Injecting Sensitivity Constraint Into Continual Learning Significantly Enhances Surrogate-Aided Optimization

Anonymous authors
Paper under double-blind review

Abstract

A myriad of scientific and engineering optimization and learning tasks involve running a numerical model to guide optimization directly or generate training data for function mapping algorithms. Surrogate models can greatly accelerate these tasks, but they often fail to capture the true input-output relationships (sensitivities) so they lose the ability to guide high-dimensional and long-horizon optimization. Online continual learning (OCL) – iteratively obtaining numerical results to continue training the surrogate – can mitigate this issue, but may still be insufficient. Here we propose scheduled injection of sensitivity constraints (SC, matching the Jacobian of the surrogate model with that of the true numerical model) for the surrogate into OCL to enforce realistic output-parameter relationships. We evaluate this approach across diverse datasets and optimization frameworks where continual surrogate training is used: (1) multi-objective multi-fidelity surrogate-assisted Bayesian optimization and Pareto front exploration; (2) hybrid end-to-end training of coupled neural networks and process-based models; and (3) a modified unifying framework for generative parameter inversion and surrogate training. Across all of these tasks, inserting SC accelerates the descent to optimality and consistently improves the main optimization outcome, as it critically improves the future trajectory of optimization. OCL improves data relevance and SC ensures sensitivity fidelity, and they together produces an efficient surrogate model that almost achieves the same effect as the full physical model, only achievable by OCL+SC. It consistently outperform pretrained-only surrogate models with SC or OCL without SC, not to mention the pretrained-only model without SC, so the benefits of two procedures reinforce each other. Even infrequent surrogate finetuning with SC injection (once every 5 epochs) can induce large benefits in optimization outcome. Together, these results demonstrate the possibility to enable large-scale optimization of complex systems for big-data learning and knowledge discovery.

1 Introduction

Optimization with physical models: Across various domains including robotic controls, physics, biomedicine, geosciences, etc., many crucial optimization tasks still involve physical models. The first is parameter estimation. Physical models solve governing equations and can ensure some expected behaviors, but they are often modulated by parameters that determine how the states evolve (Tarantola, 2004). Some of these parameters are physical parameters, i.e., thermal conductivity in heat transfer, or diffusion coefficients in porous media (Lu et al., 2021); Some of them are conceptual parameters to resolve the impacts of scale, i.e. parameterization schemes in climate modeling (Arakawa, 2004; Hourdin et al., 2017); Some others are empirical parameters to compensate for insufficient process representations (Kattge & Knorr, 2007; Medlyn et al., 2011). In many applications, the true values of these parameters or the parameterization function (the procedure to generate these parameters based on some known inputs) are not known in advance and must be inferred through iterative calibration or inverse modeling that ensuring a good match between simulations and observed system behaviors (Gupta et al., 1998). In more recent designs, we can train a neural network to capture the inverse mapping from solution to the parameters (Ardizzone et al., 2019) from large amounts of accumulated simulations. For both, we need a large number of simulations.

The second optimization task gaining popularity is knowledge discovery, where a part of the physical model formulation is predefined while some parts of the relationships are to be learned using coupled neural networks (Karniadakis et al., 2021; Yang et al., 2017; de Avila Belbute-Peres et al., 2018). The physical components help narrow down what the neural networks (NNs) can learn, so we gain some interpretability of the learned relationship. Knowledge discovery often carries out end-to-end joint training of the hybrid system on big data. A third case is design optimization, where we select design variables to maximize certain benefits or explore the Pareto front for multiple objectives (Leifsson & Koziel, 2010; Wang & Shan, 2008; Praslicka et al., 2023). A fourth case is model predictive control (MPC) (Amos et al., 2019; Schwenzer et al., 2021), where a physical model represents the physical environment and its interaction with the body to be controlled. The algorithm then searches for optimal controls that achieve certain trajectories.

In all of the aforementioned optimization tasks, a process-based model is needed to represent the physical system behavior with some rigidity and it needs to be run a large number of times. The physical model may be solving partial differential equations (PDEs) or other types of equations, sometimes over long time periods or at high spatial resolution, which can be computationally expensive. The optimization can then be prohibitively expensive, especially when the parameter dimension is high.

Surrogate-assisted optimization: Surrogate models are designed to mimic the behaviors of the full model at a fraction of the cost, deliver significant computational savings and enable large-scale or real-time analyses. Gaussian process regression, neural networks and other machine learning methods have long been used as surrogate models (Marrel & Iooss, 2024). More recently, neural operators have gained attention due to their ability to learn mappings between infinite-dimension function spaces and propagate initial or boundary conditions through space and time, allowing them to predict solutions at arbitrary locations and timesteps rather than at fixed training points (Li et al., 2021; Wang et al., 2021; Azizzadenesheli et al., 2024). Yet, in the face of high-dimensional inputs, surrogate models are challenging to train and frequently lose fidelity when used outside their training knowledge domain or require an impractically large number of parameter samples to mimic the underlying system.

Online continual learning: The default strategy for surrogate models is to pretrain them with a diverse and comprehensive set of simulations so they accurately cover the entire input space during the optimization. However, this can be challenging with high-dimensional, long-horizon optimizations where it is difficult to devise the training data. As a result, the surrogate model may lose the ability to guide as the optimization proceeds. In contrast, online training strategies dynamically generate new training data during optimization and iteratively retrain the surrogate using fresh samples computed by the physical models (Meyer et al., 2023). This has already been a widely followed path in traditional surrogate-assisted optimization (Wistuba et al., 2018). Training solely the recent examples, however, risk catastrophic forgetting. This need for continual learning is not without some resemblance to training AI models under continuously-arriving new data (Aljundi et al., 2019; Nagabandi et al., 2019), or evolving agents for changing environments (Wołczyk et al., 2021; Zhou et al., 2024; Liu et al., 2025; Wu et al., 2021), where one needs to balance adaptation to new environments as well as mitigate catastrophic forgetting.

A more recent representative example in the machine learning era is the Self-directed Online Learning Optimization (SOLO) framework, as suggested by Deng et al. (2022), which integrates deep neural network (DNNs) as surrogates with finite element simulations to iteratively approximate objective functions across problems as Truss optimization and heat transfer enhancement. Similarly, continual learning approaches aim to mitigate forgetting by continuously updating the training buffer with representative samples. The buffer can be refreshed by replacing older samples chronologically, randomly, or through gradient-based selection strategies that prioritize data most informative for maintaining model accuracy (Aljundi et al., 2019). Here we will demonstrate that even online training may not adequately address the fidelity issue – it can still lead to suboptimal optimization outcomes which could be substantially enhanced by the incorporation of sensitivity information, when available, in surrogate training.

2 Related Work: Optimization frameworks

We sample several successful optimization frameworks that utilize surrogate models and could benefit from our proposed idea of injecting sensitivity constraint.

Multi-objective multi-fidelity Bayesian optimization (MOMF-BO): (Takeno et al., 2020) is a class of methods that seeks to efficiently explore the Pareto front of multiple competing objectives under strict computational budgets. It combines evaluations of the true numerical model at different fidelity levels, e.g., using cheaper, coarser-resolution approximations alongside high-fidelity simulations — to balance accuracy with cost. The framework begins with a small initial set of evaluations to train a probabilistic surrogate model (typically a Gaussian process) that not only predicts objective values but also quantifies uncertainty. Based on these surrogate predictions at each step, an "acquisition function" selects promising candidates by balancing exploring uncertain regions and exploiting areas likely containing good solutions, dictated by a fidelity parameter (Irshad et al., 2024). These selected candidates are then evaluated using the actual numerical model at chosen fidelity levels, and the results are added to the buffer of ground-truth data. The surrogate is retrained on this expanding buffer, and the cycle repeats until the evaluation budget is exhausted to depict the Pareto fronts.

Hybrid training of neural networks and differentiable numerical models: A large class of methods train neural networks (NNs) together with physically-based numerical models that support differentiable programming (Innes et al., 2019). The physically-based component provides a behavioral backbone while the NNs learn either parameterization functions (Feng et al., 2022), or certain missing processes. To be trained with NNs jointly, the physics model is either re-implemented to enable automatic differentiation (AD) or solve the adjoint equations for gradients (Rackauckas et al., 2021; Gelbrecht et al., 2023), or a surrogate NN can be used in its place. An NN as the surrogate model can naturally support AD and joint training. In the end, the algorithm trains NNs to discover knowledge, provide physical parameters, and improve the predictive performance of the combined system.

Joint training of generative parameter inversion and surrogate model: Lingsch et al. (2024) proposed the Fast Unified Simulation and Estimation (FUSE) framework, designed to jointly address prediction tasks involving both discrete parameters and continuous fields governed by parametric PDEs. FUSE integrates a forward model and an inverse model into a single end-to-end differentiable setup. The forward model (implemented using neural operators) maps parameters and other inputs to the solution space over time. The inverse model, implemented using a Flow Matching Posterior Estimation (FMPE), generates the possible parameters satisfying the posterior distribution conditional on indirect or partial observations. The two models are trained in a combined system in an alternating order, with full model runs at the beginning to provide training data for the scheme. In the end, the training produces both a useful surrogate model and a parameter-generating model.

Physics-informed or gradient-informed surrogate training: To address the limitations of surrogate models, recent work such as physics-informed neural operators (PINO) (Li et al., 2024) has explored incorporating physical constraints into surrogate training. Towards incorporating physical constraint, Behroozi et al. (2024) proposed sensitivity-constrained Fourier Neural Operator (SC-FNO), incorporating a sensitivity-based term using the exact Jacobian of model output with respect to input parameters to regularize the gradient of the surrogate model. Their model outperformed standard and physical-equation-regularized FNOs for several parametric PDEs. Similarly encouraging results were proposed in Derivative-enhanced DeepONet (DE-DeepONet) (Qiu et al., 2024) and Derivative-informed Neural Operators (DINO) (O'Leary-Roseberry et al., 2024), which apply low-rank approximations to obtain approximate Jacobians, and Sobolev training (Czarnecki et al., 2017), which adds derivative terms to improve function approximation in Sobolev norms. Despite these advances, all such approaches have been demonstrated in offline or pretrained settings, typically with relatively low-dimensional inputs. It remains unclear whether these gradient-informed strategies remain effective in online continual learning, where surrogates are repeatedly updated and their evolving accuracy affects future optimization trajectories.

Our contribution: For each of the aforementioned optimization frameworks, we propose to incorporate a sensitivity constraint (SC) loss term, which matches the sensitivity of the surrogate model with that of the numerical model, while enabling online continual learning (OCL) of the surrogate model. We finetune the surrogate only upon scheduled intervals, e.g., once every epoch. We will demonstrate significant improvements of the main optimization from OCL+SC compared to OCL alone. Our contribution does not lie in inventing sensitivity-matching losses per se, but in the novel coupling of OCL and SC and show how their combination, applied at various frequencies, influence long-horizon optimization. To our knowledge, no previous work showed how OCL interacts with SC. Unlike prior work that requires large-scale pretraining of gradient-informed surrogates, we show that iteratively finetuning yields superior efficiency, because the sensitivity information is used exactly where it matters—along the optimization trajectory and in regions critical to the search.

3 Experiments

We evaluate our approach across three frameworks and four experiments/datasets. Each framework is sequential in nature; results in early iterations, as influenced by the surrogate models employed, determine future candidate designs or search directions. Therefore, we must assess not only the quality of the accuracy of the surrogate models but also the quality of the optimization outcome and the speed of convergence. By testing across a diverse selection of frameworks and datasets, ranging from low to high complexity, we motivate the general applicability of the continual training with sensitivity constraints (SC).

3.1 Multi-objective multi-fidelity Bayesian optimization (MOMF-BO)

The Branin-Currin problem serves as a simple, analytical benchmark representative of typical MOMF-BO tasks. In particular, this benchmark pairs the Branin and Currin functions as two objectives (see Appendix A.1) with a known Pareto front (Irshad et al., 2024). In the HV-KG framework introduced by Daulton et al. (2023), differentiable Gaussian Process (GP) surrogate models for objective functions and an acquisition function leveraging knowledge gradients to achieve high efficiency and accuracy relative to other standard optimization search algorithms. Daulton et al. (2023) trained GP-based probabilistic surrogates by maximizing the log marginal likelihood (MLL) — a statistical objective that balances data fit and generalization. We augment this setup by supplementing MLL with a sensitivity-constrained loss term generally defined by

$$L_s = \frac{1}{M}\sum_{j=1}^{M}\left\|\frac{\partial \hat{\mathbf{u}}(\mathbf{x}_j, t_j; \mathbf{p})}{\partial \mathbf{p}} - \frac{\partial \mathbf{u}(\mathbf{x}_j, t_j; \mathbf{p})}{\partial \mathbf{p}}\right\|^2, \tag{1}$$

where $\mathbf{x}_j$ and t_j represent spatiotemporal or otherwise design coordinates of points where Jacobians are evaluated, and $\mathbf{p}$ represents parameters that modulate the modeled system $\mathbf{u}$. This loss encourages the surrogate to learn accurate responses to dynamic parameter landscapes. For the Branin-Currin objective functions, this constraint is imposed on the fidelity parameter s intended to simulate high and low-fidelity approximations of the true objectives (higher fidelity being more costly in practice). We then add surrogate model finetuning capabilities so that at every stage of optimization, each objective's surrogate is refined on a growing buffer of candidate optimal designs instead of being retrained from scratch. Algorithm 1 in the Appendix details our modifications to the HV-KG algorithm.

In order to evaluate the joint contribution of OCL and SC in this framework, we tested four schemes — all of which compute an initial buffer of random candidate designs, and grow the buffer by one newly acquired instance at every optimization iteration. **(i) Standard HV-KG:** The original framework where surrogates are retrained at every iteration with randomly initialized weights; **(ii) SC HV-KG:** incorporating an SC loss term alongside the MLL; **(iii) OCL HV-KG:** the surrogates are pretrained on the initial buffer and finetuned on the expanded buffer, inheriting weights from prior iterations; and **(iv) OCL+SC HV-KG:** combining finetuning with the SC-augmented loss.

Figure 3.1 shows the results of each experiment. OCL+SC leads to the best optimization outcomes (lowest hypervolume regret) and fastest convergence in the MOMF optimization (Figure 3.1). We observe that by finetuning the surrogates each optimization iteration instead of retraining from scratch (Standard HV-KG), both OCL and OCL+SC follow steeper regret curves that converge earlier, arriving at the lowest regret value achieved by HV-KG with only one-sixth the cost. By inheriting the weights from prior iteration, the surrogate model preserves local accuracy in promising design regions, enabling the surrogates to facilitate more accuracy candidate acquisition for the Pareto front. Incorporating SC (Offline w/ SC in 3.1) to align surrogate and ground truth sensitivities is also highly beneficial, as the model reaches nearly the lowest achieved regret, though the convergence rate is not as fast as OCL+SC. Showing each augmentation individually, the synergistic value offered by both together is apparent in the OCL+SC curve, achieving both the lowest log regret and converging with the lowest cost.

Although this demonstration is on a low-dimensional synthetic task, the observed results indicate that continual learning and sensitivity regularization can work in concert, producing more sample- and cost-efficient optimizations. While there are costs associated with retraining the surrogate or the computation and injection of Jacobians for SC, such costs drastically improve the efficiency and outcome of the overall scheme. As dimensionality increases, we expect the benefits illustrated here to compound, a trend confirmed in the higher-dimensional experiments that follow.

3.2 Hybrid training of NNs coupled to process-based equations

We evaluate our framework on two Earth system science tasks where neural networks (NNs) are trained jointly with process-based models. In both settings, an NN parametrizes the connected model, producing parameters for the process-based equation with supervision via loss between the physical model's simulation and observations. Case (A) is a hydrologic model for streamflow and flood prediction introduced by Feng et al. (2022); Case (B) is a plant hydraulics model simulating sap flow (Aboelyazeed et al., 2025). Both resemble systems of ordinary differential equations with long time integrations (730 time steps during training for Case A, 360 time steps for case B). The training loss is defined on streamflow (flow rate in rivers) for Case (A) and vegetation sap flow (movement of water through the tree's vascular system) for case (B). We consider two NNs: (i) the parameterization NN, the primary object of optimization, which maps static geoscientific covariates (e.g., terrain slope, soil texture) to physical parameters needed by the numerical models, and (ii) a surrogate NN, trained to approximate the full physical model. The surrogate NN takes dynamic drivers (precipitation, temperature, radiation), time-invariant geospatial attributes (soil texture, elevation), and physical parameters as inputs to return either streamflow or sap flow. In addition, gradients of the physical model simulations with respect to physical parameters were computed using PyTorch automatic differentiation for sensitivity-constrained training. To avoid the prohibitive cost of computing full-Jacobians across all time steps, we sampled a representative subset for gradient evaluation.

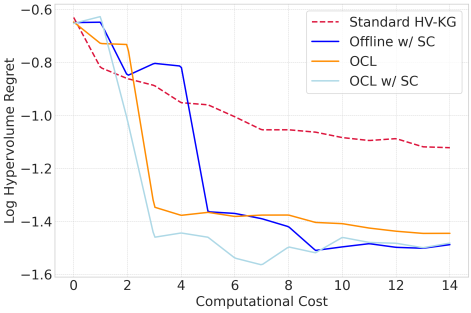

Figure 1: HV-KG log hypervolume regret versus cumulative analytical cost on the Branin-Currin MOMF benchmark. We compare the original HV-KG framework, HV-KG with online continual learning (OCL), HV-KG with sensitivity constraints (SC), and HV-KG with both OCL and SC. Log hypervolume regret quantifies optimization error and represents the gap between the hypervolume of the analytically derived true Pareto front and that of the identified solution set at a given cost. Standard HV-KG plateaus near regret ≈ -1.3 at cost ≈ 51, while OCL and OCL+SC converge faster and to lower regret. Note the cost axis only shows cost accrued during the acquisition of new candidate solutions (specifically, cost for evaluating the objective functions at different fidelities) up to the maximum cost budget and does not include the cost to assemble the initial surrogate training buffer (this initial cost is equal across all experiments here).

At a schedule time, we finetune the surrogate model based on a data loss between the output of the surrogate model and the full physical model, and optionally an SC loss for the gradient of the output with respect to the physical parameters. As the whole system is coupled end-to-end and trained using gradient descent, the gradient information through the surrogate model is part of the gradient chain to train the parameterization NN — we do not have direct observations for the parameters output by the NN. The algorithms and train-test splits are detailed in Appendix A.2. Each epoch contains approximately 194 NN parameter updates for the hydrological model, and 35 updates for the ecosystem model.

We incorporate online surrogate training and introduce a hyperparameter, n_{SC}, specifying the number of epochs after which the surrogate model is finetuned. Whenever the surrogate is pretrained or finetuned, SC is applied when that option is enabled. We compare multiple schemes. **(i) Offline:**

pretraining the surrogate on an initial buffer of physical model simulations and then used to replace the physical model during NN optimization, with its weights kept frozen; **(ii) OCL:** the surrogate model was pretrained and was subsequently finetuned (inheriting weights from previous training) once every n_{SC} epoch on an iteratively refreshed buffer with new samples computed by the physical model; **(iii) OCL+SC:** adding loss into the surrogate training; **(iv) Offline+SC:** for comparison, we also ran with a surrogate model trained with a large initial buffer and sensitivity constraint. This scheme represents setup comparable to previous gradient-informed surrogate models like SC-FNO, DINO, DE-DeepONet or Sobolev training mentioned above. Furthermore, we include the high benchmark when the full **(v) physical:** model was used without surrogate.

3.2.1 Hydrologic model

Using the offline (pretrained) surrogate model with or without sensitivity constraint (SC) or the online trained model without SC (OCL) all cause large gaps in optimization metrics from the benchmark (training with the full physical model) in the hydrologic case (Table 1). The offline surrogate hydrologic model leads to the worst result, scoring an R^2 of 0.45 compared to the benchmark level of 0.72. R^2 of 0.45 is not an operationally acceptable (”poor”) metric for flood forecasting while above 0.7 is considered good (Moriasi et al., 2015). This illustrates one of the key challenges of non-stationarity in this setting, as the region of interest to the parameterization NN in the parameter space is guaranteed to shift as the joint training proceeds. Incorporating either OCL or SC alone elevated R^2 to 0.63 and 0.65, respectively, which is a substantial gain. However, this gap is still inadequate. This means incorporating SC alone was useful,but the initial training buffer cannot satisfy the needs of optimization, especially for the high-dimensional surrogate. Incorporating OCL alone is also insufficient, which means adaptively casting points near the search optimum cannot provide adequate guidance, and even state-of-the-art adaptive learning strategies cannot produce a surrogate model of sufficient quality for this demanding task. It may be understood that the gradient of the surrogate model is part of the gradient chain for training the parameterization NN, and the learned surrogate sensitivities are often incorrect even if the prediction has high metrics.

Applying both OCL and SC nearly closes the gap of surrogate-assisted learning to the benchmark. By explicitly aligning surrogate sensitivities to those of the physical model, the surrogate achieves both high fidelity in predictive skill and stability in guiding parameter learning. This means the two options can be superimposed to leverage their compound benefits. Even with a low frequency of SC injection, $n_{SC} = 10$, the joint model performance ($R^2 = 0.68$ is noticeably higher than offline, OCL or offline+SC) (Table 1). At higher levels, the learning results are operationally sound. To completely reach the benchmark level, in the future, we envision a limited number of full-model iterations can be added toward the end to finish the optimization.

3.2.2 Ecosystem model

The ecosystem model joint training is also strongly aided by the use of OCL and SC, with improvements apparent for each procedure. The offline frozen surrogate model entirely failed the optimization, resulting in an R^2 of 0.21 and a large bias (-0.059) for the combined system, when the full model indicates the limit R^2 to be around 0.438 (see Table 1). The offline SC injection (offline+SC) elevated the performance up to an R^2 of 0.35, though it still lagged behind the benchmark. This is a prime example of pretrained surrogate model losing fidelity and guidance capability, and suggests that the system is too high dimensional and too complex to be captured by limited initial buffer. With online training only, R^2 increased significantly to 0.41 and the bias was reduced, underscoring the value of providing better fidelity near the optimality. The gap to benchmark level was further narrowed when employing online training with SC (OCL+SC). Injecting SC once per epoch ($n_{SC} = 1$), raised R^2 to 0.42, applying it every 2 epochs ($n_{SC} = 2$), pushed the performance further to an R^2 of 0.45, surpassing the benchmark.

3.2.3 Common observations

Both experiments suggest OCL+SC can almost provide a model with the same effect as the full physical model even in high-dimensional case. The surrogate model needs to nearly perfectly reproduces the behaviors and sensitivities to reach this performance. The optimization constantly probes new locations around the present optimum and, with correct sensitivities from SC training, OCL+SC provides good guidance for new test points as ensured by Taylor series expansion. We note that OCL+SC uses only $1/30$ of the computational cost in the ecosystem case (see timing in Table 5). Because of the complicated nonlinear solver and the needs to solve the adjoint equation for the gradients, the full ecosystem model is quite expensive – the optimization experiment using the full physical model requires 2 days on a V100 GPU. The surrogate model reduces an optimization run to almost 2 hours, so many learning experiments become plausible again. The costs of SC with $n_{SC} = 1$ is calculated as 37s per epoch or 26% of the time of OCL alone. Incorporating OCL and SC seem to be a necessary to enable a model like this for learning.

Both models show that merely infrequent injection of SC could already produce noticeable benefits. As we varied n_{SC} from 1 to 10, as expected, we witness a change in performance. However, even with $n_{SC} = 10$, a gradient loss insertion once after hundreds of NN parameter updates, we still achieve a decently high performance (0.68 for the hydrologic model and 0.39 for the ecosystem model). Moreover, the computational cost is affordable. From OCL without SC to OCL+SC with $n_{SC} = 1$ raises the runtime per epoch by $\tilde{1}3$%, not an extensive amount of cost to pay to obtain performance to raise R^2 from 0.63 to 0.71 for the hydrological model.

Table 1: Performance for the differentiable hydrological and ecosystem models with varying frequencies of gradient injection. Each differentiable model is trained to 50 epochs. Metrics including R^2 and bias are evaluated between the joint model outputs (streamflow or sap flow) and the observations using the full physical model and the trained parameterization NN. For online training with SC, we evaluate performance for continued surrogate training every 1, 2, 5, and 10 epochs. The time required to train the parameterization NN for one epoch in each case is given in seconds. Results for R^2 and Bias are reported as mean ± std for 5 random seeds.

	Hydrologic Model		**Ecosystem Model**	
	R^2	Bias	R^2	Bias
Physical Model (Benchmark)	0.72	0.605	0.438	-0.011
Offline	0.45 ± 0.07	0.70 ± 0.05	0.21 ± 0.12	-0.059 ± 0.012
OCL	0.63 ± 0.03	0.69 ± 0.05	0.41 ± 0.02	-0.022 ± 0.005
Offline + SC	0.65 ± 0.05	0.69 ± 0.05	0.35 ± 0.04	-0.030 ± 0.015
OCL + SC (1)	0.69 ± 0.02	0.64 ± 0.02	0.42 ± 0.03	-0.019 ± 0.005
OCL + SC (2)	0.71 ± 0.02	0.62 ± 0.02	0.45 ± 0.02	-0.014 ± 0.003
OCL + SC (5)	0.70 ± 0.03	0.63 ± 0.02	0.43 ± 0.02	-0.017 ± 0.003
OCL + SC (10)	0.68 ± 0.04	0.67 ± 0.03	0.39 ± 0.06	-0.023 ± 0.005

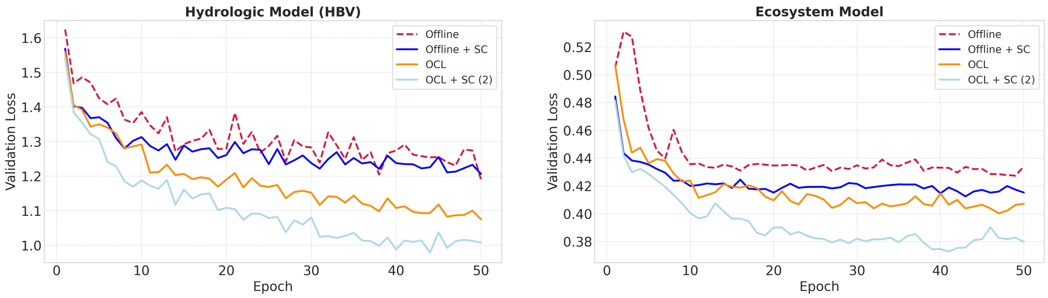

Figure 2: Validation loss during differentiable model training when using a surrogate pretrained offline with or without SC, and in an OCL scheme with or without SC. Results for the hydrological and ecosystem models are given in the left and right panels, respectively.

3.3 FUSE (JOINT GENERATIVE INVERSION-SIMULATION TRAINING)

We implemented the original FUSE design (Lingsch et al., 2024) while introducing the following two modifications to enhance its performance: 1) online FMPE training using data generated by the surrogate: the forward surrogate is used to generate new samples during training to be added to the buffer and boost the training data size for the FMPE inverse model; 2) using a sensitivity-constrained

FNO for the surrogate model (Behroozi et al., 2024) (details in Appendix A.3). For all configurations we tested, the full physical model was called only upon initial training data preparation, not iteratively during training. We evaluated these configurations on a benchmark 2D steady-state Darcy flow equation with Dirichlet boundary conditions, used in earlier FNO research (Li et al., 2024). Following their approach, we generated 1000 paired samples of (a,u) on a 32 x 32 spatial grid to train and evaluate different cases, where $a(x, y)$ represents the spatially varying diffusion coefficient and $u(x, y)$ represents the solution field. In FUSE setup with SC injected, the FNO is trained to predict the solution field u from the diffusion coefficient a using a combined loss function ($L_{\text{data}} + \lambda L_{\text{SC}}$). Where L_{data} measures the discrepancy between the predicted and true-u fields, L_{SC} penalizes mismatches between predicted and true sensitivities $\partial u/\partial a$, and the weighting factor λ is set to 1. The FMPE model is trained in parallel to approximate the posterior distribution $\rho(u|a)$ given spatially masked observations of u, using Fourier-projected representations of u as conditional inputs.

Table 2: Results for FUSE applied to Darcy flow (steady state) with a 25% spatial mask applied to the u field input to the FMPE. Continuous Ranked Probability Score (CRPS) quantifies how closely the generative FMPE's parameter estimates matches the true a field, while Relative L_1 and L_2 are the error norms of the surrogate-simulated u field. Rows labeled **True** a report forward surrogate evaluation using ground-truth diffusion coefficients, while rows labeled **Predicted** a reports the full FUSE pipeline evaluation, where FMPE first predicts a, and the surrogate then maps $a \rightarrow u$. The difference between the two quantifies the impact of a generation error on the forward simulation u.

Model	**Experiment Setup**	**CRPS**	**Rel. L$_1$ Error**	**Rel. L$_2$ Error**
FUSE	True a	—	2.91 ± 0.86	3.71 ± 0.89
	Predicted a	0.72 ± 0.16	8.40 ± 2.94	9.72 ± 2.83
Online FUSE	True a	—	2.95 ± 0.88	3.76 ± 0.90
	Predicted a	0.74 ± 0.17	9.64 ± 3.48	10.84 ± 3.30
Online FUSE + SC	True a	—	2.59 ± 0.84	3.28 ± 0.83
	Predicted a	0.48 ± 0.15	5.65 ± 2.63	6.50 ± 2.48

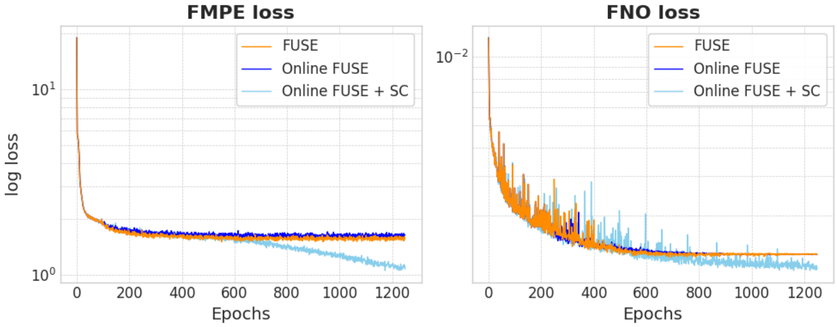

Figure 3: Validation loss during joint training for three configurations; FUSE, Online FUSE, and Online FUSE+SC.

OCL+SC shows clear advantage for the FMPE training as well as the surrogate training over standard FUSE (Table 2 and Figure 3). The FUSE and Online FUSE (OCL without SC) cases produced very similar results, and the latter has even slightly larger errors both for the FMPE (CRPS) and the forward simulation (L_1 and L_2 errors) (see the first two rows for each configuration in Table 2 and the orange/blue lines in Figure 2). This means using a surrogate model that is trained without SC does not produce simulations of high enough quality to inform FMPE. In contrast, Online Training w/ SC (Online FUSE+SC) lowered the CRPS to 0.478, L_1 to 5.65 and L_2 to 6.50. All of them are around 34% lower than either standard FUSE or Online FUSE without SC. The OCL+SC continues to push down FMPE error around 600 epochs, when the offline or the OCL without SC models stall (Figure 3). Note here the new training data for FMPE is provided by the surrogate model only and not additional full physical model runs. This means the generative FMPE can indeed benefit from a higher-quality surrogate model trained on both data and the sensitivities. Even if the surrogate model loss appears only slightly better (Figure 3 Right), its better internal relationships and robustness to perturbations (Behroozi et al., 2024) allowed it to generate more accurate new examples to improve FMPE. In a positive feedback, the more accurately generated a field, in turn, substantially lowered the forward L_1 and L_2 errors.

3.4 Further Discussion and Conclusion

We have tested Online Continual Learning with Sensitivity Constraint (OCL+SC) across four different datasets and three algorithmically distinct classes of long-horizon optimization frameworks that use surrogate models in different ways: multifidelity Pareto-front exploration, hybrid training, learning inverse maps. None of these frameworks (or many other for this matter) previously employed SC. Nor did any of the related studies comment on algorithms similar to SC as an option for substantial improvements. Diverse strategies in active learning (Ren et al., 2021) work similarly as OCL and the acquisition functions in the MOMF-BO case, but no previous effort employed SC. Yet, in each case, the incorporation of SC for surrogate training produced immediate and unmistakable improvements, with OCL+SC approaching an equivalent tool as the full physical model with orders-of-magnitude lower cost. Therefore, we argue that the incorporation of SC represents an underexplored opportunity to usher in widespread performance gains and practical utility across scientific and engineering domains. Considering recent advances in hyper-efficient AI-enabled high-resolution solutions to PDEs, e.g., neural operators, and the need of AI agents for world models, the applicable scenarios of SC in complex optimization tasks should only grow rapidly.

While sensitivity-based regularizers themselves have been proposed before, our work is the first to demonstrate their effectiveness when embedded in online continual surrogate learning loops. In this regime, surrogate accuracy is not only about pointwise fit but directly influences the future trajectory of the optimization. It was not obvious that sensitivity constraints and OCL would reinforce each other. Our ablations on different SC frequencies confirm that the combination is synergistic, creating a new design space for long-horizon surrogate-aided optimization. To our knowledge, no prior work has established this. As compared to earlier work using pretrained sensitivity-aware surrogate models (Behroozi et al., 2024; Qiu et al., 2024; O'Leary-Roseberry et al., 2024), the usefulness of OCL algorithms means we do not need to collect comprehensive and exhaustive pretraining data for the surrogate, which can be implausible for high-dimensional and long-horizon optimization task. Instead, we can spend limited computational resources along the main optimization in a more targeted fashion. The fact that the SC adds strong benefits beyond OCL alone liberates AI surrogates for many high-resolution, high-computational-demand applications.

Previous studies of surrogate models primarily focused on efficiency and forward predictions, but have severely underestimated the importance of the correct internal relationships (quantified by sensitivities), generalizability against perturbations, and their accumulative effects. In agreement with Cao et al. (2025), we argue this is a primary weakness with the majority of present surrogate models. In fact, as we searched the literature, while neural operators are widely used for forward prediction, their use within inverse algorithms (not directly learning inverse maps) is still relatively rare. The purpose of a model in an optimization is to explore, to perturb, and to guide. If we use surrogate model to generate training data the inverse mapping as in the FUSE case above, it clearly also requires the model to be error-resistant when the inputs are perturbed. The fact that seemingly high-scoring surrogate model can do poorly in these tasks is not emphasized enough in the community. The tested frameworks have a long optimization horizon. Although at a given (especially initial) step, the direct prediction error may be very low, the accumulative and sensitivity error mean that using the surrogate trained on data alone for guidance is futile.

A critical element for newly enabled SC is the rapid computation of gradients from numerical models. Differentiable programming can support the calculation of high dimensional Jacobian, but finite difference, adjoint-based methods (Behroozi et al., 2024), and low-rank approximations (Qiu et al., 2024) can also be used. As differentiable programming is gaining popularity in many domains (Ramsundar et al., 2021; Gelbrecht et al., 2023; Shen et al., 2023; AlQuraishi & Sorger, 2021; Zhu et al., 2024), we expect more and more numerical models to directly support it.

Reproducibility Statement

The Branin–Currin HV-KG experiments build on publicly available code from https://github.com/meta-pytorch/botorch, and the FUSE experiments build on https://github.com/camlab-ethz/FUSE. Both repositories can be adapted with the modifications described in this paper to reproduce the reported OCL and SC results. The differentiable hydrologic and ecosystem model experiments rely on code that is currently under internal development; this code will be released publicly upon completion of the ongoing open-sourcing process to enable full replication of these results.

References

Doaa Aboelyazeed, Chonggang Xu, Lianhong Gu, Xiangzhong Luo, Jiangtao Liu, Kathryn Lawson, and Chaopeng Shen. Inferring Plant Acclimation and Improving Model Generalizability With Differentiable Physics-Informed Machine Learning of Photosynthesis. *Journal of Geophysical Research: Biogeosciences*, 130(7): e2024JG008552, 2025. ISSN 2169-8961. doi: 10.1029/2024JG008552. URL https://onlinelibrary.wiley.com/doi/abs/10.1029/2024JG008552. _eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2024JG008552.

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pp. 11817–11826. Curran Associates Inc., Red Hook, NY, USA, December 2019.

Mohammed AlQuraishi and Peter K. Sorger. Differentiable biology: using deep learning for biophysics-based and data-driven modeling of molecular mechanisms. *Nature Methods*, 18 (10):1169–1180, October 2021. ISSN 1548-7105. doi: 10.1038/s41592-021-01283-4. URL https://doi.org/10.1038/s41592-021-01283-4.

Brandon Amos, Ivan Dario Jimenez Rodriguez, Jacob Sacks, Byron Boots, and J. Zico Kolter. Differentiable MPC for End-to-end Planning and Control, October 2019. URL http://arxiv.org/abs/1810.13400. arXiv:1810.13400 [cs].

Akio Arakawa. The Cumulus Parameterization Problem: Past, Present, and Future. *Journal of Climate*, 17(13):2493–2525, July 2004. ISSN 0894-8755, 1520-0442. doi: 10.1175/1520-0442(2004)017⟨2493:RATCPP⟩2.0.CO;2. URL https://journals.ametsoc.org/view/journals/clim/17/13/1520-0442_2004_017_2493_ratcpp_2.0.co_2.xml. Publisher: American Meteorological Society Section: Journal of Climate.

Lynton Ardizzone, Jakob Kruse, Sebastian Wirkert, Daniel Rahner, Eric W. Pellegrini, Ralf S. Klessen, Lena Maier-Hein, Carsten Rother, and Ullrich Köthe. Analyzing Inverse Problems with Invertible Neural Networks, February 2019. URL http://arxiv.org/abs/1808.04730. arXiv:1808.04730 [cs].

Kamyar Azizzadenesheli, Nikola Kovachki, Zongyi Li, Miguel Liu-Schiaffini, Jean Kossaifi, and Anima Anandkumar. Neural operators for accelerating scientific simulations and design. *Nature Reviews Physics*, 6(5):320–328, May 2024. ISSN 2522-5820. doi: 10.1038/s42254-024-00712-5. URL https://www.nature.com/articles/s42254-024-00712-5. Publisher: Nature Publishing Group.

Abdolmehdi Behroozi, Chaopeng Shen, and Daniel Kifer. Sensitivity-Constrained Fourier Neural Operators for Forward and Inverse Problems in Parametric Differential Equations. In *The Thirteenth International Conference on Learning Representations*, October 2024. URL https://openreview.net/forum?id=DPzQ5n3mNm.

Lianghao Cao, Thomas O'Leary-Roseberry, and Omar Ghattas. Derivative-informed neural operator acceleration of geometric mcmc for infinite-dimensional bayesian inverse problems. *Journal of Machine Learning Research*, 26(78):1–68, 2025. URL http://jmlr.org/papers/v26/24-0745.html.

B. O. Christoffersen, M. Gloor, S. Fauset, N. M. Fyllas, D. R. Galbraith, T. R. Baker, B. Kruijt, L. Rowland, R. A. Fisher, O. J. Binks, S. Sevanto, C. Xu, S. Jansen, B. Choat, M. Mencuccini, N. G. McDowell, and P. Meir. Linking hydraulic traits to tropical forest function in a size-structured and trait-driven model (TFS v.1-Hydro). *Geosci. Model Dev.*, 9(11):4227–4255, November 2016. ISSN 1991-9603. doi: 10.5194/gmd-9-4227-2016. URL https://gmd.copernicus.org/articles/9/4227/2016/. Publisher: Copernicus Publications.

Wojciech M. Czarnecki, Simon Osindero, Max Jaderberg, Grzegorz Swirszcz, and Razvan Pascanu. Sobolev training for neural networks. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/758a06618c69880a6cee5314ee42d52f-Paper.pdf.

Sam Daulton, Maximilian Balandat, and Eytan Bakshy. Hypervolume Knowledge Gradient: A Lookahead Approach for Multi-Objective Bayesian Optimization with Partial Information. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 7167–7204. PMLR, July 2023. URL https://proceedings.mlr.press/v202/daulton23a.html. ISSN: 2640-3498.

Filipe de Avila Belbute-Peres, Kevin Smith, Kelsey Allen, Josh Tenenbaum, and J. Zico Kolter. End-to-End Differentiable Physics for Learning and Control. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/hash/842424a1d0595b76ec4fa03c46e8d755-Abstract.html.

Changyu Deng, Yizhou Wang, Can Qin, Yun Fu, and Wei Lu. Self-directed online machine learning for topology optimization. *Nature Communications*, 13(1):388, January 2022. ISSN 2041-1723. doi: 10.1038/s41467-021-27713-7. URL https://www.nature.com/articles/s41467-021-27713-7. Publisher: Nature Publishing Group.

Dapeng Feng, Jiangtao Liu, Kathryn Lawson, and Chaopeng Shen. Differentiable, learnable, regionalized process-based models with multiphysical outputs can approach state-of-the-art hydrologic prediction accuracy. *Water Resources Research*, 58(10): e2022WR032404, 2022. ISSN 1944-7973. doi: 10.1029/2022WR032404. URL https://onlinelibrary.wiley.com/doi/abs/10.1029/2022WR032404. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2022WR032404.

Maximilian Gelbrecht, Alistair White, Sebastian Bathiany, and Niklas Boers. Differentiable programming for Earth system modeling. *Geoscientific Model Development*, 16(11):3123–3135, June 2023. ISSN 1991-959X. doi: 10.5194/gmd-16-3123-2023. URL https://gmd.copernicus.org/articles/16/3123/2023/. Publisher: Copernicus GmbH.

Hoshin Vijai Gupta, Soroosh Sorooshian, and Patrice Ogou Yapo. Toward improved calibration of hydrologic models: Multiple and noncommensurable measures of information. *Water Resources Research*, 34(4):751–763, 1998. ISSN 1944-7973. doi: 10.1029/97WR03495. URL https://onlinelibrary.wiley.com/doi/abs/10.1029/97WR03495. _eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/97WR03495.

Frédéric Hourdin, Thorsten Mauritsen, Andrew Gettelman, Jean-Christophe Golaz, Venkatramani Balaji, Qingyun Duan, Doris Folini, Duoying Ji, Daniel Klocke, Yun Qian, Florian Rauser, Catherine Rio, Lorenzo Tomassini, Masahiro Watanabe, and Daniel Williamson. The Art and Science of Climate Model Tuning. *Bulletin of the American Meteorological Society*, 98(3):589–602, March 2017. ISSN 0003-0007, 1520-0477. doi: 10.1175/BAMS-D-15-00135.1. URL https://journals.ametsoc.org/view/journals/bams/98/3/bams-d-15-00135.1.xml. Publisher: American Meteorological Society Section: Bulletin of the American Meteorological Society.

Mike Innes, Alan Edelman, Keno Fischer, Chris Rackauckas, Elliot Saba, Viral B. Shah, and Will Tebbutt. A Differentiable Programming System to Bridge Machine Learning and Scientific Computing, July 2019. URL http://arxiv.org/abs/1907.07587. arXiv:1907.07587 [cs].

Faran Irshad, Stefan Karsch, and Andreas Döpp. Leveraging Trust for Joint Multi-Objective and Multi-Fidelity Optimization. *Machine Learning: Science and Technology*, 5(1):015056, March 2024. ISSN 2632-2153. doi: 10.1088/2632-2153/ad35a4. URL http://arxiv.org/abs/2112.13901. arXiv:2112.13901 [cs].

George Em Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, June 2021. ISSN 2522-5820. doi: 10.1038/s42254-021-00314-5. URL https://www.nature.com/articles/s42254-021-00314-5. Publisher: Nature Publishing Group.

Jens Kattge and Wolfgang Knorr. Temperature acclimation in a biochemical model of photosynthesis: a reanalysis of data from 36 species. *Plant, Cell & Environment*, 30(9): 1176–1190, 2007. ISSN 1365-3040. doi: 10.1111/j.1365-3040.2007.01690.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-3040.2007.01690.x. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1365-3040.2007.01690.x.

Jens et al. Kattge. TRY plant trait database – enhanced coverage and open access. *Global Change Biology*, 26(1):119–188, January 2020. ISSN 1354-1013. doi: 10.1111/gcb.14904. URL https://doi.org/10.1111/gcb.14904. Publisher: John Wiley & Sons, Ltd.

Leifur Leifsson and Slawomir Koziel. Multi-fidelity design optimization of transonic airfoils using physics-based surrogate modeling and shape-preserving response prediction. *Journal of Computational Science*, 1(2):98–106, June 2010. ISSN 1877-7503. doi: 10.1016/j.jocs.2010.03.007. URL https://www.sciencedirect.com/science/article/pii/S1877750310000098.

Zongyi Li, Nikola Borislavov 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*, January 2021. URL https://openreview.net/forum?id=c8P9NQVtmnO.

Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-Informed Neural Operator for Learning Partial Differential Equations. *ACM / IMS J. Data Sci.*, 1(3):9:1–9:27, May 2024. doi: 10.1145/3648506. URL https://dl.acm.org/doi/10.1145/3648506.

Levi E. Lingsch, Dana Grund, Siddhartha Mishra, and Georgios Kissas. FUSE: Fast Unified Simulation and Estimation for PDEs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, November 2024. URL https://openreview.net/forum?id=dbnEf790Kv&referrer=%5Bthe%20profile%20of%20Levi%20E.%20Lingsch%5D(%2Fprofile%3Fid%3D~Levi_E._Lingsch1).

Zichen Liu, Guoji Fu, Chao Du, Wee Sun Lee, and Min Lin. Continual Reinforcement Learning by Planning with Online World Models, July 2025. URL http://arxiv.org/abs/2507.09177. arXiv:2507.09177 [cs].

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, March 2021. ISSN 2522-5839. doi: 10.1038/s42256-021-00302-5. URL https://www.nature.com/articles/s42256-021-00302-5.

Amandine Marrel and Bertrand Iooss. Probabilistic surrogate modeling by Gaussian process: A review on recent insights in estimation and validation. *Reliability Engineering & System Safety*, 247:110094, July 2024. ISSN 0951-8320. doi: 10.1016/j.ress.2024.110094. URL https://www.sciencedirect.com/science/article/pii/S0951832024001686.

Belinda E. Medlyn, Remko A. Duursma, Derek Eamus, David S. Ellsworth, I. Colin Prentice, Craig V M Barton, Kristine Y. Crous, Paolo De Angelis, Michael Freeman, and Lisa Wingate. Reconciling the optimal and empirical approaches to modelling stomatal conductance. *Global Change Biology*, 17(6):2134–2144, June 2011. ISSN 1354-1013. doi: 10.1111/j.1365-2486.2010.02375.x. URL http://www.scopus.com/inward/record.url?scp=79955035036&partnerID=8YFLogxK.

Lucas Thibaut Meyer, Marc Schouler, Robert Alexander Caulk, Alejandro Ribes, and Bruno Raffin. Training Deep Surrogate Models with Large Scale Online Learning. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 24614–24630. PMLR, July 2023. URL https://proceedings.mlr.press/v202/meyer23b.html. ISSN: 2640-3498.

D. N. Moriasi, W.Gitau Margaret, Naresh Pai, and Prasad Daggupati. Hydrologic and water quality models: Performance measures and evaluation criteria. *Transactions of the ASABE*, 58(6):1763–1785, December 2015. ISSN 21510032, 21510040. doi: 10.13031/trans.58.10715. URL http://elibrary.asabe.org/abstract.asp?aid=46548&t=3&dabs=Y&redir=&redirType=.

J. Muñoz Sabater. ERA5-Land hourly data from 1950 to present, July 2019. URL https://cds.climate.copernicus.eu/datasets/reanalysis-era5-land?tab=overview.

Anusha Nagabandi, Chelsea Finn, and Sergey Levine. Deep Online Learning via Meta-Learning: Continual Adaptation for Model-Based RL, January 2019. URL http://arxiv.org/abs/1812.07671. arXiv:1812.07671 [cs].

Andrew J. Newman, Kevin Sampson, Martyn Clark, A. Bock, Roland Viger, D. Blodgett, Nans Addor, and M. Mizukami. CAMELS: Catchment Attributes and MEteorology for Large-sample Studies, June 2022. URL https://zenodo.org/records/15529996.

Thomas O'Leary-Roseberry, Peng Chen, Umberto Villa, and Omar Ghattas. Derivative-informed neural operator: An efficient framework for high-dimensional parametric derivative learning. *Journal of Computational Physics*, 496:112555, 2024. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2023.112555. URL https://www.sciencedirect.com/science/article/pii/S0021999123006502.

Laura Poggio, Luis M. de Sousa, Niels H. Batjes, Gerard B. M. Heuvelink, Bas Kempen, Eloi Ribeiro, and David Rossiter. SoilGrids 2.0: Producing soil information for the globe with quantified spatial uncertainty. *SOIL*, 7(1):217–240, June 2021. ISSN 2199-3971. doi: 10.5194/soil-7-217-2021. URL https://soil.copernicus.org/articles/7/217/2021/. Publisher: Copernicus GmbH tex.ids= poggio2021soilgridsa.

R. Poyatos, V. Granda, V. Flo, M. A. Adams, B. Adorján, D. Aguadé, M. P. M. Aidar, S. Allen, M. S. Alvarado-Barrientos, K. J. Anderson-Teixeira, L. M. Aparecido, M. A. Arain, I. Aranda, H. Asbjornsen, R. Baxter, E. Beamesderfer, Z. C. Berry, D. Berveiller, B. Blakely, J. Boggs, G. Bohrer, P. V. Bolstad, D. Bonal, R. Bracho, P. Brito, J. Brodeur, F. Casanoves, J. Chave, H. Chen, C. Cisneros, K. Clark, E. Cremonese, H. Dang, J. S. David, T. S. David, N. Delpierre, A. R. Desai, F. C. Do, M. Dohnal, J.-C. Domec, S. Dzikiti, C. Edgar, R. Eichstaedt, T. S. El-Madany, J. Elbers, C. B. Eller, E. S. Euskirchen, B. Ewers, P. Fonti, A. Forner, D. I. Forrester, H. C. Freitas, M. Galvagno, O. Garcia-Tejera, C. P. Ghimire, T. E. Gimeno, J. Grace, A. Granier, A. Griebel, Y. Guangyu, M. B. Gush, P. J. Hanson, N. J. Hasselquist, I. Heinrich, V. Hernandez-Santana, V. Herrmann, T. Hölttä, F. Holwerda, J. Irvine, S. Isarangkool Na Ayutthaya, P. G. Jarvis, H. Jochheim, C. A. Joly, J. Kaplick, H. S. Kim, L. Klemedtsson, H. Kropp, F. Lagergren, P. Lane, P. Lang, A. Lapenas, V. Lechuga, M. Lee, C. Leuschner, J.-M. Limousin, J. C. Linares, M.-L. Linderson, A. Lindroth, P. Llorens, Á. López-Bernal, M. M. Loranty, D. Lüttschwager, C. Macinnis-Ng, I. Maréchaux, T. A. Martin, A. Matheny, N. McDowell, S. McMahon, P. Meir, I. Mészáros, M. Migliavacca, P. Mitchell, M. Mölder, L. Montagnani, G. W. Moore, R. Nakada, F. Niu, R. H. Nolan, R. Norby, K. Novick, W. Oberhuber, N. Obojes, A. C. Oishi, R. S. Oliveira, R. Oren, J.-M. Ourcival, T. Paljakka, O. Perez-Priego, P. L. Peri, R. L. Peters, S. Pfautsch, W. T. Pockman, Y. Preisler, K. Rascher, G. Robinson, H. Rocha, A. Rocheteau, A. Röll, B. H. P. Rosado, L. Rowland, A. V. Rubtsov, S. Sabaté, Y. Salmon, R. L. Salomón, E. Sánchez-Costa, K. V. R. Schäfer, B. Schuldt, A. Shashkin, C. Stahl, M. Stojanović, J. C. Suárez, G. Sun, J. Szatniewska, F. Tatarinov, M. Tesař, F. M. Thomas, P. Tor-ngern, J. Urban, F. Valladares, C. van der Tol, I. van Meerveld, A. Varlagin, H. Voigt, J. Warren, C. Werner, W. Werner, G. Wieser, L. Wingate, S. Wullschleger, K. Yi, R. Zweifel, K. Steppe, M. Mencuccini, and J. Martínez-Vilalta. Global transpiration data from sap flow measurements: the SAPFLUXNET database. *Earth System Science Data*, 13(6):2607–2649, 2021. doi: 10.5194/essd-13-2607-2021. URL https://essd.copernicus.org/articles/13/2607/2021/.

Bryton Praslicka, Cong Ma, and Narges Taran. A Computationally Efficient High-Fidelity Multi-Physics Design Optimization of Traction Motors for Drive Cycle Loss Minimization. *IEEE Transactions on Industry Applications*, 59(2):1351–1360, March 2023. ISSN 1939-9367. doi: 10.1109/TIA.2022.3220554. URL https://ieeexplore.ieee.org/abstract/document/9942344.

Yuan Qiu, Nolan Bridges, and Peng Chen. Derivative-enhanced Deep Operator Network. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, November 2024. URL https://openreview.net/forum?id=WAiqLGfqX6.

Christopher Rackauckas, Yingbo Ma, Julius Martensen, Collin Warner, Kirill Zubov, Rohit Supekar, Dominic Skinner, Ali Ramadhan, and Alan Edelman. Universal Differential Equations for Scientific Machine Learning, November 2021. URL http://arxiv.org/abs/2001.04385. arXiv:2001.04385 [cs].

Bharath Ramsundar, Dilip Krishnamurthy, and Venkatasubramanian Viswanathan. Differentiable physics: A position piece. *ArXiv*, abs/2109.07573, 2021. URL https://api.semanticscholar.org/CorpusID:237532722.

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B. Gupta, Xiaojiang Chen, and Xin Wang. A Survey of Deep Active Learning. *ACM Comput. Surv.*, 54(9), October 2021. ISSN 0360-0300. doi: 10.1145/3472291. URL https://doi.org/10.1145/3472291. Place: New York, NY, USA Publisher: Association for Computing Machinery.

Max Schwenzer, Muzaffer Ay, Thomas Bergs, and Dirk Abel. Review on model predictive control: an engineering perspective. *The International Journal of Advanced Manufacturing Technology*, 117(5):1327–1349, November 2021. ISSN 1433-3015. doi: 10.1007/s00170-021-07682-3. URL https://doi.org/10.1007/s00170-021-07682-3.

Chaopeng Shen, Alison P. Appling, Pierre Gentine, Toshiyuki Bandai, Hoshin Gupta, Alexandre Tartakovsky, Marco Baity-Jesi, Fabrizio Fenicia, Daniel Kifer, Li Li, Xiaofeng Liu, Wei Ren, Yi Zheng, Ciaran J. Harman, Martyn Clark, Matthew Farthing, Dapeng Feng, Praveen Kumar, Doaa Aboelyazeed, Farshid Rahmani, Yalan Song, Hylke E. Beck, Tadd Bindas, Dipankar Dwivedi, Kuai Fang, Marvin Höge, Chris Rackauckas, Binayak Mohanty, Tirthankar Roy, Chonggang Xu, and Kathryn Lawson. Differentiable modelling to unify machine learning and physical models for geosciences. *Nature Reviews Earth & Environment*, 4(8):552–567, August 2023. ISSN 2662-138X. doi: 10.1038/s43017-023-00450-9. URL https://www.nature.com/articles/s43017-023-00450-9. Publisher: Nature Publishing Group.

Shion Takeno, Hitoshi Fukuoka, Yuhki Tsukada, Toshiyuki Koyama, Motoki Shiga, Ichiro Takeuchi, and Masayuki Karasuyama. Multi-fidelity Bayesian Optimization with Max-value Entropy Search and its Parallelization. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 9334–9345. PMLR, November 2020. URL https://proceedings.mlr.press/v119/takeno20a.html. ISSN: 2640-3498.

Albert Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. Society for Industrial and Applied Mathematics, USA, November 2004. ISBN 978-0-89871-572-9.

G. Gary Wang and S. Shan. Review of Metamodeling Techniques in Support of Engineering Design Optimization. In *American Society of Mechanical Engineers Digital Collection*, pp. 415–426. American Society of Mechanical Engineers Digital Collection, June 2008. doi: 10.1115/DETC2006-99412. URL https://dx.doi.org/10.1115/DETC2006-99412.

Sifan Wang, Hanwen Wang, and Paris Perdikaris. Learning the solution operator of parametric partial differential equations with physics-informed DeepONets. *Science Advances*, 7(40): eabi8605, October 2021. ISSN 2375-2548. doi: 10.1126/sciadv.abi8605. URL https://www.science.org/doi/10.1126/sciadv.abi8605.

Martin Wistuba, Nicolas Schilling, and Lars Schmidt-Thieme. Scalable Gaussian process-based transfer surrogates for hyperparameter optimization. *Machine Learning*, 107(1):43–78, January 2018. ISSN 1573-0565. doi: 10.1007/s10994-017-5684-y. URL https://doi.org/10.1007/s10994-017-5684-y.

Maciej Wołczyk, Michał Zájc, Razvan Pascanu, Łukasz Kuciński, and Piotr Miłoś. Continual World: A Robotic Benchmark For Continual Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 28496–28510. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/ef8446f35513a8d6aa2308357a268a7e-Abstract.html.

Tongtong Wu, Massimo Caccia, Zhuang Li, Yuan-Fang Li, Guilin Qi, and Gholamreza Haffari. Pretrained Language Model in Continual Learning: A Comparative Study. In *International Conference on Learning Representations*, October 2021. URL https://openreview.net/forum?id=figzpGMrdD.

Fan Yang, Zhilin Yang, and William W Cohen. Differentiable Learning of Logical Rules for Knowledge Base Reasoning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/0e55666a4ad822e0e34299df3591d979-Abstract.html.

Da-Wei Zhou, Hai-Long Sun, Jingyi Ning, Han-Jia Ye, and De-Chuan Zhan. Continual Learning with Pre-Trained Models: A Survey, April 2024. URL http://arxiv.org/abs/2401.16386. arXiv:2401.16386 [cs].

Shang Zhu, Bharath Ramsundar, Emil Annevelink, Hongyi Lin, Adarsh Dave, Pin-Wen Guan, Kevin Gering, and Venkatasubramanian Viswanathan. Differentiable modeling and optimization of non-aqueous Li-based battery electrolyte solutions using geometric deep learning. *Nature Communications*, 15(1):8649, October 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-51653-7. URL https://doi.org/10.1038/s41467-024-51653-7.

A Appendix

A.1 Multiobjective multifidelity exploration of the Pareto front

Daulton et al. (2023) deploy a hypervolume knowledge gradient (HV-KG) multi-objective optimization framework. As proposed in their study, we apply our online paradigm to the classical Branin-Currin synthetic multi-objective multi-fidelity (MOMF) optimization problem. This synthetic problem is characterized by the joint optimization of the Branin

$$B(\boldsymbol{x}) = a(x_2 - bx_1^2 + cx_1 - r)^2 + p(1-t)\cos(x_1) + p, \tag{2}$$

and Currin

$$C(\boldsymbol{x}) = \left[1 - \exp\left(-\frac{1}{2x_2}\right)\right] \frac{2300x_1^3 + 1900x_1^2 + 2092x_1 + 60}{100x_1^3 + 500x_1^2 + 4x_1 + 20} \tag{3}$$

objective functions, where $x_{11} = 15(x_1) - 5$, $x_{22} = 15x_2$, $a = 1$, $b(s) = 5.1/(4\pi^2) - 0.01(1-s)$, $c(s) = 5/\pi - 0.1(1-s)$, $r = 6$, $p = 10$, and $t(s) = (1/(8\pi)) + 0.05(1-s)$ with fidelity parameter $s \in \mathbb{R}$ (Irshad et al., 2024).

Algorithm 1 Sensitivity-Constrained MOMF optimization

```
Input: Initial computational budget $C_{\text{initial}}$, maximum evaluation budget $C_{\text{max}}$
Initialize buffer $B$ with samples evaluated under cost $C_{\text{initial}}$
Pre-train surrogate models on buffer $B$ *
while $C \leq C_{\text{max}}$ do
    if not 1st iter then
        Train surrogate models on buffer $B$ *
    end if
    Acquire new candidate points via acquisition function
    Evaluate candidates at chosen fidelities and append to buffer $B$
    Add cost for new candidates to cumulative cost
end while
Train final surrogate models on buffer $B$
Compute optimal Pareto frontier
```

* Denotes procedures for online learning with SC.

A.2 Hybrid training of coupling neural networks and process-based equations

We used the physical models to generate training buffers for the surrogates including dynamic input data with dimensions *batchsize* × *timesteps* × *number of dynamic variables*, static attributes with dimensions *batchsize* × *number of static variables*, physical model parameters with dimensions *batchsize* × *timesteps* × *number of parameters*, corresponding physical model outputs with dimensions *batchsize* × *timesteps* × *number of outputs*, and Jacobian sensitivities with dimensions *number of outputs* × *batchsize* × *timesteps* × *number of parameters*.

Hydrologic model: For hydrologic (streamflow) applications, we use the differentiable model dHBV first introduced by Feng et al. (2022)) which couples an LSTM NN with the conceptual hydrologic model Hydrologiska Byråns Vattenbalansavdelning (HBV). HBV is a collection of differentiable bucket-based physical processes that are parameterized by 12 time-invariant, spatially-distinct parameters learned by the NN.

Following the standard of approach for this model demonstrated by Feng et al. (2022), we use the open-access Catchment Attributes and Meteorology for Large-sample Studies - Dataset (CAMELS) produced by Newman et al. (2022), which provides access to all dynamic inputs, attributes, and the target streamflow observations to use as ground truth. This dataset includes 34 years of data at a selection of 671 USGS gages around the continental US. Keeping with the standard benchmarks for this model in our experiments, we select a subset of 531 of these gages and and use train data from 1 October 1999 to 30 September 2008, and validation data 1 October 1989 to 30 September 1999.

For this model, we chose to implement an LSTM surrogate for its ability to handle the highly dynamic nature streamflow data, which can oscillate between high and low flow extremes in response

Algorithm 2 Sensitivity-Constrained Training for ML-physics hybrid models

```
Input: Initial buffer $B$, maximum epochs $E$
Initialize buffer $B$ with batched physical model simulations
Pre-train surrogate model on buffer $B$ to resemble the physical model outputs *
for epoch = 1 to $E$ do
    Train NNs using data loss between surrogate model simulations and ground-truth observations
    Perform full model evaluation on random batches using the physical model
    Update buffer $B$ with newly generated samples *
    Update surrogate model by backpropagating from combined loss:
        $L_{\text{combined}} = L_{\text{data}} + \lambda L_{\text{SC}}$ (SC loss for selected parameters) *
end for
Evaluate full model using physical model without surrogate
```

* Denotes procedures for online learning with SC.

to the influx of water during precipitation events. We made the initial training buffer for this surrogate equivalent in spatio-temporal scope of training data as seen by the differentiable model's parameterization LSTM during training (531 locations and 3288 timesteps, as mentioned above). In particular the buffer contains 194 minibatches of data that are each seen once per training epoch. These minibatches include the dynamic drivers with shape $[100, 730, 3]$, static attributes with shape $[100, 35]$, parameters with shape $[100, 730, 192]$, and streamflow model output and ground truth, both with shape $[100, 730, 1]$. Jacobian sensitivities of the output to the input parameters are also stored with shape $[1600, 2, 4]$, where we stored the four most sensitive parameters for the middle and last timesteps of each minibatch optimization window. We try generating parameters for the buffer with both Latin Hypercube Sampling (LHS) to uniformly sample the entire parameter space, as well as the parameterization LSTM itself (trained for 1 epoch).

Table 3: Descriptions for HBV model parameters.

Parameter	**Description**
β	Recharge non-linearity (-)
FC	Field capacity (mm)
K_0	Runoff coefficient (day^{-1})
K_1	Subsurface flow coefficient (day^{-1})
K_2	Groundwater storage coefficient (day^{-1})
LP	Wilting point as fraction of FC (-)
c	Percolation rate (mm·day^{-1})
UZL	Upper zone storage limit (mm)
TT	Threshold temperature for snowfall (°C)
CFMAX	Degree-day factor (mm · °C^{-1} · day^{-1})
CFR	Refreezing coefficient (-)
CWH	Water holding capacity as a fraction of current snowpack (-)

Ecosystem model: We used the developed differentiable ecosystem model (δ_{psn}) described in Aboelyazeed et al. (2025) whose physical component is based on the photosynthesis and plant hydraulic modules originally derived from the Functionally Assembled Terrestrial Ecosystem Simulator (FATES Development Team, 2020). The parameter calibration module is composed of three multi-layer perceptron NNs which map static and dynamic inputs to 7 physical parameters required by the physical component in δ_{psn} (see Table 4). Among these parameters, $V_{c,max25}$ is dynamic, while the rest of the parameters are static.

For this model, we chose to implement an FNO surrogate due to its high computational efficiency which is highly required for such expensive physical model. For the initial buffer, we ran the differentiable framework (physical model + parameterization NNs) for one epoch, during which the parameterizations NNs were trained concurrently with data generation. The initial buffer is composed of 35 batches, each with a batch size of 100 trees. Each batch includes: (i) dynamic inputs of size [100, 360, 14], (ii) static attributes [100, 4], (iii) physical parameters [100, 360, 70], (iv) sap flow simulations [100, 360,1], and (iv) sensitivity gradients [100, 360,7]. Here 100 refers to the batch size, 360 to the number of timesteps, 14 to the number of dynamic inputs, 4 to the static attributes, 70 to the physical parameters (of which 7 are learnt by NNs). For the sensitivity gradients, we computed the derivative of the sap flow simulation at the last timestep with respect to the learnable parameters.

For observation data, we used Tree-level sap flow observations from SAPFLUXNET database (Poyatos et al., 2021), which compiles hourly records from sites around the world. For each site, we selected three-month periods during growing season at hourly resolution, using two months for training and one month for testing. Our selected dataset includes observations from more than 100 sites and covers more than 120 species across different plant types from only overstory trees that receive full-sunlight. Beside sap flow observations, we used weather data provided by SAPLUXNET (i.e., temperature, radiation, wind speed, relative humidity), and tree-specific attributes such as tree height, diameter at breast height, and sapwood area. We further incorporated hourly soil moisture from ERA5 Reanalysis dataset (Muñoz Sabater, 2019), as well as soil texture data (sand percentage, clay percentage, organic matter) from soilGrids-ISRIC (Poggio et al., 2021) at multiple depths. Species-specific traits such as wood density and specific leaf area were obtained from TRY database (Kattge, 2020), while hydraulic traits were computed as described in (Christoffersen et al., 2016).

Table 4: Descriptions for Ecosystem model parameters.

Parameter	**Description**
$V_{c,\max25}$	Maximum carboxylation rate at 25 °C
$P_{50,\mathrm{gs}}$	Water potential at 50% loss of conductivity for stomata
a_{gs}	Shape parameter for stomatal vulnerability
g_1	Medlyn slope parameter for Medlyn stomatal conductance model (Medlyn et al., 2011)
$K_{s,\max,x}$	Maximum xylem conductivity per unit sapwood area
Leaf scaler coeff$_1$	Coefficient one for decrease in leaf Nitrogen through the canopy (Lloyd et al., 2010)
Leaf scaler coeff$_2$	Coefficient two for decrease in leaf through the canopy (Lloyd et al., 2010)

Table 5: Average runtime per epoch for the differentiable ecosystem model. All configurations were trained for 50 epochs. For online training with sensitivity constraint (SC), we evaluate performance for continued surrogate training every n_{SC} =1, 2, 5, and 10 epochs. The numbers in the parentheses is n_{SC}. Reported times in seconds represent the full simulation time divided by the total number of epochs. Because OCL finetunes the surrogate for each epoch, it can be more expensive than OCL+SC(5) and OCL+SC(10) even if it does not compute the Jacobians. At $n_{SC} = 1$, the cost of computing Jacobians and training on SC is calculated as $180s - 143s = 37s$ per epoch, or 26% of the time of OCL alone.

	Runtime per Epoch (s)
Physical Model(Benchmark)	4176
Offline	17
OCL	143
Offline + SC	21
OCL + SC (1)	180
OCL + SC (2)	144
OCL + SC (5)	120
OCL + SC (10)	107

A.3 FUSE (JOINT GENERATIVE INVERSION-SIMULATION TRAINING)

Algorithm 3 Sensitivity-Constrained FUSE

```
Input: Initial buffer $B$, maximum epochs $E$, warm-up epochs $E_w$
Initialize buffer $B$ with batched data
for epoch = $1$ to $E$ do
    Train FMPE inverse model using $\mathbf{L}_{\text{FMPE}}$
    Train FNO forward model using
      $L = L_{\text{data}}$ (data loss only) or $L = L_{\text{data}} + \lambda L_{\text{SC}}$ (with sensitivity-constrained loss) *
    if epoch $\geq E_w$ then
        Generate additional training samples using FNO surrogate *
        Update buffer $B$ with generated samples *
    end if
end for
Evaluate:
    (a) Combined SC-FUSE model (FMPE + FNO) for full-inversion prediction task, or *
    (b) FNO forward model using ground-truth parameters
```

* Denotes procedures for online learning with SC.