–100539, 2025.

Jacob K Christopher, Stephen Baek, and Ferdinando Fioretto. Constrained synthesis with projected diffusion models. *Advances in Neural Information Processing Systems*, 37:89307–89333, 2024.

Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022a.

Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. *Advances in neural information processing systems*, 35:25683–25696, 2022b.

Stephen E Cohn, Arlindo Da Silva, Jing Guo, Meta Sienkiewicz, and David Lamich. Assessing the effects of data selection with the dao physical-space statistical analysis system. *Monthly Weather Review*, 126(11):2913–2926, 1998.

Philippe Courtier. Dual formulation of four-dimensional variational assimilation. *Quarterly Journal of the Royal Meteorological Society*, 123(544):2449–2461, 1997.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Christopher AT Ferro. Fair scores for ensemble forecasts. *Quarterly Journal of the Royal Meteorological Society*, 140(683):1917–1923, 2014.

Vincent Fortin, Mabrouk Abaza, Francois Anctil, and Raphael Turcotte. Why should ensemble spread match the rmse of the ensemble mean? *Journal of Hydrometeorology*, 15(4):1708–1713, 2014.

Zachary Frangella, Joel A Tropp, and Madeleine Udell. Randomized nyström preconditioning. *SIAM Journal on Matrix Analysis and Applications*, 44(2):718–752, 2023.

Hans Hersbach. Decomposition of the continuous ranked probability score for ensemble prediction systems. *Weather and Forecasting*, 15(5):559–570, 2000.

Jiahe Huang, Guandao Yang, Zichen Wang, and Jeong Joon Park. Diffusionpde: Generative pde-solving under partial observation. *Advances in Neural Information Processing Systems*, 37: 130291–130323, 2024.

Bowen Jing, Bonnie Berger, and Tommi Jaakkola. Alphafold meets flow matching for generating protein ensembles. *arXiv preprint arXiv:2402.04845*, 2024.

Gavin Kerrigan, Giosue Migliorini, and Padhraic Smyth. Functional flow matching. *arXiv preprint arXiv:2305.17209*, 2023.

Jeongsol Kim, Bryan Sangwoo Kim, and Jong Chul Ye. Flowdps: Flow-driven posterior sampling for inverse problems. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12328–12337. IEEE, 2025.

Dana A Knoll and David E Keyes. Jacobian-free newton–krylov methods: a survey of approaches and applications. *Journal of Computational Physics*, 193(2):357–397, 2004.

Alaina Kolli, Theodoros Xenakis, Utkarsh Utkarsh, Pengfei Cai, Rafael Gomez-Bombarelli, Alan Edelman, and Christopher Vincent Rackauckas. Snap-fm: Sparse nonlinear accelerated projection for physics-constrained generative modeling. *arXiv preprint arXiv:2607.00095*, 2026.

Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, et al. Voicebox: Text-guided multilingual universal speech generation at scale. *Advances in neural information processing systems*, 36:14005–14034, 2023.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

Jae Hyun Lim, Nikola B Kovachki, Ricardo Baptista, Christopher Beckham, Kamyar Azizzadenesheli, Jean Kossaifi, Vikram Voleti, Jiaming Song, Karsten Kreis, Jan Kautz, et al. Score-based diffusion models in function space. *Journal of Machine Learning Research*, 26(158):1–62, 2025.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Jianming Ma, Qiyue Yang, Yang Zhang, Liyun Yan, Zhanxiang Cao, Yazhou Zhang, and Yue Gao. Polyflow: Safe and efficient polytope-constrained flow matching with constraint embedding and projection-free update. *arXiv preprint arXiv:2606.13400*, 2026.

Ashwini Pokle, Matthew J Muckley, Ricky TQ Chen, and Brian Karrer. Training-free linear image inverses via flows. *arXiv preprint arXiv:2310.04432*, 2023.

Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pp. 63–71. Springer, 2003.

François Rozet and Gilles Louppe. Score-based data assimilation. *Advances in Neural Information Processing Systems*, 36:40521–40541, 2023.

Yousef Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.

Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International conference on learning representations*, 2023.

Utkarsh Utkarsh, Pengfei Cai, Alan Edelman, Rafael Gomez-Bombarelli, and Christopher Rackauckas. Physics-constrained flow matching: Sampling generative models with hard constraints. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 160217–160252. Curran Associates, Inc., 2025. doi: 10.52202/085713-5354. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/eacce1f7d11e2c3a7568468e0eff5d33-Paper-Conference.pdf.

Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. *arXiv preprint arXiv:2212.00490*, 2022.

Yuanzhi Zhu, Kai Zhang, Jingyun Liang, Jiezhang Cao, Bihan Wen, Radu Timofte, and Luc Van Gool. Denoising diffusion models for plug-and-play image restoration. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1219–1229. IEEE, 2023.

A Related work

Constrained generation with diffusion and flow models uses gradient guidance (Chung et al., 2022a; Huang et al., 2024), trajectory optimisation (Ben-Hamu et al., 2024), projection at every denoising step (Christopher et al., 2024), projection-free polytope constraints (Ma et al., 2026), and gradient-free hard constraints (Cheng et al., 2025). PCFM (Utkarsh et al., 2025) handles nonlinear, coupled constraints by Gauss–Newton projection. GOSP adds model-aware guidance to PCFM's projection. SNAP-FM (Kolli et al., 2026), concurrent with this work, speeds up PCFM's projection by exploiting a sparse Jacobian; our Jacobians are dense, with Grams structured by translation invariance (Appendix H).

Combining a gradient step with measurement consistency is standard in diffusion inverse solvers. Manifold-constrained gradients follow the likelihood gradient by a measurement projection at every step (Chung et al., 2022b); the denoising null-space model keeps the range-space part of a linear measurement and leaves the null space to the model (Wang et al., 2022); DiffPIR alternates a proximal data step with a denoising step (Zhu et al., 2023); and pseudoinverse guidance replaces the likelihood gradient by a pseudoinverse-weighted one (Song et al., 2023). The same constructions have been carried to flow models (Pokle et al., 2023; Kim et al., 2025). Two things differ here. Their measurement operators are linear with a known pseudoinverse, and for linear h with $\varepsilon \to 0$ the step (2) is the null-space decomposition of Wang et al. (2022); the constraints in this paper are nonlinear and coupled, and in one set the sampling points move with the state. More importantly, those samplers advance the state with the flow's or the diffusion's own update, which carries a guided displacement into the next step, so only the component along the measurement rows is removed. PCFM instead rebuilds the state by interpolating from the stored prior draw, which removes the rest. The erasure this paper is about is therefore a property of a re-interpolating sampler, and the charge repairs the combination inside it rather than combining guidance and projection for the first time.

In data assimilation, score-based data assimilation (Rozet & Louppe, 2023) trains a score model on short trajectory segments and assimilates observations by posterior sampling over a whole trajectory, learning the prior for that task; we condition a pretrained flow model at inference instead. The matrix (13) appears in the dual formulation of 3D- and 4D-Var (Courtier, 1997; Cohn et al., 1998), in the representer method (Bennett, 1992) and in Gaussian-process regression (Rasmussen, 2003). Our matrix-free projection follows Jacobian-free Newton–Krylov methods (Knoll & Keyes, 2004; Saad, 2003).

B Discussion and limitations

GOSP has no target distribution. The charged source is not a draw from π_0, the final samples depend on ζ, the schedule and the projection, and none of our diagnostics establishes that they follow a conditional law. This is why we score the noisy-sensor ensembles as forecasts rather than as posterior samples: $\varepsilon = \sigma^2$ there is a damping chosen to avoid interpolating the readings, and the spread–skill ratios near one are an empirical consequence of it. On the pinned-reference sets we score closeness to one trajectory, so how much variability of the unconstrained field the ensembles keep is not measured.

What the method can do is bounded by the projection it wraps. On pointwise kinetic energy an early Gauss–Newton step overshoots; the projection starts at $\tau_0 = 0.95$, GOSP reduces to guidance followed by a few projections, and it is then only as accurate as DPS. We found no schedule that makes early corrections stable there (Appendix F.2), and none of the starting times or damping schedules, which we took from PCFM runs, comes with a rule for a new constraint. The guidance step has its own free parameter: too large a ζ makes guidance diverge (Figure 4), the tuned results in Table 1 choose it against the reference trajectory, and its coupling with the damping ε_τ is unexplored. Each step also costs one backward pass through the network, and on shallow water ECI remains the more accurate sampler.

The solvers and the evidence behind them are narrower than the construction. The kernel Gram needs pointwise functions of translation-invariant operators on a periodic grid or a Dirichlet rectangle, and everything else goes to the matrix-free solver, whose cost grows with the conditioning of the Gram and which at 128^2 was two orders of magnitude slower. The evaluation rests on one Navier–Stokes model, whose data vary only the forcing phase for a given initial vorticity, and one shallow-water model, which fails without its conservation rows; the Dirichlet images are checked against automatic

differentiation but not run in a sampler, and SNAP-FM is compared by structure only (Appendix H). The charge itself needs only a sampler whose state is a known combination of a source variable and a prediction of the final sample: for an interpolant $x_\tau = \alpha_\tau x_0 + \beta_\tau x_1$ the same bookkeeping charges $-\zeta g/\alpha_\tau$ to the source, and deterministic diffusion samplers that form a noise estimate admit the analogous update. We have tested it only on the linear flow-matching path.

C METHOD DETAILS

C.1 DERIVATIONS

Identity (13)**.** Row i of J is $\sum_a g_a(x_i)\,\partial z_a(x_i)/\partial w = \sum_a g_a(x_i)\,K_a(x_i - \cdot)$ by the chain rule, with $K_a = \mathcal{F}^{-1}[\mu_a]$ the kernel of L_a. Hence $(JJ^\mathsf{T})_{ij} = \sum_{a,b} g_a(x_i) g_b(x_j) \sum_y K_a(x_i - y) K_b(x_j - y)$, and the inner sum is the cross-correlation of two kernels at offset $x_i - x_j$, whose transform is $\mu_a \overline{\mu_b}$. Operators on different channels, and rows on different time slices, do not pair.

Off-grid, moving and bounded points. The rows stay weighted kernel translates. For bilinear interpolation, $z_a(x_i) = \sum_c \beta_{ic} z_a(q_{ic})$ over four corners, and the entry becomes $\sum_{c,d} \beta_{ic}\beta_{jd} G_{ab}(q_{ic} - q_{jd})$. Differentiating $q\big((Lw)(x^\star), x^\star\big) = 0$ gives (14). On a Dirichlet rectangle the odd extension E to the doubled periodic grid commutes with the discrete Laplacian and central differences, so $J = J_{\text{per}} E$ and $EE^\mathsf{T} = \sum_\sigma \operatorname{sign}(\sigma) R_\sigma$, which turns each inner product into four reflected cross-correlations, (15).

Properties of the GOSP update. With $\zeta = 0$, (5) changes nothing and, on steps with a projection, (6)–(7) are PCFM's steps. On a step without one, $(1-\tau')(\tilde{u} - \tau v) + \tau'(\tilde{u} + (1-\tau)v) = \tilde{u} + \Delta\tau\, v$, the Euler step. With $\tilde{u} = u_\tau - \zeta g$ and $v = v_\theta(u_\tau, \tau)$ this is $u_\tau + \Delta\tau\, v_\theta(u_\tau, \tau) - \zeta g$, which is DPS's step (3): both evaluate the velocity and the gradient at u_τ and add the guidance to the state, and both keep the held initial-condition slice. By (5) and (7), $u_\tau = (1-\tau)u_0 + \tau\hat{\xi}_{\text{prev}}$ holds at every step. The charge of step k enters $u_{\tau'_j}$ through the prior draw with weight $(1-\tau'_j)/(1-\tau_k) \le 1$, and u_1 with weight zero. Reusing $v_\theta(u_\tau, \tau)$ in the look-ahead changes it by $-(1-\tau)\zeta\, \partial_u v_\theta\, g + O(\zeta^2)$.

Nonlinear residuals, equation (11)**.** Write $G = JJ^\mathsf{T} + \varepsilon I$ and $\lambda = G^{-1}h$, all at ξ. Differentiating $P(\xi) = \xi - J^\mathsf{T}\lambda$ along $d\xi$ gives $dP = d\xi - (dJ)^\mathsf{T}\lambda - J^\mathsf{T} d\lambda$ with $d\lambda = G^{-1}\big(J\,d\xi - (dJ\,J^\mathsf{T} + J\,dJ^\mathsf{T})\lambda\big)$. With $(dJ)^\mathsf{T}\lambda = \sum_i \lambda_i \nabla^2 h_i\, d\xi = H_\lambda d\xi$, and $(dJ\,J^\mathsf{T}\lambda)_i = (J^\mathsf{T}\lambda)^\mathsf{T}\nabla^2 h_i\, d\xi = (K_\lambda d\xi)_i$, we get $dP = \big(I - J^\mathsf{T}G^{-1}J\big)d\xi - \big(I - J^\mathsf{T}G^{-1}J\big)H_\lambda d\xi + J^\mathsf{T}G^{-1}K_\lambda d\xi$, which is (11) with $M = I - J^\mathsf{T}G^{-1}J$. For linear h the Hessians vanish and $DP = M$. The guided endpoint is $P(\xi - \zeta g) = P(\xi) - \zeta DP(\xi) g + O(\zeta^2)$ when P is twice differentiable along the segment, so the linear statement holds to first order in ζ with M replaced by DP.

Linear residuals, equations (9) **and** (10)**.** For $h(\xi) = A\xi - y$ the damped step is $\hat{\xi} = \tilde{\xi} - A^\mathsf{T}(AA^\mathsf{T} + \varepsilon I)^{-1}(A\tilde{\xi} - y) = M\tilde{\xi} + A^\mathsf{T}(AA^\mathsf{T} + \varepsilon I)^{-1} y$, so $\hat{\xi}(\zeta) - \hat{\xi}(0) = -\zeta M g$ exactly, g being computed before the move. $MA^\mathsf{T} = A^\mathsf{T} - A^\mathsf{T}(AA^\mathsf{T} + \varepsilon I)^{-1}AA^\mathsf{T} = A^\mathsf{T}(AA^\mathsf{T} + \varepsilon I)^{-1}\big((AA^\mathsf{T} + \varepsilon I) - AA^\mathsf{T}\big) = \varepsilon A^\mathsf{T}(AA^\mathsf{T} + \varepsilon I)^{-1}$, whose norm is $\max_s \varepsilon s/(s^2 + \varepsilon)$ over the singular values s of A. This is at most $\varepsilon/s_{\min}$ if the nonzero s are at least $s_{\min}$, and at most $\sqrt{\varepsilon}/2$ in general. The gradient of $\|h(u + (1-\tau)v_\theta(u,\tau))\|$ is $(\partial_u \xi)^\mathsf{T} A^\mathsf{T} r$ with $\partial_u \xi = I + (1-\tau)\partial_u v_\theta$, which gives (10). The next state is $(1-\tau')\tilde{u}_0 + \tau'\hat{\xi}(\zeta)$, whose change from $\zeta = 0$ is $-\zeta\big[(1-\tau')g/(1-\tau) + \tau' M g\big]$.

C.2 CONSTRAINT SETS AND KERNEL-GRAM EXTENSIONS

All fields live on the periodic unit square with N cells a side, and every operator is applied by its Fourier symbol with $\widehat{\Delta} = -|2\pi k|^2$ and the Nyquist row and column zeroed. With $\Gamma = 1/|2\pi k|^2$ ($\Gamma(0) = 0$) the streamfunction is $\hat{\psi} = \Gamma\hat{w}$, so $-\Delta\psi = w$, and $u = \nabla^\perp\psi = (\partial_y\psi, -\partial_x\psi)$, matching the solver that generated the Navier–Stokes data.

Integral sets. The vorticity integrals impose the initial vorticity and the total vorticity of each level; shallow water imposes the initial state and the mass, both momenta and energy of each level. After

restriction each row is an integral over one slice, a kernel-Gram constraint with a single sampling functional, and the Gram has at most four rows per level at every resolution.

Velocity at sensors. Both components of u, bilinearly interpolated at P uniformly random positions fixed for the run and matched to the reference at $t \in \{10, 20, 30, 40, 49\}$, together with the total-vorticity rows: $m = 10P + 49$. The interpolation is a fixed sparse matrix $\mathcal{I}$ with four entries per row, so the block is $\mathcal{I} G_{ab} \mathcal{I}^{\mathsf{T}}$, formed and factored once for the run.

Kinetic energy. $h(x,t) = \big(|u(x,t)|^2 - |u_{\text{ref}}(x,t)|^2\big)/c$ on a subgrid of stride s, with c the root mean square of the target; two operators, u and v, and $f = z_1^2 + z_2^2$. The set keeps the total-vorticity rows, which are orthogonal to the kinetic-energy rows because the symbols vanish at $k = 0$.

Streamline crossings. On each of L_ℓ lines $y = \eta_i = (i + 0.37)/L_\ell$ and at each of L_c contour values (quantiles 0.1 to 0.9 of the reference ψ at that level), the constraint is $h = (x^\star - x^\star_{\text{ref}})/\Delta x$ at five levels, where $x^\star$ is the root of the bilinear interpolant of $\psi - c_i$ along the line. With $q\big((Lw)(x_i^\star), x_i^\star\big) = 0$ and $d_i = \partial_z q \, \partial_x (Lw)(x_i^\star) + \partial_x q \neq 0$, the implicit function theorem gives

$$\frac{\partial x_i^\star}{\partial w} = -\frac{\partial_z q}{d_i} K(x_i^\star - \cdot), \qquad J_i = g_i \, K(x_i^\star - \cdot), \qquad g_i = -\phi'(x_i^\star) \, \frac{\partial_z q}{d_i}, \tag{14}$$

with $q(z, x) = z - c_i$ and $d_i = \partial_x \psi(x_i^\star, \eta_i)$, so points and weights are read off the state at every correction. The root is searched within a window around the reference crossing (four cells, wider early in the schedule of Appendix F.3) and differentiated by one Newton step. Crossings with slope below 0.2 of the median are dropped, crossings closer than two cells on one line are thinned to the steeper one, and there is at most one line per grid row, so no two rows nearly coincide. A crossing that leaves its search window during sampling is replaced by the reference location, with residual and Jacobian row set to zero; every reported residual is recomputed on the final sample by the independent root search of §4.

Dirichlet rectangles. On a box with $\psi = 0$ on the boundary, Δ^{-1} is not translation invariant, but the images give

$$(JJ^{\mathsf{T}})_{ij} = \sum_{a,b} g_a(x_i) \, g_b(x_j) \sum_{\sigma \in \{\pm 1\}^2} \operatorname{sign}(\sigma) \, C^\sigma_{ab}(x_i - R_\sigma x_j), \qquad C^\sigma_{ab} = \mathcal{F}^{-1}\big[\widehat{K_a} \, \overline{\widehat{K_b \circ R_\sigma}}\big], \tag{15}$$

four look-ups per entry. For kinetic energy on such a box this Gram and its correction agree with automatic differentiation of the bounded-domain code to 2.2×10^{-14} ($N = 16, 32, 64$); it has not been run in a sampler. Neumann conditions would need boundary corrections we have not derived.

Scope. Constraints through a Lagrangian map are also outside the class (§3.4).

C.3 Algorithms

Cost. Dense assembly holds $O(mn)$ and forms the Gram in $O(m^2 n)$; the kernel Gram replaces the factor n by a look-up. The block needs $O(m_t^2)$ memory, which on an 80 GB device limits m_t to a few tens of thousands per level.

C.3.1 Matrix-free projection

The step (2) needs only the action of JJ^{T}, and that action is one reverse-mode product followed by one forward-mode product, $G_\varepsilon z = J(J^{\mathsf{T}} z) + \varepsilon z$. Mapping both over the batch keeps the batched Jacobian block diagonal without storing it, and batched conjugate gradients with per-sample convergence masks solves the regularised system to a relative tolerance, holding $O(n+m)$ per sample. The Gram action agrees with the assembled one to machine precision. Rows of these Jacobians differ in scale by up to three orders of magnitude; where the row norms have closed forms they give an exact Jacobi (row-equilibration) preconditioner at $O(n)$ cost, applied with the stopping test kept on the unscaled residual. In our per-correction timings (supplementary material) the closed-form Gram was cheaper wherever the block fits, and the iteration count grew with the spread of row norms and

Algorithm 1 GOSP, one sampling run

Require: prior draw u_0, network v_θ, residual h, steps K, guidance ζ, damping schedule ε_τ, projection start τ_0

$u \leftarrow u_0$
for $k = 0, \dots, K-1$ **do**
  $\tau \leftarrow k/K$, $\tau' \leftarrow \tau + 1/K$; $v \leftarrow v_\theta(u, \tau)$
  $g \leftarrow \nabla_u \|h(u + (1-\tau)v_\theta(u,\tau))\|$ ▷ one backward pass through v_θ
  $u \leftarrow u - \zeta g$; $u_0 \leftarrow u_0 - \zeta g/(1-\tau)$; $\xi \leftarrow u + (1-\tau)v$ ▷ charge to the prior draw
  if $\tau \geq \tau_0$ **then** $\hat{\xi} \leftarrow \xi - J^\mathsf{T}(JJ^\mathsf{T} + \varepsilon_\tau I)^{-1} h(\xi)$ ▷ observation-space solve, §3.4
  else $\hat{\xi} \leftarrow \xi$; $u_0 \leftarrow \hat{\epsilon} = u - \tau v$, the initial-condition slice of u_0 kept ▷ implied draw (8): Euler step, as DPS
  end if
  $u \leftarrow (1-\tau')u_0 + \tau' \hat{\xi}$
end for
return $u - J^\mathsf{T}(JJ^\mathsf{T} + \varepsilon_1 I)^{-1} h(u)$

Algorithm 2 Kernel-Gram projection, one correction

Require: restricted state w, sampling points $\{x_i\}$ per level, multipliers $\{\mu_a\}$, damping ε

$G_{ab} \leftarrow \mathcal{F}^{-1}[\mu_a \overline{\mu_b}]$ for every pair on a common channel ▷ once
for each constrained level t **do**
  $z_a \leftarrow \mathcal{F}^{-1}[\mu_a \mathcal{F} w_t]$; $h_i \leftarrow f(z(x_i)) - y_i$; $g_a(x_i) \leftarrow \partial_a f(z(x_i))$ ▷ closed-form partials
  $\mathbf{G}_{ij} \leftarrow \sum_{a,b} g_a(x_i)\, g_b(x_j)\, G_{ab}(x_i - x_j) + \varepsilon\, \delta_{ij}$ ▷ $O(m_t^2 r^2)$ look-ups
  $\lambda \leftarrow \mathbf{G}^{-1} h$ ▷ Cholesky; LU if numerically indefinite
  $p_t \leftarrow \sum_a \mathcal{F}^{-1}\big[\overline{\mu_a}\, \mathcal{F}\big(\sum_i g_a(x_i) \lambda_i\, \delta_{x_i}\big)\big]$ ▷ $J^\mathsf{T}\lambda$, one FFT per operator
end for
return $w - p$

with near-vanishing rows, which equilibration does not fully remove. The Nyström preconditioner follows Frangella et al. (2023): an orthonormal Gaussian sketch $\Omega \in \mathbb{R}^{m \times \rho}$, $Y = JJ^\mathsf{T}\Omega$ from ρ matrix-free products, a stabilising shift γ, and the eigendecomposition $\widehat{U}\widehat{\Lambda}\widehat{U}^\mathsf{T}$ of the resulting rank-ρ approximation; the preconditioner applies $M^{-1} = (\hat{\lambda}_\rho + \varepsilon)\, \widehat{U}(\widehat{\Lambda} + \varepsilon I)^{-1}\widehat{U}^\mathsf{T} + (I - \widehat{U}\widehat{U}^\mathsf{T})$. Conjugate gradients needs on the order of $\sqrt{\kappa}$ iterations without a good preconditioner, which for $\kappa \sim 10^6$–10^7 exceeds the 300-iteration cap we used.

D Experimental setup

D.1 Guidance baselines and tuning

Table 5: Conditioning methods. R1: the update differentiates the network. R2: the final sample is a projection (exact for linear constraints, one Gauss–Newton step for nonlinear ones). R3: network passes per sample for K Euler steps and I optimisation iterations, and projections.

method	R1 steering	R2 enforcement	R3 cost per sample	tuned
DPS	✓	—	$2K$ passes	ζ
DiffusionPDE	—	—	K passes	ζ
D-Flow	✓	—	about $2IK$ passes	η, I
ECI	—	✓	K passes, K projections	—
PCFM	—	✓	K passes, K projections	—
GOSP (ours)	✓	✓	$2K$ passes, K projections	ζ

Every method starts from the same prior draws (seed 42, batch 2) and holds the initial condition by restriction: the fixed slice is set at flow time τ to $(1-\tau)u_0 + \tau u_{\mathrm{IC}}$, the interpolant the projection produces there. *DPS* steps $u \leftarrow u + \Delta\tau\, v_\theta - \zeta \nabla_u \|h(\xi)\|$ with the gradient through the network; *DiffusionPDE* holds v_θ fixed in the gradient; *D-Flow* runs 20 L-BFGS iterations on the prior draw

Table 6: Constraint error and cost of the runs of Table 1. Close to ref.: samples with trajectory error below 0.1, PCFM/DPS/GOSP. Resid.: residual of the observation or invariant rows, mean over samples in range; streamline crossings by the independent root search of §4; DPS + proj.: the DPS samples after a final kernel-Gram projection. Time: seconds for 100 samples.

		residual				time (s)		
set	close to ref.	PCFM	DPS	DPS + proj.	GOSP	PCFM	DPS	GOSP
vorticity integrals	—	1.1e−7	0.022	8.9e−5	1.1e−7	33	42	32
shallow water	100/100/100	1.9e−4	3.97	1.1e−3	1.9e−4	18	22	31
velocity, $P = 16$	100/100/100	1.5e−6	0.11	3.2e−6	1.3e−6	27	49	56
kinetic energy	0/92/93	1.2e−6	0.4	0.026	2.0e−6	24	48	53
streamline crossings	54/62/95	0.16	112	—	3.7e−5	89	58	128

Table 7: MSE of the ensemble mean over the samples that every method of Table 1 keeps in range. Common: their number, of 100.

					DPS		GOSP	
set	common	PCFM	ECI	D-Flow	tuned	no ref.	tuned	no ref.
vorticity integrals	100	0.17	0.21	0.14	0.19	0.2	0.17	0.15
shallow water	98	2.0e−3	4.6e−4	3.4e−3	6.8e−3	7.8e−3	2.0e−3	2.2e−3
velocity, $P = 16$	88	1.1e−3	1.0e−3	4.8e−3	9.9e−5	0.025	4.4e−5	6.7e−5
kinetic energy	88	0.45	0.6	0.046	2.7e−3	2.7e−3	2.6e−3	2.6e−3
streamline crossings	19	1.2e−4	2.5e−4	0.11	1.5e−3	0.014	4.9e−5	4.9e−5

with the gradient of $\|h(u_1)\|^2$ through all Euler steps, checkpointed per step; *ECI* is PCFM's projection with a fresh prior draw at every interpolation. *Guidance + projection, velocity form* adds the projection's change of the look-ahead to the DPS velocity, $u \leftarrow u - \zeta g + \Delta\tau\, v_\theta + \frac{\Delta\tau}{1-\tau}(\hat{\xi} - \xi)$, which with $\zeta = 0$ is not PCFM; before a late projection start it has no projection term and is DPS. The step ζ is chosen from $\{0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30\}$ for DPS and DiffusionPDE and from $\{0, 0.1, 0.3, 1, 3, 10, 30\}$ for GOSP and the velocity form, and D-Flow's learning rate from $\{0.01, 0.1, 1\}$, on 10 samples with prior seed 1 and the same reference, by the median trajectory error over all samples with samples out of range counted as failures (MSE for the vorticity integrals). Every run, with the MSE of the ensemble mean and of the ensemble standard deviation, trajectory errors and times, is listed in the supplementary material.

D.2 Data and model at 128^2

The 128^2 data follow PCFM's Navier–Stokes generator (100×100 training trajectories, 10×100 held-out ones with another seed), and the 128^2 model is the 64^2 Fourier neural operator fine-tuned on them for 40,000 steps. Fine-tuning lowered the validation loss but made unconstrained sampling less stable (70 of 100 samples in range, against 100 for the 64^2 weights used zero-shot). With 2048 or more sensors, 100 of 100 samples stay in range with either model. With the 64^2 weights the kernel Gram takes 499 ms and 1.04 s per sample, brings 100 samples close to the reference and meets the sensors to 3×10^{-5}. Table 4 uses the fine-tuned model so that model and data have the same resolution.

E Additional GOSP results

E.1 Sparse, noisy sensors: every method

Reference k uses initial vorticity $k \bmod 10$ of the held-out file, forcing index $(17 + 41k) \bmod 100$, sensor layout k and noise seed $1000 + k$; tuning uses $k = 100, \ldots, 103$. For an ensemble $x_1, \ldots, x_M$

and truth y at one grid value, the fair CRPS (Ferro, 2014) is

$$\mathrm{CRPS} = \frac{1}{M}\sum_{i=1}^{M}|x_i - y| - \frac{1}{2M(M-1)}\sum_{i\neq j}|x_i - x_j|,$$

the finite-ensemble adjustment of $\int \big(F(z) - \mathbf{1}\{y \le z\}\big)^2 dz$ for the ensemble's empirical distribution F (Hersbach, 2000); we average it over grid values. The CRPS skill score is $1 - \mathrm{CRPS}/\mathrm{CRPS}_{\text{none}}$, relative to the unconstrained sampler: 0 matches it, and 1 is a perfect forecast. The spread–skill ratio is $\big(\frac{M+1}{M}\,\overline{s^2}\big)^{1/2} / \big(\overline{(\bar{x}-y)^2}\big)^{1/2}$, with s^2 the unbiased ensemble variance, $\bar{x}$ the ensemble mean, bars averages over grid values, and $(M+1)/M$ the finite-ensemble factor under which a consistent ensemble has ratio one (Fortin et al., 2014); the *skill* in its name is the RMSE of the ensemble mean, not the CRPS skill score. Outside is the share of grid values where the truth lies outside the ensemble (0.06 if the truth is exchangeable with the 32 members); misfit is the RMS sensor residual in units of σ.

Table 8: Sparse, noisy velocity sensors with a free initial state, every method: 20 references, 32 samples each, medians over references. Err.: relative squared error of the ensemble mean. CRPSS: CRPS skill score. SSR: spread–skill ratio. Time: seconds per reference.

method	in range	err.	CRPS	CRPSS	SSR	outside	misfit	time
$P = 16$, $\sigma = 0.1$								
none	640/640	0.47	0.325	0.00	0.86	0.12	9.88	6
PCFM	633/640	6.2e−3	0.032	0.90	1.56	0.04	0.00	6
PCFM, $\varepsilon = \sigma^2$	640/640	0.014	0.051	0.84	1.90	0.01	0.34	6
final projection	640/640	0.46	0.314	−0.03	1.15	0.03	0.00	6
ECI, $\varepsilon = \sigma^2$	640/640	7.1e−3	0.034	0.89	0.96	0.14	0.30	6
D-Flow, lr 0.01	592/640	0.15	0.165	0.46	1.36	0.03	6.19	350
DPS, $\zeta = 10$	640/640	2.8e−3	0.021	0.93	0.69	0.29	0.91	14
DPS, likelihood step, $\zeta = 0.001$	640/640	0.051	0.091	0.71	1.45	0.04	3.46	14
GOSP, $\varepsilon = 10^{-6}$, $\zeta = 1$	637/640	5.4e−3	0.029	0.90	0.91	0.33	0.00	15
GOSP, $\zeta = 1$	640/640	4.5e−3	0.026	0.91	1.12	0.17	0.28	15
GOSP, likelihood step, $\zeta = 0.001$	640/640	5.1e−3	0.030	0.90	1.18	0.09	0.28	15
$P = 16$, $\sigma = 0.3$								
none	640/640	0.47	0.325	0.00	0.86	0.12	3.44	6
PCFM	173/640	0.26	0.073	0.74	2.82	nan	0.00	6
PCFM, $\varepsilon = \sigma^2$	640/640	0.2	0.199	0.35	1.47	0.01	1.73	6
final projection	640/640	0.47	0.318	−0.04	1.14	0.03	0.00	6
ECI, $\varepsilon = \sigma^2$	640/640	0.1	0.140	0.54	1.29	0.06	1.27	6
D-Flow, lr 0.01	608/640	0.13	0.145	0.52	1.42	0.03	2.24	350
DPS, $\zeta = 10$	640/640	7.5e−3	0.045	0.86	0.62	0.35	0.95	14
DPS, likelihood step, $\zeta = 0.01$	640/640	0.041	0.082	0.74	1.48	0.04	1.43	14
GOSP, $\varepsilon = 10^{-6}$, $\zeta = 0.3$	185/640	0.075	0.071	0.76	1.80	0.25	0.00	15
GOSP, $\zeta = 1$	640/640	1.0e−2	0.044	0.85	1.20	0.05	0.79	15
GOSP, likelihood step, $\zeta = 0.01$	640/640	0.014	0.050	0.82	1.16	0.06	0.79	15
$P = 4$, $\sigma = 0.1$								
none	640/640	0.47	0.325	0.00	0.86	0.12	10.48	6
PCFM	445/640	0.041	0.094	0.71	1.94	nan	0.00	6
PCFM, $\varepsilon = \sigma^2$	565/640	0.17	0.187	0.49	1.56	0.00	1.06	6
final projection	640/640	0.48	0.322	−0.05	1.15	0.03	0.00	6
ECI, $\varepsilon = \sigma^2$	629/640	0.089	0.134	0.61	1.34	0.05	0.68	6
D-Flow, lr 0.1	608/640	0.02	0.056	0.84	1.60	0.03	2.01	349
DPS, $\zeta = 10$	640/640	6.7e−3	0.038	0.88	0.48	0.49	0.69	14
DPS, likelihood step, $\zeta = 0.01$	640/640	0.019	0.062	0.82	1.09	0.06	1.84	14
GOSP, $\varepsilon = 10^{-6}$, $\zeta = 3$	640/640	7.5e−3	0.038	0.88	0.91	0.30	0.00	14
GOSP, $\zeta = 3$	640/640	7.4e−3	0.035	0.88	0.97	0.23	0.18	14
GOSP, likelihood step, $\zeta = 0.01$	640/640	0.011	0.045	0.86	1.03	0.12	0.22	15
$P = 4$, $\sigma = 0.3$								
none	640/640	0.47	0.325	0.00	0.86	0.12	3.68	6
PCFM	314/640	0.06	0.095	0.71	1.81	nan	0.00	6
PCFM, $\varepsilon = \sigma^2$	640/640	0.39	0.282	0.08	1.24	0.02	2.54	6
final projection	640/640	0.48	0.322	−0.06	1.15	0.03	0.00	6
ECI, $\varepsilon = \sigma^2$	640/640	0.29	0.236	0.23	1.07	0.08	2.19	6
D-Flow, lr 0.1	600/640	0.025	0.065	0.79	1.27	0.05	1.15	350
DPS, $\zeta = 10$	640/640	0.02	0.076	0.75	0.34	0.54	0.81	14
DPS, likelihood step, $\zeta = 0.1$	640/640	0.025	0.070	0.77	0.84	0.15	1.06	14
GOSP, $\varepsilon = 10^{-6}$, $\zeta = 3$	594/640	0.033	0.095	0.69	0.52	0.51	0.00	14
GOSP, $\zeta = 3$	640/640	0.029	0.078	0.74	0.71	0.29	0.68	14
GOSP, likelihood step, $\zeta = 0.1$	640/640	0.03	0.073	0.76	0.75	0.28	0.69	15

E.2 Ablations, diagnostics and robustness

Table 9: Every ablation on the main reference, 100 samples. Close: trajectory error below 0.1. Resid.: residual of the observation rows over samples in range. CRPS and spread–skill ratio (SSR) against the reference, as in Appendix E.1. $\hat{\epsilon}$ energy: largest median energy per entry of the implied prior draw over all steps. Time: seconds for 100 samples. Every step is tuned for its own variant; on kinetic energy the GOSP likelihood step was re-tuned with the rule for unprojected steps and again selects $\zeta = 1$.

sampler	in range	close	MSE	resid.	CRPS	SSR	$\hat{\epsilon}$ energy	time
velocity, $P = 16$								
PCFM	100	100	1.1e−3	1.5e−6	0.016	1.64	1.53	27
DPS, $\zeta = 10$	100	100	9.9e−5	0.11	3.8e−3	0.81	1.01	49
DPS, likelihood step, $\zeta = 1$	73	66	3.3e−3	1.39	0.019	2.93	1.03	52
GOSP, $\zeta = 3$	100	100	4.4e−5	1.3e−6	3.0e−3	0.76	1.02	56
GOSP, no charge to u_0, $\zeta = 3$	100	100	8.7e−4	2.1e−6	0.014	1.65	1.36	57
GOSP, $v_\theta(\tilde{u})$ recomputed, $\zeta = 3$	100	100	4.3e−5	2.1e−6	3.0e−3	0.75	1.02	76
GOSP, likelihood step, $\zeta = 0.1$	100	100	1.2e−4	2.1e−6	5.9e−3	1.23	1.03	58
Euler-form combination, $\zeta = 10$	100	100	9.8e−5	1.3e−6	3.8e−3	0.81	1.02	56
kinetic energy								
PCFM	100	0	4.3e−1	1.2e−6	0.35	1.07	1.9e3	24
DPS, $\zeta = 10$	100	92	2.7e−3	0.4	0.027	2.85	1.07	48
DPS, likelihood step, $\zeta = 1$	100	61	5.0e−2	1.45	0.11	1	1.06	51
GOSP, $\zeta = 10$	100	93	2.6e−3	2.0e−6	0.026	2.9	1.08	53
GOSP, no charge to u_0, $\zeta = 10$	100	92	2.6e−3	1.9e−6	0.026	2.87	1.06	53
GOSP, $v_\theta(\tilde{u})$ recomputed, $\zeta = 10$	94	83	4.0e−3	1.9e−6	0.028	3.57	1.05	72
GOSP, likelihood step, $\zeta = 1$	100	62	4.9e−2	2.0e−6	0.1	1.01	1.86	53
Euler-form combination, $\zeta = 10$	100	92	2.7e−3	1.2e−6	0.027	2.84	1.07	51
streamline crossings								
PCFM	54	54	2.0e−4	0.16	6.7e−3	1.85	1.11	89
DPS, $\zeta = 1$	96	62	2.2e−2	112	0.032	2.44	1.03	58
DPS, likelihood step, $\zeta = 0.01$	100	28	1.5e−1	—	0.15	1.22	1.1	60
GOSP, $\zeta = 0.3$	95	95	4.7e−5	3.7e−5	3.2e−3	0.79	1.02	128
GOSP, no charge to u_0, $\zeta = 0.3$	55	55	1.1e−4	0.014	4.8e−3	1.74	1.03	129
GOSP, $v_\theta(\tilde{u})$ recomputed, $\zeta = 0.3$	90	90	4.8e−5	2.5e−4	3.2e−3	0.74	1.02	150
GOSP, likelihood step, $\zeta = 0.001$	89	89	8.0e−5	—	4.5e−3	0.86	1.02	130
Euler-form combination, $\zeta = 0.3$	100	94	1.2e−3	12.8	5.1e−3	4.2	1.01	128

Table 10: The prior draw implied by state and network, $\hat{\epsilon} = u_\tau - \tau v_\theta(u_\tau, \tau)$, taken at the state the network is called on, before the guidance step; the draw stored by (8) is the same expression after it, differing by $-\zeta g$. Over 100 samples, for PCFM, DPS, GOSP and GOSP without the charge: range over steps $\tau \leq 0.9$ of the median energy per entry (the prior's is 1), its largest value over all steps, and the largest ratio of the low-wavenumber energy share ($|k_x|, |k_y| \leq N/8$) to its white-noise value. The first time slice is excluded.

sampler	energy, $\tau \leq 0.9$	energy, max	low-k ratio, max
velocity, $P = 16$			
PCFM	0.999–1.019	1.53	4.84
DPS	0.985–1.009	1.01	1.34
GOSP	0.990–1.007	1.02	1.13
GOSP, no charge to u_0	0.997–1.017	1.36	3.98
kinetic energy			
PCFM	0.990–1.007	1.9e3	2.73
DPS	0.993–1.007	1.07	2.09
GOSP	0.992–1.007	1.08	2.35
GOSP, no charge to u_0	0.993–1.008	1.06	2.10
streamline crossings			
PCFM	1.000–1.037	1.11	2.56
DPS	0.985–1.003	1.03	1.67
GOSP	0.990–1.008	1.02	1.27
GOSP, no charge to u_0	0.998–1.024	1.03	1.58

Table 11: PCFM, DPS and GOSP over 15 settings (five held-out initial vorticities and phases, three prior seeds or sensor layouts each), 100 samples each, ζ fixed at its main-reference value: median [min, max] over settings. Resid.: constraint residual against each setting's own reference, mean over samples in range; for streamline crossings, from the independent root search. The largest GOSP streamline residual (0.4) comes from one of the 87 samples in range of one setting, in which one of 426 crossings has no root on its line because its contour no longer reaches it; that crossing counts as half a line (32 cells), and without that sample the setting's residual is 2×10^{-2}.

sampler	close	MSE	resid.	CRPS	spread–skill ratio
velocity, $P = 16$					
PCFM	100 [99, 100]	6.8e−4 [2.4e−4, 2.3e−3]	1e−6 [1e−6, 2e−6]	0.016 [0.012, 0.022]	2.40 [1.21, 3.31]
DPS	100 [92, 100]	1.1e−4 [6.5e−5, 1.1e−3]	1e−1 [9e−2, 5e−1]	0.004 [0.003, 0.007]	3.14 [0.76, 4.74]
GOSP	100 [100, 100]	4.5e−5 [2.0e−5, 7.9e−5]	1e−6 [1e−6, 1e−6]	0.003 [0.002, 0.004]	0.73 [0.56, 1.13]
kinetic energy					
PCFM	0 [0, 1]	3.4e−1 [2.3e−1, 6.1e−1]	5e−4 [4e−5, 6e−3]	0.313 [0.252, 0.434]	1.23 [0.89, 1.45]
DPS	80 [30, 94]	2.4e−2 [2.7e−3, 9.0e−2]	8e−1 [4e−1, 2]	0.069 [0.027, 0.146]	1.84 [1.04, 3.51]
GOSP	79 [29, 94]	2.6e−2 [2.6e−3, 8.7e−2]	4e−6 [2e−6, 6e−4]	0.069 [0.026, 0.147]	1.95 [1.09, 3.51]
streamline crossings					
PCFM	61 [45, 77]	1.5e−4 [5.1e−5, 4.1e−4]	2e−1 [5e−2, 2]	0.007 [0.005, 0.009]	1.89 [1.31, 3.47]
DPS	61 [31, 82]	2.2e−2 [1.1e−3, 1.8e−1]	1e2 [4e1, 2e2]	0.032 [0.007, 0.153]	2.43 [1.16, 9.16]
GOSP	90 [73, 95]	4.9e−5 [2.8e−5, 7.5e−5]	1e−4 [4e−5, 4e−1]	0.003 [0.003, 0.004]	0.77 [0.58, 1.10]

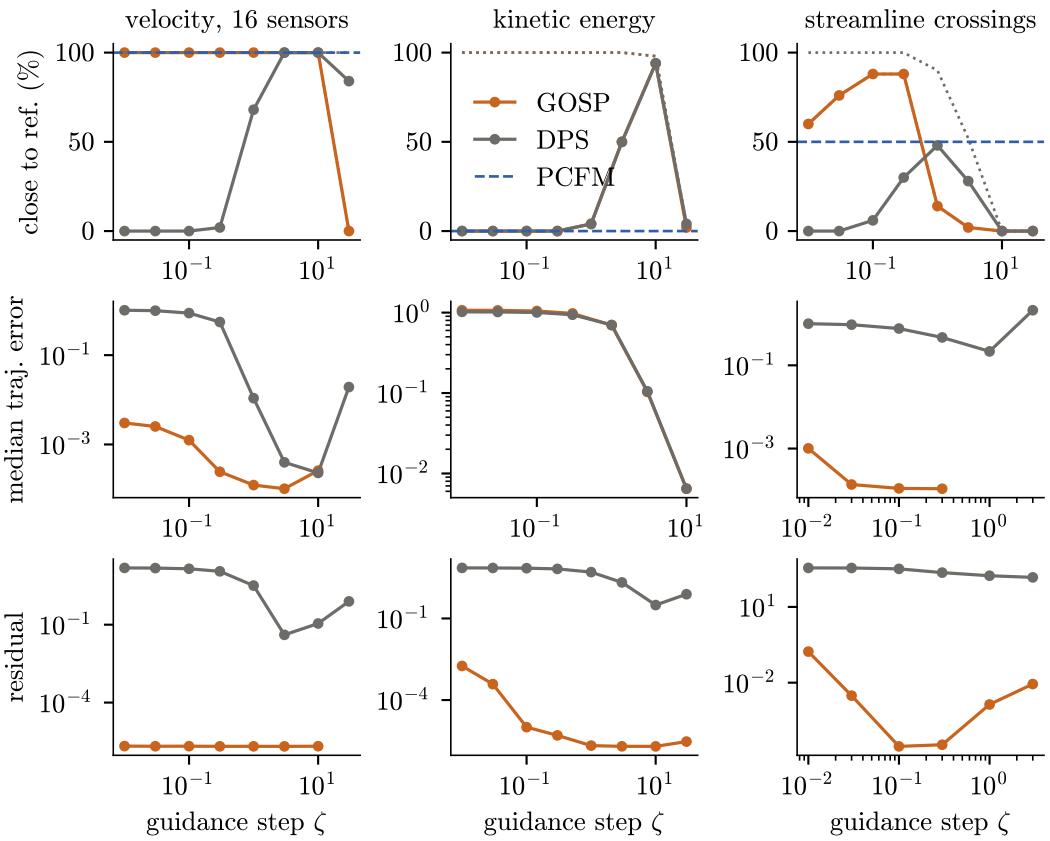


Figure 4: Sensitivity to the guidance step, 50 samples from a prior seed used neither for tuning nor for evaluation. Top: samples close to the reference (solid) and in range (dotted). Middle: median trajectory error over all samples, samples out of range counted as failures (missing where the median is infinite). Bottom: constraint residual over samples in range. PCFM: dashed, top. On kinetic energy the GOSP and DPS curves nearly coincide.

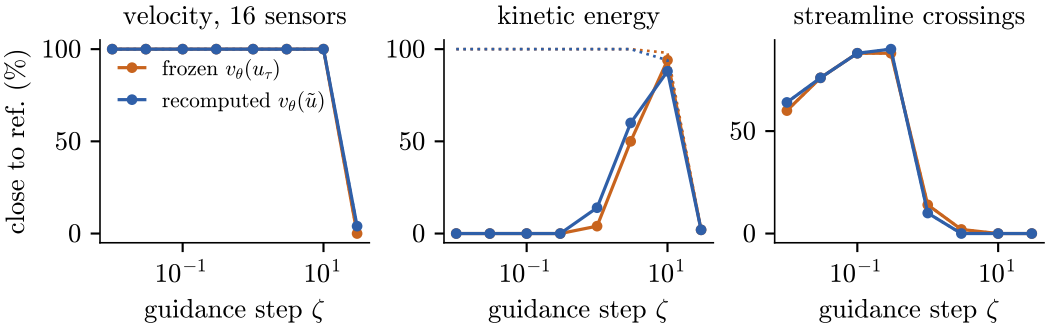


Figure 5: Frozen velocity $v_\theta(u_\tau)$ (GOSP) against the velocity recomputed at the moved state, over the guidance step, 50 samples from the prior seed of Figure 4: samples close to the reference (solid) and in range (dotted).

Table 12: The network's response to the guidance step when the velocity is recomputed: for each sample the largest $(1-\tau)\|v_\theta(\tilde{u},\tau) - v_\theta(u_\tau,\tau)\|/\|\tilde{\xi}\|$ over steps, median over the samples that end in range and over those that leave it; 50 samples, prior seed of Figure 4.

set	ζ	in range	response, in range	response, out of range
velocity, $P = 16$	1	$50/50$	0.49	—
velocity, $P = 16$	3	$50/50$	0.69	—
velocity, $P = 16$	10	$50/50$	2.57	—
streamline crossings	0.1	$44/50$	0.5	0.26
streamline crossings	0.3	$45/50$	0.75	1.81
streamline crossings	1	$5/50$	1	5.78
kinetic energy	1	$50/50$	0.15	—
kinetic energy	3	$50/50$	0.39	—
kinetic energy	10	$47/50$	0.74	59.1

Likelihood step. DPS's step $\zeta\nabla\|h\|^2/\|h\|$ is $2\zeta\nabla\|h\|$; the Gaussian log-likelihood gradient $\nabla\|h\|^2/2\sigma^2$ vanishes with the residual. With ζ tuned separately for each, the likelihood step is worse for DPS and for GOSP on every set (Table 9), and with noisy sensors it raises DPS's CRPS in three of four configurations (Table 8).

E.3 Steps before the projection starts

On kinetic energy the projection starts at $\tau_0 = 0.95$. Table 13 compares four treatments of the steps before τ_0, from the same prior draws and with $\zeta = 10$. Charging the stored prior draw on every step leaves the level the energy constrains accurate but raises the error on the unconstrained levels. Without a projection, that update is the DPS step plus $(1-\tau')(\tilde{u}_0 - \hat{\epsilon})$, and each charge widens the gap between the stored and the implied prior draw by $\zeta\tau g/(1-\tau)$. Holding the initial condition, as DPS does, repairs the early levels, and interpolating from the implied draw repairs the rest. GOSP in the main text uses both and reaches DPS's accuracy with an energy residual of order 10^{-6}.

Table 13: Kinetic energy, 100 samples, $\zeta = 10$: treatments of the steps before the projection starts. Level MSE: MSE of the ensemble mean on single time levels ($t = 49$ is the constrained one). Residual: energy residual over samples in range. $\hat{\epsilon}$ energy: largest median energy per entry of the implied prior draw over all steps, as in Table 10.

				level MSE				
sampler	in range	close	MSE	$t = 5$	$t = 25$	$t = 49$	residual	$\hat{\epsilon}$ energy
DPS	100	92	2.7e−3	9.9e−3	1.1e−3	2.5e−4	0.4	1.07
GOSP, stored draw charged on every step	99	92	0.014	0.025	0.023	9.3e−3	2.0e−6	1.35
GOSP, stored draw, initial condition held	99	92	0.012	2.4e−3	0.022	0.013	1.9e−6	—
GOSP, implied draw before τ_0, initial condition free	100	93	5.5e−3	0.027	9.4e−4	2.0e−4	2.0e−6	—
GOSP (implied draw before τ_0, initial condition held)	100	93	2.6e−3	9.9e−3	1.0e−3	1.7e−4	2.0e−6	1.08

F Projection without guidance

F.1 Projecting at every step or only at the end

Table 14: MSE when the constraints are applied at every sampling step, only to the final sample, to the final sample with 100 Gauss–Newton steps at the in-sampler damping (10 for velocity), or never; 100 samples, same prior draws; in brackets the samples in range where not all are. None: no constraints. Close: trajectory error below 0.1 (in-sampler/final only/final, matched). Traj. error, all: median trajectory error over all 100 samples, counting samples out of range as failures (in-sampler/final only). Kinetic energy: stride 8 at $t = 49$, corrections from $\tau_0 = 0.95$ with three Gauss–Newton steps. Streamline crossings: continuation schedule (§F.1), $\varepsilon = 10^{-1}$ for final, matched.

set	in-sampler	final only	final, matched	none	close	traj. error, all
vorticity integrals	1.7e−1	2.1e−1	2.1e−1	2.1e−1	—	—
shallow water	2.0e−3	diverged	diverged	diverged	100/0/0	2.7e−2/∞
velocity, $P = 16$	1.1e−3	4.2e−1	4.2e−1	4.4e−1	100/0/0	4.0e−3/9.9e−1
kinetic energy, $t = 49$, $\tau_0 = 0.95$	4.3e−1	4.4e−1	4.4e−1	4.4e−1	0/0/0	1.1/1.1
streamline crossings, continuation	2.0e−4 (54)	4.4e−1	4.3e−1	4.4e−1	54/0/0	3.8e−3/1.1

Table 14 compares four schedules from the same prior draws. Projecting at every step helps on the vorticity integrals (MSE 0.174 against 0.207 for a final projection, lower in 15 of 15 held-out

settings), on shallow water, whose model leaves the physical range without its conservation rows, and on velocity, where a final projection meets the sensors with MSE 0.39–0.43, close to 0.44 without constraints; the in-sampler MSE was lower in all 15 held-out settings at $P = 4$, 16 and 64, and with a free initial slice as well. A final projection in a spectral metric leaves the unobserved levels at their unconstrained error (1.31), and 100 Gauss–Newton steps change no MSE in Table 14 by more than 0.002. Projecting at every step does not help on kinetic energy (§F.2). On streamline crossings 54 of 100 samples are close to the reference with the continuation schedule of §F.3, and 45–77 over 15 held-out settings, against at most 1 after a final projection.

F.2 Why kinetic energy fails

With PCFM's default schedule, kinetic-energy targets diverge with the kernel Gram and with dense assembly alike. Per-step traces (stride 8, $t = 49$) show that the first step, linearised at a look-ahead that is mostly prior noise, overshoots: the residual goes from 7.0 to 142 as the state moves by 124% of its norm. Later steps that a line search accepts still move the field by about 20%. In the traces the look-ahead, near a conditional mean, has less energy than the reference; each correction adds energy, the network's velocity grows, and the look-ahead norm grows by about 36 per step until it overflows.

Levenberg–Marquardt damping and a trust region did not help. From $\tau_0 \leq 0.9$, Levenberg–Marquardt damping kept 1–9 of 100 samples in range, against 1–5 with a line search. A trust region capping each step at 5% of the state norm kept most samples in range, with trajectory error 1.23–1.39, no better than without energy rows. The few samples that stay in range with a line search are much closer to the reference (trajectory error 0.21–0.26, against 1.24 without energy rows). From $\tau_0 = 0.95$ the energy error at the constrained level drops from 0.86 to 0.41 while the vorticity error there rises. We do not know a schedule that keeps early corrections stable for this target.

F.3 Streamline crossings: schedule and root events

With PCFM's $\varepsilon = 10^{-6}$ throughout, 35 of 100 samples stay in range; with $\varepsilon = 10^{-1}$ all stay in range but only 25 are close to the reference. The continuation holds $\varepsilon = 10^{-1}$ until $\tau = 0.5$ and decreases it log-linearly to 10^{-6} while the search window shrinks from 16 to 4 cells; it was fixed on the main reference and not tuned further.

Root events. During sampling, the continuation schedule halves the share of crossings that leave their search window compared with fixed damping $\varepsilon = 10^{-1}$, although its wider early window lets more roots change branch or order early in sampling; on the main reference no crossing of GOSP's final samples is outside its window. Per-correction root-event statistics are in the supplementary material.

G Projection solvers

Complete runs with the kernel Gram give the same samples as dense assembly up to float32 rounding ($\Delta_{\text{dense}} \leq 4 \times 10^{-4}$). On the vorticity integrals and shallow water a complete run takes 1963 s and 4080 s with the released implementation, 90 s and 108 s with the matrix-free projection, and 33 s and 18 s with the kernel Gram and the initial condition restricted (Table 15). The closed-form Gram and its correction agree with the Jacobian from automatic differentiation to rounding error in double precision on all five sets (Table 16).

Figure 6 sweeps one correction over problem size. On the observables the kernel Gram is 2–180 times faster than dense assembly and continues to $m = 81920$ in under a second, where dense assembly runs out of memory from $m = 40960$; matrix-free CG converged on the integral sets and on 5 of 12 observation configurations of the sweep (hollow markers in Figure 6 did not), whose damped blocks have condition numbers up to 3×10^7. The sweep also covers three further translation-invariant constraint sets (vorticity and enstrophy, kinetic energy at off-grid sensors, and gravity-wave energy of shallow water); their definitions, exactness and per-correction timings, and the resolution and damping sweeps, are in the supplementary material.

Table 15: Complete PCFM runs of 100 samples, batch 2, same checkpoint and prior draws: time in the projection and in total, in seconds. Kernel Gram, dense assembly and the last CG column restrict the initial condition; the released projection and the first CG column keep it as identity rows. †Dense Jacobian factored once, its assembly counted in the total. Δ_{dense}: per-sample relative distance between kernel-Gram and dense samples.

	kernel Gram		dense		released	CG, total		
set	proj.	total	proj.	total	total	IC rows	restr.	Δ_{dense}
vorticity integrals	12.7 s	32.8 s	40.8 s	61 s	1963 s	89.8 s	64 s	2.4e−4
shallow water	8.82 s	17.6 s	40.7 s	50.1 s	4080 s	108 s	125 s	3.8e−5
velocity, $P = 1024$†	27.4 s	47.8 s	40 s	115 s	—	—	—	2.3e−4
kinetic energy, $s = 8, t = 49$	3.42 s	23.6 s	7.66 s	27.9 s	—	—	—	2.0e−4
streamline crossings, 16×8	67.6 s	113 s	116 s	136 s	—	—	—	3.9e−4

Table 16: Exactness in double precision on a perturbed state, against the Jacobian assembled by automatic differentiation, largest over 16^2 and 64^2 grids: relative error of the closed-form Gram and of the correction, and of directional derivatives against central differences. On every set the Gram has no entry between different time levels.

set	Gram	correction	fin. diff.
vorticity integrals	0	3e−15	5e−9
shallow water	6e−16	2e−12	9e−9
velocity	4e−16	2e−15	4e−8
kinetic energy	2e−15	7e−15	3e−8
streamline crossings	5e−16	4e−13	8e−8

The direct solve needs no tolerance or preconditioner. Most undamped Grams are numerically singular, so ε is still needed; a damping sweep in the supplementary material shows that it trades precision against how much of the residual one step removes.

Past the memory limit of the per-level block we apply the Gram without forming it. $J^{\mathsf{T}} z$ is a scatter and an FFT per operator and Jy is an FFT and a gather, so one CG iteration costs about 8 times less than through automatic differentiation. With $P = 65536$ sensors, where the Cholesky block does not fit, CG converges in 0.58 s for velocity ($m = 655360$, Jacobi) and in 8.5 s for $|u|^2$ (Nyström, rank 2048), using at most 24 MB. These runs are in the supplementary material.

H Relation to SNAP-FM (concurrent work)

SNAP-FM (Kolli et al., 2026) targets the same cost, the linear algebra inside PCFM's projection. We did not run its sparse projection and compare the methods by structure only (Table 17).

Table 18 measures the assembled Jacobian of each set at its benchmark size, in float64. The integral sets are sparse, because the initial condition contributes one identity row per cell and only a few dozen rows are dense, so they are where the two approaches overlap. On the point sets every constraint row is nonzero on essentially all of its time slice, and most of that fill survives a threshold at 10^{-3} of the row's largest entry; within a level the Gram block is dense. An exact sparse representation therefore stores as many numbers per level as a dense one, and eliminating the state from the KKT system forms the dense per-level Schur complement from products of those rows, the block the kernel Gram writes from kernel look-ups.

Table 17: Sparse projection (SNAP-FM) and the kernel Gram.

	SNAP-FM	kernel Gram (ours)
form of J	sparse	dense
structure used	local coupling	convolution kernel and point sampling
what is formed	J and a sparse KKT system	only JJ^T, from kernel lookups
typical constraints	initial and boundary conditions, local conservation, Godunov updates	quantities derived through Δ^{-1} or a Helmholtz projection: velocity, kinetic energy, streamline crossings, off-grid sensors, gravity-wave energy
sampling points move with the state	outside the scope	yes, through (14)
limit	fill-in on large three-dimensional problems	the dense $m_t \times m_t$ block per level, $O(m_t^3)$
outer algorithm	full nonlinear program, interior point	one damped Gauss–Newton step, as in PCFM

Table 18: Sparsity of the assembled Jacobians. Row fill: smallest fraction of its own time slice (on the channels it touches) at which a constraint row is nonzero, at 10^{-10} and 10^{-3} of the row's largest entry. Density of J over all $m \times n$ entries, including initial-condition rows. Gram block: density of JJ^T within a level, constraint rows only; not shown where a level holds at most four rows. The last three sets are those of Figure 6.

			row fill in slice			
constraint set	m	m_t	10^{-10}	10^{-3}	density of J	Gram block
vorticity integrals $64^2 \times 50$	4145	1	100%	100%	0.048%	—
shallow water $48^2 \times 25$	7008	4	100%	95%	0.028%	—
kinetic energy, $s = 4$	1329	256	99.9%	90%	2%	100%
streamline crossings, 64×32	2663	557	100%	99%	2%	100%
vorticity and enstrophy $64^2 \times 50$	4194	2	100%	99.7%	0.071%	—
off-grid sensors, $P = 1024$	5120	1024	100%	91%	2%	100%
wave energy, $s = 4$	720	144	99.9%	26%	4%	99.9%

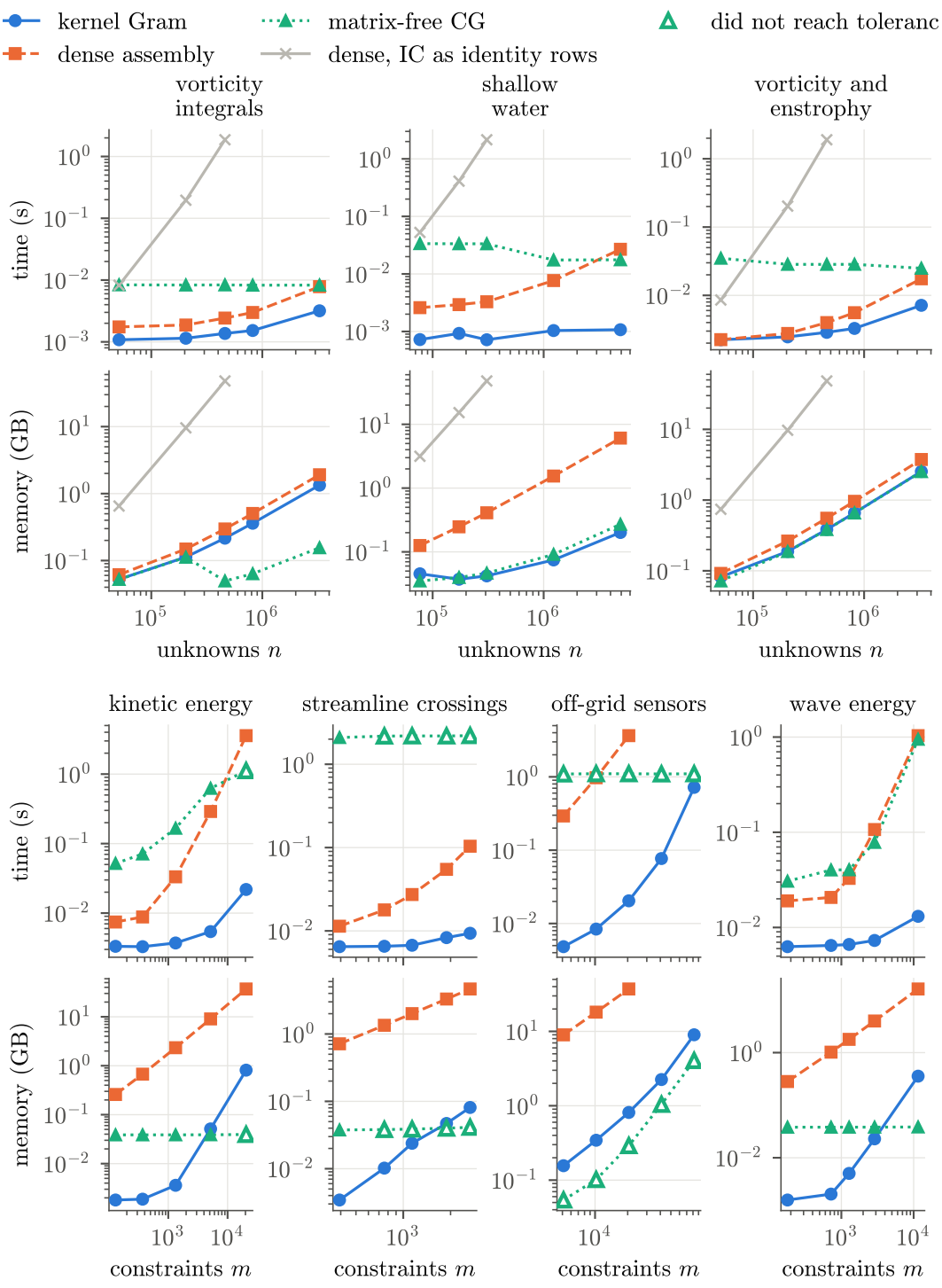


Figure 6: One correction against problem size, batch 1, for every set except velocity. Top block: the integral sets against the number of unknowns n, swept over grid size (32^2 to 256^2; kernel Gram is the closed-form integral block); grey: dense assembly with the initial condition as identity rows. Bottom block: the point sets against the number of constraints m, swept over sampling density (for streamline crossings, over lines $\times$ contours, from 16×8 to 64×32). In each block time is above and peak memory below. Dense assembly stops where the Jacobian no longer fits on the device. A hollow marker is a matrix-free solve that did not reach the tolerance.