

|  |  |  |
| --- | --- | --- |
| PNNL-XXXXX |  | |
|  | Keeping Deep-Learning Surrogates Accurate for Long-Term Groundwater Remediation: Concept & Prototype Plan  Enter Subtitle Here (or delete)  Publish Date (Month Year)  1First M Last  2First M Last 3First M Last 4First M Last | |
|  | Text  AI-generated content may be incorrect. |  |
| Prepared for the U.S. Department of Energy  under Contract DE-AC05-76RL01830 |

**DISCLAIMER**

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor Battelle Memorial Institute, nor any of their employees, makes **any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights**. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or Battelle Memorial Institute. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

PACIFIC NORTHWEST NATIONAL LABORATORY

*operated by*

BATTELLE

*for the*

UNITED STATES DEPARTMENT OF ENERGY

*under Contract DE-AC05-76RL01830*

**Printed in the United States of America**

**Available to DOE and DOE contractors from**

**the Office of Scientific and Technical Information,**

**P.O. Box 62, Oak Ridge, TN 37831-0062**

[**www.osti.gov**](http://www.osti.gov)

**ph: (865) 576-8401**

**fox: (865) 576-5728**

**email:** [**reports@osti.gov**](mailto:reports@osti.gov)

**Available to the public from the National Technical Information Service**

**5301 Shawnee Rd., Alexandria, VA 22312**

**ph: (800) 553-NTIS (6847)**

**or (703) 605-6000**

**email:** [**info@ntis.gov**](mailto:info@ntis.gov)

**Online ordering:** [**http://www.ntis.gov**](http://www.ntis.gov)

Keeping Deep-Learning Surrogates Accurate for Long-Term Groundwater Remediation: Concept & Prototype Plan

Enter Subtitle Here (or delete)

Publish Date (Month Year)

1First M Last   
2First M Last  
3First M Last  
4First M Last

Prepared for  
the U.S. Department of Energy  
under Contract DE‑AC05‑76RL01830

Pacific Northwest National Laboratory

Richland, Washington 99354

Abstract

Use Body Text for paragraphs in this section. PNNL reports use <http://www.chicagomanualofstyle.org/home.html> for document style. Right-click and choose open hyperlink to view the style guide.

Summary

Groundwater remediation at legacy waste sites demands fast, reliable long-term predictions of contaminant behavior. Physics-based simulators (MODFLOW, TOUGH2, STOMP, PFLOTRAN, et al.) provide high-fidelity forecasts but are too slow for real-time decision support. Deep learning (DL) surrogates—trained on prior simulation output—can predict very rapidly, but lose accuracy over time as site conditions evolve. This project develops a hybrid recalibration framework that combines physics and ML: it continuously monitors surrogate uncertainty and triggers targeted physics runs to update the DL model only when needed. In essence, the deep-learning networks serve as "sophisticated surrogates" of the true physics model, and the framework automates "smart surrogate updating" so that predictions remain accurate without retraining from scratch.

The framework is organized around three core pillars that interact in a feedback loop. A Conceptual Governance pillar sets objectives, success metrics and regulatory constraints while defining essential concepts like uncertainty quantification, active learning triggers, and continual learning strategies. A Modeling Engines pillar includes the physics-based simulator (the "truth" model) and the DL surrogate (e.g. U-Net, LSTM or diffusion networks) for fast inference. A Recalibration Mechanisms pillar continuously monitors surrogate performance via uncertainty quantification and initiates active learning updates (e.g. additional simulator runs and/or incremental model fine-tuning) when new data or conditions lie outside the original training domain. By integrating these pillars in a dynamic feedback loop, the project balances physical consistency, computational efficiency, and adaptability.

This approach transforms deep learning models from static "interpolation tools" into adaptive, physics-aware forecasting systems capable of maintaining reliability across multi-year operational periods while providing the speed necessary for responsive remediation management. The framework enables near-real-time groundwater scenario analysis with *10³–10⁴×* speed improvements over direct physics simulation while maintaining prediction accuracy within acceptable thresholds through systematic recalibration. By coupling uncertainty quantification with active learning principles, the system focuses expensive physics-based simulations only on high-uncertainty conditions where model updates provide maximum value, potentially reducing computational costs by xx-xx% over traditional approaches. (up to change, as we may not active training)

Some implication (add later)

Acknowledgments

Use Body Text for paragraphs in this section. PNNL reports use <http://www.chicagomanualofstyle.org/home.html> for document style. Right-click and choose open hyperlink to view the style guide.

Acronyms and Abbreviations

Use Acronyms Word style for paragraphs in this section. Acronym use in PNNL reports follows <http://www.chicagomanualofstyle.org/book/ed17/part2/ch10/toc.html>. Right-click and choose open hyperlink to view the style guide.

Contents

[Abstract ii](#_Toc220495381)

[Summary iii](#_Toc220495382)

[Acknowledgments iv](#_Toc220495383)

[Acronyms and Abbreviations v](#_Toc220495384)

[1.0 Introduction 10](#_Toc220495385)

[2.0 Framework Overview (Xuehang et. all) 12](#_Toc220495386)

[2.1 Three Pillars Structure 12](#_Toc220495387)

[2.2 Seven Steps Operational Workflow 13](#_Toc220495388)

[3.0 Methods 16](#_Toc220495389)

[3.1 Governance Essentials (Xuehang) 16](#_Toc220495390)

[3.1.1 Regulatory Alignment and Quality Assurance Foundations 16](#_Toc220495391)

[3.1.2 Model Usage Scope and Decision Context 17](#_Toc220495392)

[3.1.3 Acceptance Criteria and Recalibration Triggers 17](#_Toc220495393)

[3.1.4 Operational Monitoring and Review Schedule 18](#_Toc220495394)

[3.1.5 Transparency, Auditability, and Configuration Control 18](#_Toc220495395)

[3.1.6 Summary 20](#_Toc220495396)

[3.2 Modeling Engines (Ross, Xuehang, Yilin) 21](#_Toc220495397)

[3.2.1 Site Challenges and Modeling Needs 21](#_Toc220495398)

[3.2.2 Physics-Based Numerical Models 22](#_Toc220495399)

[3.2.3 Surrogate Model Landscape 23](#_Toc220495400)

[3.2.4 General Surrogate Model Training Setup (Xuehang, Ross, TC): 31](#_Toc220495401)

[3.3 Recalibration Mechanisms (TC, Xuehang, Jason) 32](#_Toc220495403)

[3.3.1 Uncertainty Quantification of Surrogate Models (TC & Jason) 32](#_Toc220495404)

[3.3.2 Trigger logic (TC & Jason) 34](#_Toc220495405)

[ Tipping Point–Inspired Resilience Metrics 34](#_Toc220495406)

[ Performance and Uncertainty-Based Metrics 35](#_Toc220495407)

[ Distributional and Representation-Level Drift Detection 35](#_Toc220495408)

[ Probabilistic Integration and Decision Logic 35](#_Toc220495409)

[3.3.3 Continual Learning (Xuehang) 38](#_Toc220495410)

[3.3.4 Active Learning (Xuehang) 39](#_Toc220495411)

[3.3.5 Multi-fidelity Data Fusion (Jason, Xuehang): 40](#_Toc220495412)

[4.0 Validation & Metrics (should we move this to 3.0?) 47](#_Toc220495416)

[5.0 Software Architecture 48](#_Toc220495417)

[5.1 Software Requirements 48](#_Toc220495418)

[5.2 System Architecture 49](#_Toc220495419)

[5.2.1 Independent Software Packages and the M&R Software Package 50](#_Toc220495420)

[5.2.2 On-Line Operational System Architecture 53](#_Toc220495421)

[5.2.3 Basic User Interface 55](#_Toc220495422)

[5.3 Software Development Guidelines 55](#_Toc220495423)

[5.3.1 Version Control 55](#_Toc220495424)

[5.3.2 Testing and CI/CD 55](#_Toc220495425)

[5.4 Not Considered 56](#_Toc220495426)

[5.4.1 Framework 56](#_Toc220495427)

[5.4.2 Software Maintenance 58](#_Toc220495428)

[5.4.3 System Architecture 59](#_Toc220495429)

[6.0 Prototype & Schedule 61](#_Toc220495431)

[6.1 U‑FNO Surrogate for 2D cross‑section Transport Prototype 61](#_Toc220495432)

[6.1.1 PFLOTRAN Baseline Model 61](#_Toc220495433)

[6.1.2 U‑FNO Surrogate and Training 63](#_Toc220495434)

[6.2 U‑Net Architecture for 2D Pump‑and‑Treat Plume Prediction 64](#_Toc220495435)

[6.2.1 2D Training Data Generation 65](#_Toc220495436)

[6.2.2 Architectural Modifications for Improving Spatial Pattern Reconstruction 65](#_Toc220495437)

[6.2.3 Training Workflow 66](#_Toc220495438)

[7.0 Roadmap and Impact (Xuehang) 67](#_Toc220495439)

[8.0 References 68](#_Toc220495440)

[9.0 Appendix 74](#_Toc220495441)

[Concepts for High-TRL Modeling and Calibration User-Facing Software Interface 170](#_Toc220495523)

Figures

[**Figure 1.** *The* article publication growth in surrogate modeling for groundwater and subsurface transport, based on a Web of Science topic search (2005-2025). We observe a modest number of surrogate studies prior to 2010, followed by a rapid increase after ~2018 as deep learning and operator learning architectures enter the field. 24](#_Toc219733275)

[**Figure 2*.*** Method evolution in a screened surrogate modeling publication record. Stacked counts of representative groundwater or subsurface surrogate modeling papers by publication year and primary surrogate family. Papers published from 1994 to 2016 are aggregated into a single bin because the curated set in that period is dominated by regression and classical approaches. Classification is based on the dominant surrogate architecture used for the main modeling task (one primary label per paper), and the figure is intended as a representative method overview that complements the full-corpus publication trend shown in ***Figure* 1**. 25](#_Toc219733276)

[**Figure 3.** Conceptual mind-map of surrogate model families in Pillar B, organized by input*-*output structure and typical EM use*-*cases. Synthesized from published reviews of groundwater surrogates (Luo et al., 2023).*Abbreviations: enc-dec: encoder-decoder, vars: variables, BCs: boundary conditions, ICs: initial conditions.* 25](#_Toc219733277)

[**Figure 4.** Heat map summary of the selected, application focused surrogate modeling corpus. Top: counts of papers by surrogate method family versus application. Bottom: counts of papers by surrogate method family versus project domain. Cell values denote the number of papers assigned to each pairing. Each paper is assigned a single primary label per axis to prevent double counting. 30](#_Toc219733278)

[Figure : Modeling control panel and landing screen. Buttons transition to dashboards and deeper configuration applications. 170](#_Toc219733279)

[Figure : Dashboard for interactive modeling metrics and modeling quality view. Metrics can provide coloring based on their state (good, warning, requires recalibration) 170](#_Toc219733280)

[Figure : Dashboard for viewing sensor measurement histories. 171](#_Toc219733281)

[Figure : Dashboard for modeling comparisons. 171](#_Toc219733282)

[Figure : Control Flow Diagram for Waterlevel Monitoring and Recalibration System. User interacts with the GUI interface (A). The GUI acts as a visual representation for the state of the system and provides the ability to interact with the system (B). The monitoring and recalibration software (by schedule or manual initiation) upload new measurements from the client to the operational database (C), orchestrate processing in a generic HPC connection for model training and evaluation(E), then upload those results to the monitoring database (D). tA high level diagram of the monitoring and calibration software framework is provided in Figure 1. The primary consideration here is that all boxed components must function as isolated services, independent of the activity or availability of the other services. This means that errors or failure in one subsystem does not affect the operation of other subsystems. It also means that each of these subsystems can be generically hosted on local and online resources, allowing for dynamic configuration of the system. 59](#_Toc219733283)

Tables

**No table of figures entries found.**

# Introduction

Large DOE waste sites often require decades of groundwater monitoring and remediation demanding long-term forecasts of contaminant plume evolution to adapt remedies (e.g. pump-and-treat systems or reactive barriers) over time. Physics-based flow-and-transport models can simulate plume dynamics accurately, but each high-fidelity run may take hours to days. In contrast, data-driven deep learning (DL) surrogates trained on past simulations can generate new predictions in seconds.

However, these environmental systems are inherently nonstationary: new wells and data streams could be added, remediation actions alter subsurface conditions, and climate variability changes recharge and boundary inputs. In practice, a model’s ability to generalize relies on the assumption of stationarity, which “in real-world environments is often violated” as the system “is dynamically changing all the time”. In other words, a static surrogate will drift from reality unless it adapts to such changes.

Key challenges identified in literature include:

* Concept/Distribution Drift: When site conditions or forcing change, a DL model trained on historical data can quickly become invalid. For example, a surrogate calibrated under one pumping regime may mis-predict if a new extraction well alters the flow field. Concept-drift theory shows that when the statistical properties of the domain change, model performance degrades over time. Without ongoing recalibration or adaptation, predictive error can increase abruptly once operations or site condition shift.
* Data and Compute Scarcity: Data and Compute Scarcity: High-fidelity reactive transport simulations (e.g. multi-species, multi-process models) are computationally intensive. so fresh training data come only in small, costly batches. Deep models also tend to be data-hungry; studies found that a convolutional surrogate needed on the order of thousands or tens of thousands training samples to match the accuracy of a detailed physical model. In practice, field campaigns and simulations can only produce limited new data, ruling out brute-force retraining on large data.
* Physical Realism and Trust: Stakeholders and regulators demand that forecasts remain physically plausible and transparent. Surrogate models omit some physics, so including prediction uncertainty bounds is essential for trust.

These limitations mean that traditional modeling approaches fall short for long-term forecasting. Periodic retraining “from scratch” is time-consuming and ignores efficient reuse of prior knowledge. Plain DL models (without correction) simply interpolate within their original data envelope, leading to large errors under novel conditions. For instance, Liu et al. (2022) found that deep networks for regional groundwater predict well only when data are dense and continuous; performance degrades sharply when records are sparse. Similarly, studies of source-zone plumes (Yang et al. 2018) show that long-term plume behavior is controlled by complex processes (dissolution, sorption, back diffusion) that can produce heavy “long tail” persistence – conditions that a static surrogate may never have seen. To overcome these challenges, researchers advocate continual learning, multi-fidelity methods and rigorous uncertainty quantification. For example, Kontos et al. (2022) demonstrated transferring DL models from synthetic to field regimes via triggered updates: they pretrained on large ensembles and then fine-tuned only when residuals or uncertainty passed thresholds. In short, the literature agrees that maintaining long-term accuracy requires adaptive strategies – not just one-time training. (XS: save my literature reviewer for future adoption/update)

To overcome these challenges, recent research advocates adaptive strategies. Continual learning and multi-fidelity methods allow the surrogate to update intelligently as new data arrive, while uncertainty quantification tools flag when predictions become unreliable. For instance, one approach is to pretrain a neural surrogate on large synthetic ensembles and then trigger fine-tuning only when model residuals or uncertainty exceed a threshold. Concept-drift literature similarly recommends adaptive learning pipelines that detect shifts and retrain a model on-demand. In short, experts agree that maintaining long-term accuracy in subsurface forecasting requires moving beyond one-time training. Our work is motivated by this need: we develop an adaptive deep-learning surrogate framework that continually incorporates new observations and physics-based insight, with the goal of preserving predictive accuracy and physical consistency as site conditions evolve.

# Framework Overview (Xuehang et. all)

## Three Pillars Structure

The proposed recalibration framework addresses these gaps with a layered, three-pillar architecture. Its pillars are:

* Pillar A: Conceptual Design & Governance. This pillar defines the project’s objectives, success metrics, and regulatory requirements. It establishes trigger criteria for when to update models (e.g. threshold on predictive uncertainty or prediction–observation residual). It also include data policies (which new measurements to assimilate) and interpretability standards. In short, this layer ensures the framework aligns with DOE mission goals and compliance needs, guiding how data flows and decisions are governed.
* Pillar B: Modeling Engines. This core pillar contains the simulation and surrogate models. It includes the high-fidelity physics-based simulator (the “truth” source) and the DL surrogate architectures. The physics simulator is used offline to generate training scenarios (e.g. Monte Carlo ensembles of hydrogeologic variability). The DL module (e.g. a U-Net convolutional model or LSTM network) is trained on this synthetic data to capture complex spatiotemporal plume dynamics. By learning the input–output mapping of the simulator, the surrogate can predict contaminant concentrations orders of magnitude faster than real-time computation. Initially, both models are validated against benchmarks to ensure baseline accuracy.
* Pillar C: Recalibration Mechanisms. This pillar performs ongoing model maintenance. It embeds uncertainty quantification (UQ) and active learning workflows that watch for surrogate failure. Operationally, the system continuously compares surrogate predictions with monitoring data and calculates uncertainty (using techniques like Monte Carlo dropout or deep ensembles). When uncertainty or residuals exceed the conceptual thresholds (e.g. indicating extrapolation beyond the training regime), the framework launches targeted physics simulations to gather new data. These new high-fidelity runs are used to fine-tune the surrogate via continual learning methods (e.g. Elastic Weight Consolidation or replaying a small buffer of past cases). This incremental update avoids full retraining, preserving previous knowledge while adapting to change. The updated model’s uncertainty is re-calibrated (for example by adjusting probability outputs or using Bayesian layers), and predictions are re-deployed for decision support.

Together, these pillars form an automated feedback loop. **Figure 1** illustrates the high-level architecture: the Modeling Engines produce forecasts; the Recalibration Mechanisms monitor errors and uncertainty; and the Governance pillar adjusts policies or retraining triggers as needed.

Diagram

AI-generated content may be incorrect.

Crucial ML techniques are embedded throughout. Active learning ensures new simulation runs are targeted to the most informative conditions (e.g. sampling at decision boundaries or rare scenarios), thereby reducing the number of expensive model runs. Continual learning methods (such as layer freezing or regularization like EWC) allow the surrogate to ingest new data without catastrophic forgetting. Uncertainty quantification (e.g. MC-dropout, Bayesian nets) provides the diagnostic “self-awareness” needed to know *when* recalibration is needed. Finally, multi-fidelity data integration fuses diverse sources: routine monitoring observations, remote sensing data, and low-fidelity models are assimilated alongside high-fidelity simulation results to refine predictions.

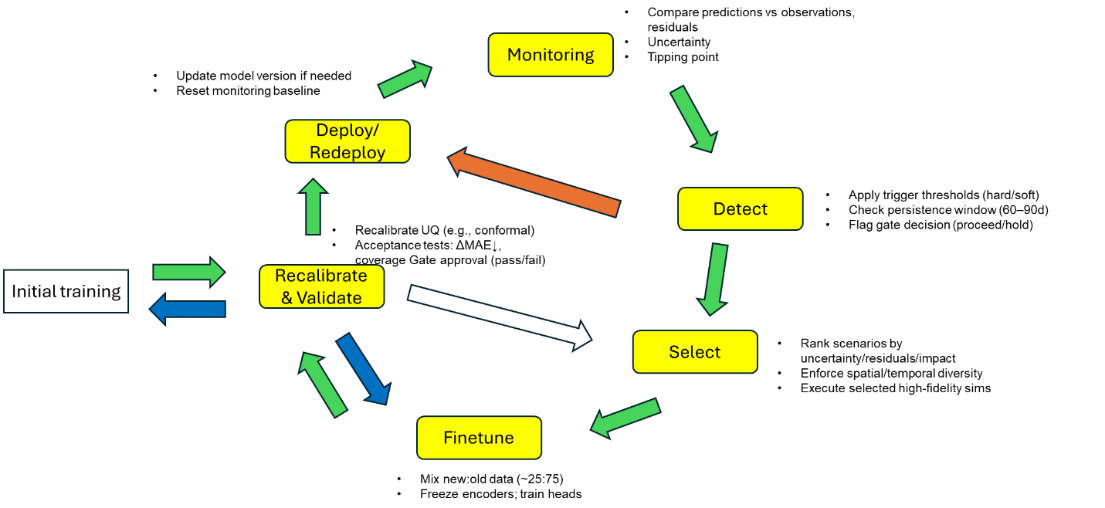
In summary, the framework overcomes the identified limitations by combining physics-based truth with smart ML adaptation. It systematically updates the surrogate only as needed, thus maintaining accuracy without incurring the cost of full retraining. By leveraging insights from recent studies (Yang et al. 2018; Arshadi et al. 2019; Kontos et al. 2022; Liu et al. 2022; Goebel et al. 2019, etc.), the plan employs best practices in active sampling, transfer learning and uncertainty-aware learning. The result is a continuous, adaptive management framework that delivers fast and trustworthy long-term forecasts for groundwater remediation decisions.

## ****Seven Steps Operational Workflow****

The framework operates through a continuous seven-step recalibration cycle that automatically maintains surrogate accuracy:

* Step 1: Monitor - The system continuously tracks surrogate predictions against available monitoring data, calculating prediction-observation residuals and uncertainty metrics using techniques like Monte Carlo dropout or deep ensembles.
* Step 2: Detect - Automated diagnostics identify when the surrogate begins to fail by comparing uncertainty levels and residuals against predefined thresholds established in Pillar A. Detection triggers include: (a) predictive uncertainty exceeding confidence bounds, (b) systematic bias in prediction-observation residuals, or (c) extrapolation beyond the original training domain.
* Step 3: Select (optional) - Active learning algorithms identify the most informative new simulation scenarios to run. Rather than random sampling, the system targets conditions at decision boundaries, rare scenarios, or regions of high uncertainty where new data will maximally improve model performance.
* Step 4: Targeted physics simulations (optional)
* Step 5: Fine-tune - Continual learning methods update the DL surrogate using the new simulation data. Techniques like Elastic Weight Consolidation (EWC) or replay buffers prevent catastrophic forgetting, allowing the model to learn new patterns while preserving previously acquired knowledge.
* Step 6: UQ - The updated model's uncertainty estimates are recalibrated to ensure predicted confidence intervals accurately reflect true prediction reliability. This may involve adjusting probability outputs, updating Bayesian layers, or retraining uncertainty quantification components
* Step 7: Redeploy - The improved surrogate is deployed with comprehensive audit trails documenting what triggered the update, which data was used, and how performance changed. All recalibration events are logged for regulatory transparency and quality assurance.

This cycle repeats continuously during site operations, with the frequency of updates determined by the rate of change in site conditions and the availability of new monitoring data. The automated nature ensures that surrogates remain accurate without requiring manual intervention or complete retraining.



**Timeline

AI-generated content may be incorrect.**

# Methods

This section defines how we turn physics runs into training data, how we design and train the surrogate, and how we decide that the surrogate is accurate enough for decision support. Detailed numbers and paper‑by‑paper comparisons appear in the summary table at the end of the draft.

## Governance Essentials (Xuehang)

This section defines the governance framework that controls how deep-learning surrogate models are developed, evaluated, monitored, and updated for long-term groundwater remediation across U.S. DOE-EM sites. The purpose of this governance layer is to ensure that surrogate modeling is scientifically defensible, operationally stable, and aligned with existing regulatory and quality-assurance expectations, while enabling efficient scenario analysis and adaptive planning over multi-year remediation timelines.

The governance framework is designed to be applicable across all DOE EM programs and sites, and is grounded in established regulatory and quality standards, including EPA Comprehensive Environmental Response, Compensation, and Liability Act (CERCLA) guidance, DOE quality assurance orders, NQA-1 software quality requirements, and EPA modeling and data quality manuals. Within this framework, surrogate models are treated as decision-support tools that primarily support screening-level analyses, planning, and scenario exploration, while high-fidelity physics-based numerical models remain the authoritative basis for final regulatory and remedy decisions.

### Regulatory Alignment and Quality Assurance Foundations

Governance of the surrogate modeling framework is anchored in existing federal environmental modeling and quality-assurance guidance. Under CERCLA, remedial decisions must remain protective of human health and the environment over long operational horizons, and analytical tools used to support these decisions must be transparent, defensible, and periodically evaluated. (U.S. Environmental Protection Agency, A Guide to Preparing Superfund Five-Year Reviews, EPA 540-R-01-007, https://www.epa.gov/laws-regulations/summary-comprehensive-environmental-response-compensation-and-liability-act)

Consistent with CERCLA expectations, this framework adopts the principles articulated in the EPA’s guidance on environmental modeling, which emphasizes clear documentation of model purpose, assumptions, uncertainty, validation, and limitations before models are relied upon for decision support (U.S. Environmental Protection Agency, Guidance on the Development, Evaluation, and Application of Environmental Models, EPA/100/K-09/003, https://www.epa.gov/measurements-modeling/guidance-development-evaluation-and-application-environmental-models). These principles directly inform the governance requirements for surrogate use, update triggers, and performance acceptance.

Governance for surrogate modeling activities need to be align with DOE O 414.1D Chg 2 (LtdChg), Quality Assurance, which establishes a management system requiring defined acceptance criteria, performance monitoring, corrective action, and continuous improvement for analytical tools used in mission-critical applications (U.S. Department of Energy, DOE O 414.1D Chg 2 (LtdChg), Quality Assurance, https://www.directives.doe.gov/directives-documents/400-series/0414.1-BOrder-d-chg2-ltdchg). The surrogate modeling framework is further governed under DOE EM-QA-001 (https://emcbc.doe.gov/SEB/Files/WVDPPhase1B/DocLib/EM-QA-001%20R2%20FINAL%20(04-10-19).pdf), which incorporates ASME NQA-1 standards and applies them to environmental management activities, including software and modeling systems.

Together, these regulatory and quality standards establish the expectation that surrogate models must be managed as controlled analytical software, with formal verification, validation, documentation, and configuration management processes that support regulatory confidence and long-term stewardship.

### Model Usage Scope and Decision Context

A central governance principle is the explicit definition of how surrogate models may be used within DOE EM decision workflows. Within this framework, surrogate models are intended primarily to support screening-level analyses, such as rapid evaluation of alternative remediation scenarios, sensitivity analyses, uncertainty exploration, and preliminary planning studies. In these contexts, surrogates provide substantial value by enabling fast exploration of large scenario spaces that would be impractical using high-fidelity numerical models alone.

For formal regulatory or final remedy decisions, surrogate model outputs are not used as stand-alone evidence. Instead, surrogate results may be used to inform and prioritize analyses, guide the selection of scenarios for further evaluation, or support interpretation of system behavior, while final decisions remain grounded in high-fidelity physics-based modeling and empirical data. This distinction aligns with EPA remedial investigation and feasibility study guidance, which emphasizes that models must be used in a manner consistent with their purpose and limitations, and that higher-consequence decisions require correspondingly higher levels of evidentiary support. (U.S. Environmental Protection Agency, Remedial Investigation/Feasibility Study Guidance, EPA 540-G-89-004, this is pretty old report though)

Governance therefore ensures that surrogate models function as advisory, efficiency-enhancing tools, rather than replacements for established numerical simulators. Any application of surrogate outputs beyond screening-level support requires explicit documentation of assumptions, uncertainty, and corroboration with physics-based results, consistent with EPA modeling guidance and DOE quality expectations.

### Acceptance Criteria and Recalibration Triggers

The governance framework establishes predefined acceptance criteria and recalibration triggers that determine when surrogate models are considered reliable and when updates are warranted. These criteria are defined prior to deployment and are based on project-specific data quality objectives and decision needs, consistent with EPA quality assurance guidance for modeling applications (U.S. Environmental Protection Agency, Guidance for Quality Assurance Project Plans for Modeling, EPA QA/G-5M).

Acceptance criteria typically include:

* Quantitative performance metrics (e.g., error statistics at priority monitoring locations),
* Calibration of uncertainty estimates (e.g., empirical coverage of prediction intervals),
* Spatial and temporal consistency with observed plume behavior, and
* Stability of performance over time under routine operational conditions.

Recalibration triggers are defined to detect meaningful degradation in surrogate performance or applicability, such as sustained increases in prediction residuals, inflation or miscalibration of uncertainty estimates, or documented changes in boundary conditions or operational regimes that move the system outside the surrogate’s original training domain. These triggers are evaluated using objective, documented thresholds rather than ad hoc judgment, consistent with DOE quality assurance requirements for corrective action when performance expectations are not met. (U.S. Department of Energy, DOE O 414.1D Chg 2 (LtdChg).

Importantly, the presence of recalibration trigger does not automatically mandate recalibration. Instead, governance requires formal justification that an update is necessary, proportional, and expected to improve performance. This conservative approach aligns with best practices for model lifecycle management and reduces the risk of destabilizing otherwise reliable surrogate model behavior.

### Operational Monitoring and Review Schedule

The governance framework defines a structured but adaptable monitoring Schedule for surrogate model oversight. As a baseline, automated diagnostics are conducted on a regular basis to track surrogate performance, uncertainty behavior, and consistency with incoming monitoring data. These diagnostics support early identification of emerging issues without triggering unnecessary updates.

At a higher level, summary evaluations are conducted on a periodic basis (e.g., monthly or quarterly), synthesizing diagnostic results into management-level indicators of model performance, data quality, and operational relevance. Formal decision points for potential recalibration are typically evaluated on a quarterly basis, providing a predictable and auditable schedule for governance review.

This schedule is not fixed and may be adjusted to reflect site-specific monitoring frequencies, rates of system change, and regulatory reporting cycles. The guiding principle is that model review frequency should be commensurate with the pace of environmental and operational change, consistent with EPA and DOE expectations for adaptive environmental management (U.S. Environmental Protection Agency, Guidance on the Development, Evaluation, and Application of Environmental Models, EPA/100/K-09/003, https://www.epa.gov/measurements-modeling/guidance-development-evaluation-and-application-environmental-models).

### Transparency, Auditability, and Configuration Control

To support regulatory confidence and long-term stewardship, the governance framework requires full transparency and auditability for surrogate model deployment and updates. All surrogate model versions are subject to formal configuration control, including version tracking, documentation of training data and assumptions, and preservation of prior approved models.

Each recalibration or significant update produces a documented governance record that includes the trigger rationale, data sources used, evaluation against acceptance criteria, and verification results. These records are retained as part of the project’s quality assurance documentation and support independent review or audit if required, consistent with ASME NQA-1 software quality requirements (ASME, NQA-1: Quality Assurance Requirements for Nuclear Facility Applications, Subpart 2.7, Quality Assurance Requirements for Computer Software, https://www.asme.org/codes-standards/find-codes-standards/quality-assurance-requirements-for-nuclear-facility-applications?).

Data provenance is also explicitly governed. Monitoring data, numerical model outputs, and surrogate training datasets are tracked with metadata describing source, validation status, and applicability. This ensures that the influence of any dataset on surrogate behavior can be reconstructed, supporting traceability and corrective action if data quality concerns arise (U.S. Environmental Protection Agency, Guidance for Quality Assurance Project Plans for Modeling, EPA QA/G-5M).

Rollback is defined within this framework as the formal withdrawal of a surrogate model from operational use, rather than reversion to a previously approved surrogate version. Rollback is invoked when an attempted incremental update fails to meet predefined acceptance criteria, indicating that the surrogate’s training domain or structure is no longer scientifically valid.

Under a rollback condition, neither the updated surrogate nor the prior surrogate is permitted for continued screening or planning analyses, because the initiation of the update already reflects documented evidence that the prior surrogate has degraded or fallen outside its domain of applicability. Surrogate-based decision support is therefore explicitly suspended, and analytical support reverts exclusively to the high-fidelity physics-based numerical model, which remains the authoritative reference throughout the framework.

Rollback automatically triggers escalation to a governed full retraining process, in which a new surrogate baseline is developed from scratch using updated physics-model ensembles, revised assumptions, and refreshed validation datasets. Full retraining is conducted under configuration control and subject to the same verification, validation, uncertainty calibration, and acceptance criteria as an initial deployment. The retrained surrogate is promoted to operational use only after all governance requirements are satisfied.

This definition of rollback reflects DOE expectations for conservative software lifecycle management in high-consequence environmental applications. Incremental updates are preferred when scientifically defensible but continued use of any surrogate is not permitted once validity is in question. By redefining rollback as withdrawal and retraining rather than version reversion, the framework prevents unmanaged risk and preserves regulatory confidence in long-term remediation planning.

Reference:

U.S. Environmental Protection Agency, A Guide to Preparing Superfund Five-Year Reviews, EPA 540-R-01-007, <https://www.epa.gov/laws-regulations/summary-comprehensive-environmental-response-compensation-and-liability-act>

U.S. Environmental Protection Agency, Guidance on the Development, Evaluation, and Application of Environmental Models, EPA/100/K-09/003, https://www.epa.gov/measurements-modeling/guidance-development-evaluation-and-application-environmental-models

U.S. Department of Energy, DOE O 414.1D Chg 2 (LtdChg), Quality Assurance, <https://www.directives.doe.gov/directives-documents/400-series/0414.1-BOrder-d-chg2-ltdchg>

DOE EM-QA-001 <https://emcbc.doe.gov/SEB/Files/WVDPPhase1B/DocLib/EM-QA-001%20R2%20FINAL%20(04-10-19).pdf>)

U.S. Environmental Protection Agency, Remedial Investigation/Feasibility Study Guidance, EPA 540-G-89-004, this is pretty old report though

U.S. Environmental Protection Agency, Guidance for Quality Assurance Project Plans for Modeling, EPA QA/G-5M).

ASME, NQA-1: Quality Assurance Requirements for Nuclear Facility Applications, Subpart 2.7, Quality Assurance Requirements for Computer Software, <https://www.asme.org/codes-standards/find-codes-standards/quality-assurance-requirements-for-nuclear-facility-applications>?).

### Summary

In summary, the governance framework defined in this section ensures that deep-learning surrogate models are integrated into DOE Environmental Management groundwater remediation workflows in a controlled, transparent, and regulator-aligned manner. Governance is grounded in established EPA and DOE quality assurance standards, clearly limits surrogate use to screening- and planning-level decision support, and preserves the primacy of high-fidelity physics-based numerical models for final regulatory and remedy decisions. Objective acceptance criteria, documented recalibration triggers, and formal configuration control ensure that surrogate models are used only within their validated domain. When surrogate validity is in question, governance procedures require explicit withdrawal of surrogate use and escalation to a controlled full retraining process, preventing unmanaged risk. Collectively, this governance foundation enables efficient use of advanced modeling tools to support long-term remediation planning while maintaining scientific rigor, operational stability, and regulatory confidence.

## Modeling Engines (Ross, Xuehang, Yilin)

### Site Challenges and Modeling Needs

Numerical groundwater and reactive transport models provide the high‑fidelity representation of site‑specific subsurface processes that underpin Pillar B. At EM sites, these models describe how contaminants move and transform in complex hydrogeologic settings, and how engineered controls, such as pump‑and‑treat systems, can influence plume evolution. They explicitly resolve groundwater flow, mass transport, and the key processes that drive long‑term risk, providing the “truth” trajectories against which surrogate models can be trained and evaluated.

The physical phenomena represented typically include advection and dispersion of dissolved species, sorption onto solid phases, and associated reaction processes (Sethi and Di Molfetta, 2019; Šimůnek and van Genuchten, 2016). These processes control how contaminant mass is transported, retarded, and transformed in the subsurface and therefore directly affect the magnitude and timing of risks at receptors.

Realistic EM sites are characterized by strong heterogeneity in hydraulic and transport properties. Diffusion‑controlled mass exchange between transmissive and low permeability zones leads to anomalous (non‑Fickian) plume spreading, delayed arrival times, and long-term tailing (Guo et al., 2021). Back diffusion from fine grained or low permeability units is recognized as a major control on plume persistence after source removal, based on field case studies and analytical and numerical models (Yang et al., 2016, 2017). These mechanisms help explain why plumes can remain above regulatory thresholds long after active remediation in the source zone has ceased, so they must be explicitly represented in numerical models.

Density and buoyancy effects can strongly influence plume migration in saline systems (e.g., seawater intrusion) as well as at EM sites impacted by dense and light nonaqueous phase liquids (i.e., DNAPL and LNAPL). At many contaminated sites NAPLs density contrasts govern whether the product floats, sinks, or penetrates below the water table, with strong implications for plume evolution and long term mass discharge (Huling, 1991; McCaulou, 1995; Newell, 1995; RMCS, 2017). In seawater intrusion, contrasts in fluid density modify flow patterns and plume geometry, and ignoring these effects can lead to substantial errors in predicting contaminant migration and mass discharge (Werner et al., 2013; Xu et al., 2019). Consequently, predictive modeling and remediation design for both saline plumes and LNAPL or DNAPL source zones should represent density dependence and, where appropriate, multiphase flow and transport to reduce bias in migration forecasts.

In addition to subsurface physics, EM-relevant models also represent engineered operations and boundary and forcing conditions. Pump‑and‑treat wells, injection and extraction networks, time-varying boundary heads and fluxes, and distributed recharge, are explicitly represented when designing or optimizing remediation systems (Ko et al., 2005). This allows planners to evaluate how different well configurations, pumping rates, permeable reactive barriers, or hydraulic barriers might alter plume pathways and concentrations at compliance points. Across different EM sites, these models have been applied to a range of problems, such as predicting the inland migration of seawater intrusion under coastal pumping, assessing the persistence of back‑diffusion‑controlled plumes at inland facilities, and evaluating how new wells or barriers might alter plume pathways and concentrations at receptors. In Pillar B, this combination of detailed subsurface physics and explicit representation of engineered controls provides the foundation on which surrogate models and recalibration strategies are built.

### Physics-Based Numerical Models

Numerical models are valuable tools because many of the central questions in EM stewardship are predictive and counterfactual in nature. Decision makers need to understand how plumes will behave over decades under alternative remediation strategies, monitoring designs, and future forcing scenarios (e.g., pumping, land use, climate), which cannot be fully resolved by observation alone (Neuman et al., 2003). Field experiments at that scale are rarely feasible, and simple analytical approximations cannot capture the combined effects of heterogeneity, complex geochemistry, and engineered infrastructure (Bear and Cheng, 2010; Zheng and Bennett, 2002). High‑fidelity simulators allow us to explore various scenarios and assumptions, quantify potential outcomes before implementation, and support design and regulatory decision making with transparent, physics‑based predictions (Fienen et al., 2010).

In practice, a few well-established simulators serve as the workhorses for these tasks. Regional groundwater flow is commonly modeled with MODFLOW, a modular finite difference code that supports complex multi‑layer systems and diverse boundary conditions (Harbaugh, 2005), while conservative and reactive solute transport are often simulated with MT3DMS or related extensions (Zheng and Wang, 1999). For more detailed representations of reactive transport, multiphase flow, or density dependent behavior, projects often rely on high-performance simulators such as PFLOTRAN, eSTOMP, or TOUGH2, which can couple variably saturated flow with geochemical reactions, non‑isothermal processes, and variable fluid density (Hammond et al., 2014; Pruess et al., 1999; White and Oostrom, 2003). PARFLOW provides integrated surface and subsurface flow capabilities at high resolution (Maxwell et al., 2015), while HYDRUS and related tools are widely used for vadose zone and near-surface transport where unsaturated flow and root‑zone processes are important (Šimůnek et al., 2016). Within Pillar B, these codes collectively form the high-fidelity modeling layer that defines the governing equations, numerical schemes, and parameterizations we treat as authoritative reference.

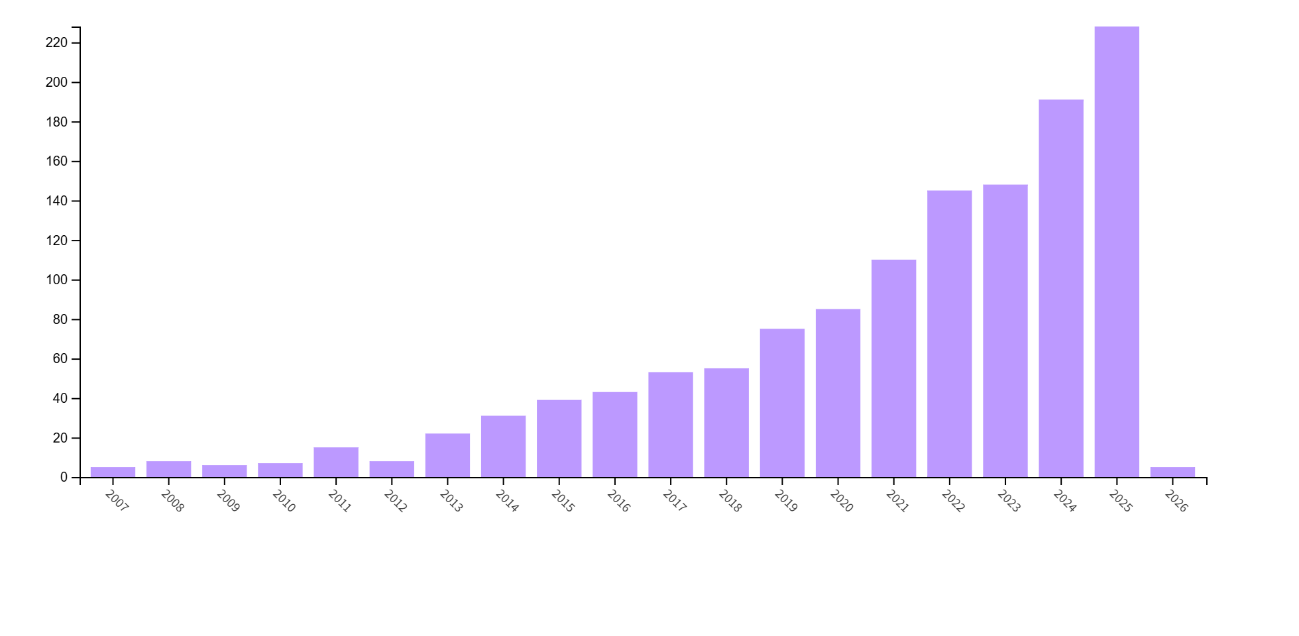
Despite their fidelity, these simulators impose significant computational constraints. Realistic EM applications typically involve three-dimensional, heterogeneous domains, long simulation horizons, and tight numerical tolerances to preserve mass balance and stability. Individual forward runs can take from hours to days on standard computing resources (Hammond et al., 2014; Maxwell et al., 2015). Many of the planning tasks, however, require ensembles of simulations, Monte Carlo realizations of hydrogeologic variability, scenario sweeps over remediation designs, or sensitivity analyses with respect to uncertain parameters (Rubin, 2003). The number of model evaluations needed for robust decision support can easily grow into the hundreds or thousands, making it impractical for planners to interactively explore design space or to fully propagate uncertainty using only the high-fidelity model.

Historically, these numerical models have been calibrated using a combination of expert judgment and automated inverse methods. A common workflow is to iteratively adjust hydraulic conductivities, recharge, boundary conditions, and transport parameters to match observed heads and concentrations, either manually or with parameter estimation tools such as PEST (Doherty, 2004; Hill and Tiedeman, 2007). In some cases, especially where dense time series data are available, data assimilation or ensemble‑based methods (e.g., ensemble Kalman filtering) have been used to update parameters and states in a statistically consistent way (Evensen, 2009; Hendricks Franssen and Kinzelbach, 2008). All of these approaches require repeated forward simulations and careful tuning, which can be labor intensive and computationally expensive, particularly when new data arrives and models must be recalibrated.

These limitations motivate the data-driven surrogate modeling engines that sit on top of the numerical model (Pillar B). Recent work in hydrogeology and subsurface reactive transport has demonstrated that convolutional neural networks, operator-learning architectures such as Fourier Neural Operators (FNO), sequence models, and generative models can emulate high-fidelity simulators with orders-of-magnitude speedup while retaining accuracy (Li et al., 2020). We organize existing surrogate approaches into families, image‑to‑image convolutional models (e.g., U-Net and related CNNs), operator-learning methods such as FNO, sequential architectures that focus on temporal responses, generative models that capture ensembles, and traditional ANN-based response surfaces for low-dimensional metrics. The numerical models described in this section provide training data, physical grounding, and validation targets for these surrogates, while the surrogates aim to reproduce key numerical results at a fraction of the computational cost.

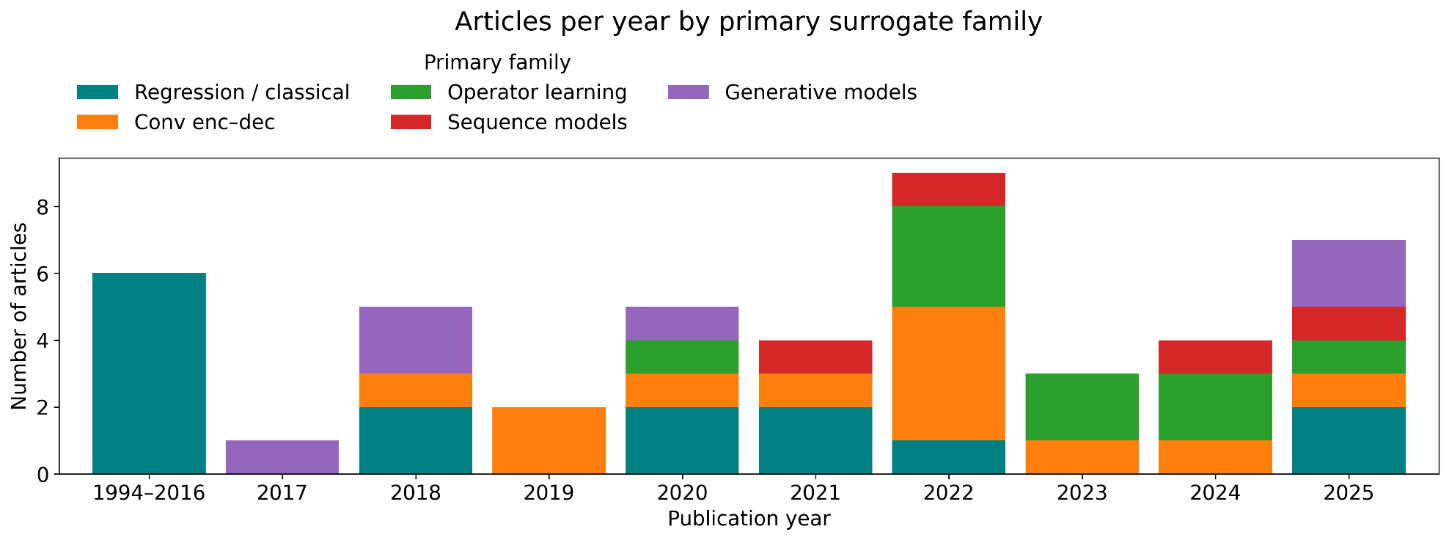
### Surrogate Model Landscape

Surrogate modeling publication trends were quantified using the Web of Science Core Collection (Clarivate Analytics). We performed a topic search (12/5/2025) combining groundwater and subsurface keywords with surrogate-modeling terms (TS = ((groundwater OR subsurface OR aquifer OR "reactive transport" OR "porous media") AND ("surrogate model\*" OR emulator\* OR "reduced-order" OR metamodel\* OR "response surface" OR "Fourier neural operator\*" OR "neural operator\*" OR "operator learning" OR "deep learning surrogate" OR "physics-informed")), limited to articles and reviews in English for the period 2005 to 2026. We then used the Web of Science “Analyze Results” tool to extract the annual number of records by publication year and plotted these counts as the surrogate-model publication trend shown in Figure 1.



**Figure 1.** The article publication growth in surrogate modeling for groundwater and subsurface transport, based on a Web of Science topic search (2005-2025). We observe a modest number of surrogate studies prior to 2010, followed by a rapid increase after ~2018 as deep learning and operator learning architectures enter the field.

Additionally, we constructed a method family breakdown using a screened selection of representative surrogate modeling papers (Appendix Table A2). This corpus of 47 peer-reviewed research articles (excluding review articles) was assembled from papers judged most relevant to subsurface flow and/or transport surrogate modeling and DOE-relevant decision workflows, while ensuring coverage across five primary method families (regression/classical surrogates, convolutional encoder-decoder models, operator-learning/neural operators, sequence models, and generative models). Each paper was assigned a single primary family based on the dominant surrogate architecture used for the main prediction task: 1) regression and classical surrogates (Gaussian processes, Kriging, shallow ANNs, response surfaces), 2) convolutional encoder-decoder models (e.g., U-Net/ResNet) for field-to-field prediction, 3) operator-learning models (e.g., FNO/U‑FNO/DeepONet) that learn discretization-invariant PDE solution operators, 4) sequence models (LSTM/GRU/Transformer variants) for low-dimensional time series targets such as breakthrough curves and well responses, and 5) generative models (VAEs/GANs/diffusion) for ensemble sampling and probabilistic inversion. Figure 2 illustrates that early work (pre‑2017) in this curated set is dominated by classical regression style surrogates, while post-2017 studies increasingly emphasize deep learning surrogates, particularly convolutional and operator-learning approaches, alongside emerging generative methods and sequence models. Because the family breakdown is derived from a screened library rather than the full Web of Science corpus, it is intended to summarize representative method evolution and support the conceptual grouping used in the remainder of this section.

******

**Figure 2*.*** Method evolution in a screened surrogate modeling publication record. Stacked counts of representative groundwater or subsurface surrogate modeling papers by publication year and primary surrogate family. Papers published from 1994 to 2016 are aggregated into a single bin because the curated set in that period is dominated by regression and classical approaches. Classification is based on the dominant surrogate architecture used for the main modeling task (one primary label per paper), and the figure is intended as a representative method overview that complements the full-corpus publication trend shown in ***Figure* 1**.

Diagram

AI-generated content may be incorrect.

**Figure *3*.** Conceptual mind-map of surrogate model families in Pillar B, organized by input*-*output structure and typical EM use*-*cases. Synthesized from published reviews of groundwater surrogates (Luo et al., 2023).*Abbreviations: enc-dec: encoder-decoder, vars: variables, BCs: boundary conditions, ICs: initial conditions.*

Classes of surrogate approaches and where they apply (instead of list all methods plainly, thinking about grouping them to categories):

1. Convolutional encoder–decoder models (U-Nets, ResNets) for high-resolution spatial fields (e.g., heads, salinity, plume extent) where local patterns and interfaces is important. (noted this is one of the model we will test)

Convolutional encoder-decoder models, such as U-Net and its Residual Networks, treat the spatial fields in groundwater and reactive transport models as images and learn a direct mapping from input fields (e.g., hydraulic conductivity, boundary conditions, pumping patterns) to output fields (heads, saturation, concentrations). Because convolutions are local and translation-equivariant, these networks are especially effective at preserving interfaces, sharp fronts, and channelized structures, making them well suited for high resolution plume extent, salinity, or head distributions. Recent work has shown that U‑Net‑type architectures can emulate steady state or transient groundwater flow with high accuracy while reducing runtimes by one to two orders of magnitude compared to MODFLOW or similar finite-difference or finite-element models. Taccari et al. (2022), for example, used an Attention U-Net surrogate to predict steady state hydraulic head fields in heterogeneous aquifers, demonstrating close agreement with numerical solutions and substantial speedups.

For time dependent problems, encoder-decoders are often combined with autoregressive strategies. The surrogate takes the previous predicted state (e.g., saturation or pressure at time 𝑡𝑘) as an additional input to predict the next state at 𝑡𝑘+1. Earlier work for multiphase flow cast surrogate modeling as image‑to‑image regression, highlighting how these architectures naturally fit the “field‑in, field‑out” structure of subsurface PDE solvers (Mo et al., 2019). Jiang et al. (2021) introduced a deep residual U-Net with an autoregressive scheme (AR‑ResUNet) to emulate dynamic multiphase flow and solute transport in channelized geological systems, achieving accurate saturation and pressure fields over long time windows with far fewer training samples than CNN baselines. Lauzon (2024) further showed that a U‑Net surrogate can replace a transient groundwater flow model inside a stochastic inverse framework, enabling fast uncertainty quantification over heterogeneous hydraulic conductivity and boundary conditions. In Pillar B, the convolutional encoder-decoder models will be one of our primary high resolution spatial surrogates, closely aligned with the numerical model grids and with a strong track record in preserving localized plume and interface features.

1. Operator-learning models (e.g., Fourier Neural Operators, U-FNO variants) for problems that require learning input–output maps across many boundary and parameter configurations, especially when we want to reuse the same architecture across multiple sites. (noted this is one of the model we will test)

Operator-learning models generalize beyond pointwise mappings to learn the entire solution operator of the PDE. Given arbitrary input functions (e.g., spatially variable conductivity, source terms, boundary conditions), they approximate the mapping to output functions (e.g., head or concentration fields). The Fourier Neural Operator (FNO) introduced by Li et al. (2020) learns this operator in Fourier space by parameterizing an integral kernel in the frequency domain, enabling resolution‑invariant surrogates that can be evaluated on different meshes and parameter configurations than those seen in training. Follow-on work extended FNO to multiphase flow and complex porous media problems. Wen et al. (2022) proposed U‑FNO, which augments FNO layers with a U‑Net path to improve expressiveness for CO₂-water multiphase systems, achieving higher accuracy and data efficiency than original FNO.

In the groundwater context, Taccari et al. (2024) developed a deep neural operator (DeepONet) surrogate for confined aquifer flow, showing that DeepONet can emulate the impact of pumping on head distributions across multiple scenarios (forward, inverse, and nonlinear problems) with strong generalization across boundary and pumping configurations. Building on U-FNO specifically, Jiang et al. (2025) introduced the GeoFUSE framework for seawater intrusion prediction. A U-FNO surrogate is trained on PFLOTRAN simulations of pressure and salinity over a 20-year period and coupled with PCA-based geological parameterization and an Ensemble Smoother with Multiple Data Assimilation (ESMDA) scheme. GeoFUSE achieves roughly 10⁵ speedup per inference relative to PFLOTRAN while maintaining high fidelity in pressure and salinity fields and uses the surrogate within ESMDA to calibrate heterogeneous permeability and porosity models and substantially reduce uncertainty in salinity accumulation and plume extent over time. Physics‑informed variants of U-FNO have also been proposed to support climate stress testing at contaminated sites. (Wang et al., 2022) used a U‑FNO‑based surrogate to model long‑term flow and transport at the Savannah River Site F‑Area, embedding PDE constraints and boundary conditions directly into the loss function to improve extrapolation under future climate scenarios.

Collectively, FNO/U‑FNO architectures offer a natural way to build site‑agnostic, reusable surrogates. A single operator model can be applied across multiple sites and boundary conditions, making this modeling class central to our goal of redeployable modeling engines in Pillar B.

1. Sequence models (temporal CNNs, LSTMs/Transformers) for 1D/low-dimensional time series such as well breakthrough curves or aggregated metrics that are monitored at high frequency.

Sequence models focus on low-dimensional time series outputs, such as breakthrough curves at wells, aggregate mass discharge, or scalar performance metrics, that are monitored at high frequency but are not spatially distributed fields. Rather than predicting a full 2D or 3D state at each time, these surrogates learn the input to time‑series mapping directly. Long short‑term memory (LSTM) and gated recurrent unit (GRU) networks are widely used here because they can capture long‑range temporal dependencies. For example, Chen et al. (2021) and later Spatiotemporal Attention (STA)‑GRU work (Xie and Zhang, 2024) combine numerical groundwater simulators with GRU‑based surrogates to predict groundwater levels and water quality time series, using attention mechanisms to emphasize critical time windows and locations.

In groundwater contaminant applications, Anshuman and Eldho (2023) proposed an encoder-decoder LSTM framework that takes transient boundary and source terms as inputs and outputs well concentration breakthrough curves, which are then embedded in an uncertainty quantification loop for contaminant source identification. More recent hybrid architectures couple temporal RNNs with shallow CNNs to capture both spatial correlations among multiple monitoring locations and temporal dynamics. For instance, Li et al. (2025) use a Deep Separable Convolutional Neural Network (DSCNN)-GRU surrogate to approximate transient groundwater head fields while explicitly modeling the correlation structure among multiple time series outputs.

Additionally, deep surrogates with spatiotemporal awareness have been developed for water quality sensor networks and other hydrologic systems, reinforcing the value of attention‑enhanced RNNs and temporal CNNs for multi‑site, multi‑variable time series (Zhang and Thorburn, 2022).

In Pillar B, this model family is most appropriate when the decision variables and objectives are defined on time series at a limited number of locations (e.g., compliance wells), rather than on full high resolution fields.

1. Generative models (diffusion models or VAEs) for tasks that require sampling full state ensembles consistent with physics and observations, for example generating multiple plausible plume realizations conditioned on monitoring data. (SRNL’s monitoring network desgin)

Generative surrogates aim not just to predict a single trajectory but to sample entire ensembles of plausible states and parameter fields consistent with physics and observations. Early work in subsurface applications used variational autoencoders (VAEs) and generative adversarial networks (GANs) to generate facies realizations and geological models, and to perform Bayesian inversion conditioned on sparse pressure or saturation data in CO2 storage settings (Graham and Chen, 2020). More recently, diffusion models have emerged as the state of the art for multivariate subsurface generation and probabilistic inversion. Miele and Linde (2025) demonstrate that diffusion models produce more statistically robust and flexible multivariate subsurface property fields than VAEs or GANs and can be naturally conditioned on hard and indirect data (e.g., well logs and seismic).

For flow and transport, Wang et al. (2025c) introduce a pretrained diffusion model that jointly learns the distribution of subsurface parameters and state variables for porous‑media flow. With Bayesian conditional sampling, a single trained model can be reused across tasks including unconditional generation, forward prediction, uncertainty quantification, and inverse modeling with sparse noisy data.

Conditional diffusion frameworks have also been proposed specifically for geologic CO2 storage and plume forecasting, where diffusion-based generators are conditioned on monitoring data to produce ensembles of pressure and saturation fields, enabling inverse modeling and uncertainty quantification without repeated full‑order simulations (Wang et al., 2025b).

In the context of EM‑style sites, these generative surrogates are particularly attractive for monitoring network design and risk assessment. Given an existing monitoring network, the model can generate many plausible plume realizations consistent with measured heads or concentrations, which downstream optimization can then use to compare alternative sensor placements or remedial strategies under ensemble uncertainty.

1. Simpler regression surrogates (Gaussian processes, shallow ANNs) for low-dimensional mappings (e.g., design variables → scalar performance metrics) where interpretability and small-data robustness are more important than high-dimensional resolution.

For many decision‑support tasks, the quantities of interest are low‑dimensional scalar or vector metrics. For example, total remediation cost, cleanup time to reach a regulatory threshold, or a risk index, defined as outputs of a high‑fidelity transport simulator. In these cases, simpler regression surrogates such as Gaussian process (GP) models, Kriging, and shallow artificial neural networks (ANNs) remain powerful tools, especially when data are limited and interpretability matters. Asher et al. (2015) review surrogate modeling in groundwater and emphasize that Kriging, GP and low‑complexity ANNs are still widely used to emulate scalar metrics and to support optimization and uncertainty analysis. Siade et al. (2020) applied reduced‑dimensional GP regression for groundwater allocation planning, using singular value decomposition to compress the input space and then building a GP surrogate for allocation performance metrics, which enabled efficient swarm‑based optimization over extraction and injection strategies.

Gaussian process surrogates have also been used directly in remediation design. Shams et al. (2021) constructed a GP surrogate of a surfactant‑enhanced aquifer remediation model at DNAPL contaminated sites and used it within an optimization loop to evaluate different well configurations and operational strategies at greatly reduced computational cost. At a more methodological level, recent reviews by Marrel and Iooss (2024) summarize advances in GP surrogate construction, hyperparameter estimation, and validation, providing guidance on building robust probabilistic surrogates with credible uncertainty bounds.

Shallow ANNs and response‑surface models (e.g., polynomial chaos expansions, radial basis function ANNs) are also frequently used as scalar surrogates in groundwater contaminant modeling and optimization, as cataloged in the recent review by Luo et al. (2023). In Pillar B, we view this family as the workhorse for design‑variable to metric mappings, where model dimensionality is low, training data may be scarce, and we care about having uncertainty estimates and interpretable relationships more than capturing fine‑scale spatial structure.

1. **Structured domain surrogates**

A complementary class of surrogates represents the system state on a structured basis or on a graph or mesh and then learns the evolution operator in that representation. This includes projection-based reduced-order models such as Proper Orthogonal Decomposition (POD) and Dynamic Mode Decomposition (DMD), Koopman operator approaches, and graph/mesh neural surrogates (Sanchez-Gonzalez et al., 2020). These methods are common in broader scientific computing and fluid mechanics, and they are increasingly being adapted to porous-media flow and transport settings where the governing dynamics can be represented in low-dimensional coordinates or on unstructured discretization. Rather than learning direct input–output mappings, these methods focus on learning how system states change through underlying operators defined on evolving structured representations, such as low-dimensional modes (Schmid, 2010), lifted observable spaces (Mezić, 2013), or graphs and meshes (Pfaff et al., 2020). Because these approaches are less prevalent in the screened groundwater and remediation corpus used for Figures 2 and 4, we discuss them here as adjacent methods and do not include them in the five-family counts.

Diagram

AI-generated content may be incorrect.

**Figure 4.** Heat map summary of the selected, application focused surrogate modeling corpus. Top: counts of papers by surrogate method family versus application. Bottom: counts of papers by surrogate method family versus project domain. Cell values denote the number of papers assigned to each pairing. Each paper is assigned a single primary label per axis to prevent double counting.

**Figure 4** summarizes how the selected, application-focused surrogate-model papers distribute across application family and project domain, stratified by primary surrogate method family. Each cell reports the number of papers assigned to that pair. Two patterns stand out. First, regression/classical surrogates remain most prevalent in design & optimization workflows (e.g., surrogate-assisted remediation planning), reflecting their efficiency and interpretability for low-dimensional decision variables and scalar objectives. Second, deep-learning methods show strong specialization. Convolutional encoder-decoder models are most represented in inverse & calibration uses where high-dimensional spatial fields are central. Operator-learning methods concentrate in forward emulation, consistent with learning reusable solution operators across many boundary/parameter settings. Sequence models appear primarily in monitoring & time-series prediction, and generative models cluster in UQ & ensemble generation and probabilistic inversion tasks.

Project domains in **Figure 4** indicate the physical system in which each surrogate approach is demonstrated. Contaminant hydrogeology/remediation includes subsurface contaminant transport and engineered cleanup problems (e.g., pump-and-treat design, DNAPL dissolution, reactive transport, plume persistence and back-diffusion), where surrogates are valuable because decision workflows are long-horizon and often require many repeated simulations for design, compliance, and uncertainty analysis. Groundwater flow/fluid mechanics covers Darcy-scale groundwater flow and related porous-media flow problems that emphasize forward emulation and inverse estimation of properties (e.g., conductivity fields), making field-to-field surrogate mappings (CNNs and operator-learning models) especially useful. Coastal aquifers/seawater intrusion focuses on density-driven flow and salinity transport under pumping and recharge forcing, where long transient horizons and many management scenarios motivate surrogates for rapid forecasting, optimization, and data assimilation. Groundwater monitoring/forecasting includes studies centered on predicting and interpreting well/sensor time series (levels, water quality signals), where surrogates enable operational forecasting, drift detection, and near-real-time analytics, which often aligning naturally with sequence models. CO2 storage/geoenergy covers geologic carbon storage and related geoenergy applications (pressure buildup, plume migration), where high-dimensional 3D simulations and probabilistic risk assessment drive the need for fast surrogate inference (commonly operator-learning and CNN-based). Finally, general scientific computing includes method-development and benchmark-focused papers evaluated on canonical PDE testbeds (e.g., Darcy, Burgers, Navier-Stokes), these works provide transferable architectures and theory that are frequently adopted later by groundwater and subsurface application studies.

The domain heatmap indicates that the selected corpus is weighted toward contaminant hydrogeology/remediation and groundwater flow/fluid mechanics, while operator-learning and some deep models also appear in adjacent domains such as CO₂ storage/geoenergy and general scientific computing, illustrating cross-domain transfer of surrogate modeling architectures.

### General Surrogate Model Training Setup (Xuehang, Ross, TC):

* + Data splits; augmentation; convergence/early stopping
  + Splits: Hold out realizations across time and parameter ranges. (Jiang et al., 2025) used separate training and test ensembles, which is a practical benchmark for our initial runs.
  + Normalization: Standardize each channel and record the scaling in metadata so inference is consistent.
  + Early stopping: Track the validation metric on the decision outputs, not only on global averages, and save the best checkpoint.
  + Runtime reporting: Record forward runtime and surrogate inference time so speedups can be reported next to accuracy. The (Jiang et al., 2025) case study provides useful reference numbers for this comparison.

## Recalibration Mechanisms (TC, Xuehang, Jason)

The recalibration component operates as a single, governed loop designed for stability, transparency, and regulatory credibility. The sequence is monitor → decide → update (if warranted) → verify. During routine operations, the surrogate remains unchanged; the team conducts monthly monitoring to track (i) accuracy relative to the most recent stable baseline and (ii) calibration of stated uncertainty (for example, the empirical coverage of the 95% prediction interval). Operational changes (e.g., revisions to pumping schedules or boundary conditions) are logged because they can move inputs outside the surrogate’s original training setting.

On a fixed decision window (e.g., quarterly), the program determines whether an update is justified. Action is taken only when evidence of drift has persisted for 60–90 days or when a documented regime change has occurred. When action is justified, the framework does not undertake a whole retrain; instead, it executes a small, targeted update supported by a limited number of physics simulations and the most recent field observations. The revised surrogate is then evaluated against pre-registered acceptance criteria (Section 3.3), including accuracy, uncertainty coverage, and performance at priority wells or zones. Throughout, the high-fidelity physics model remains the authoritative reference used to inform and validate changes.

In rare circumstances—such as repeated quarterly failures or substantive changes to the operating domain, the program may invoke a full retrain (see subsection below).

### Uncertainty Quantification of Surrogate Models (TC & Jason)

The integration of Uncertainty Quantification (UQ) within hybrid modeling is motivated by the need to reconcile the rapid execution of Machine Learning (ML) surrogates with the rigorous physical consistency of Data Assimilation (DA). By formalizing UQ across both domains, researchers provide reliable confidence intervals, identifying when a system is operating outside its known regime, to make sure that surrogate models remain physically bounded and that assimilation cycles are informed by high-fidelity error characterizations, ultimately enhancing the reliability of autonomous decision-making in complex dynamical systems.

Possible workflow:

* Surrogate UQ Methodologies — For surrogate-based UQ, several approaches can be leveraged to partition aleatoric and epistemic uncertainties. Monte Carlo dropout (Gal and Ghahramani, 2016) and deep ensembles (Lakshminarayanan et al., 2017) are commonly used to estimate epistemic uncertainty by measuring model disagreement. Alternatively, conformal prediction offers statistically valid uncertainty bounds without requiring strong distributional assumptions, ensuring guaranteed coverage for predictive intervals (Shafer and Vovk, 2008).
* Physical Model UQ via Data Assimilation — In contrast, physical model UQ is handled via data assimilation techniques that reconcile model states with noisy observations. The Ensemble Kalman Filter (EnKF) is applied when Gaussian assumptions hold and provides a computationally efficient way to propagate covariance (Evensen 2003). For non-Gaussian or highly nonlinear contexts, particle filters are preferred to represent the posterior distribution through a set of weighted samples (Särkkä, 2013).
* Sequential Bayesian Inference — For complex or high-dimensional systems, sequential Bayesian inference is used to iteratively refine posterior distributions as new observations are integrated (Kennedy and O'Hagan 2001). This iterative process ensures that the physical model’s state remains aligned with the evolving real-world system.
* Hybrid Integration Strategy — This strategy treats surrogate UQ and physical-model DA as complementary. Surrogates are utilized for fast, uncertainty-aware forecasting, while the DA framework handles rigorous state correction and parameter tuning using real-time observations (Diaa-Eldeen et al., 2025).
* Feedback and Continual Evolution — A formal feedback loop is established where surrogate outputs, specifically the predicted mean and uncertainty, to guide the localization and weight of assimilation updates (Cheng et al., 2023).

References:

Cheng, S., Quilodrán-Casas, C., Ouala, S., Farchi, A., Liu, C., Tandeo, P., ... & Arcucci, R. (2023). Machine learning with data assimilation and uncertainty quantification for dynamical systems: a review. IEEE/CAA Journal of Automatica Sinica, 10(6), 1361-1387.

Diaa-Eldeen, T., Berg, C. F., & Hovd, M. (2025). Data-Driven Spectral Methods for Stochastic Surrogate Modeling and Efficient Data Assimilation in Subsurface Flows. Mathematical Geosciences, 1-25.

Evensen, Geir. 2003. "The Ensemble Kalman Filter: Theoretical Formulations and Practical Implementations." Ocean Dynamics 53 (4): 343–67.

Gal, Yarin, and Zoubin Ghahramani. 2016. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning." In International Conference on Machine Learning (ICML), 1050–59.

Kennedy, Marc C., and Anthony O'Hagan. 2001. "Bayesian Calibration of Computer Models." Journal of the Royal Statistical Society Series B: Statistical Methodology 63 (3): 425–464.

Lakshminarayanan, Balaji, Alexander Pritzel, and Charles Blundell. 2017. "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles." Advances in Neural Information Processing Systems (NeurIPS) 30.

Särkkä, Simo. 2013. Bayesian Filtering and Smoothing. Cambridge: Cambridge University Press.

Shafer, Glenn, and Vladimir Vovk. 2008. "A Tutorial on Conformal Prediction." Journal of Machine Learning Research 9 (3): 371–421.

Yang, Yibo, and Paris Perdikaris. 2018. "Adversarial Uncertainty Quantification in Physics-Informed Neural Networks." Journal of Computational Physics 394: 136–52.

Zhu, Yinhao, Nicholas Zabaras, P-S. Koutsourelakis, and Paris Perdikaris. 2019. "Physics-Constrained Deep Learning for High-Dimensional Surrogate Modeling and Uncertainty Quantification without Labeled Data." Journal of Computational Physics 394: 56–81.

### Trigger logic (TC & Jason)

XS: Signals: residuals vs monitoring, uncertainty inflation (variance > threshold), drift tests (10.3389/frai.2024.1330257/full).

XS: Gating/Threshold for each signa

#### As models operate over time, data distributions and environmental conditions evolve, a trigger logic needs to be established that activates model recalibration to maintain model trustworthiness and regulatory compliance. Early signs of model drift, uncertainty inflation, or residual divergence tell when surrogate models or DA systems are no longer reliable, therefore statistical thresholds with domain-informed criteria can be combined to allow adaptive learning while maintaining and improving model interpretability and performance. Regime Drift Monitoring Framework

We propose a drift monitoring framework consists of multiple, complementary metrics and indicators of regime drift. Below we list potential categories of such metrics and indicators, including tipping point–inspired resilience metrics, performance- and uncertainty-based indicators, distributional and representation-level drift detection, and probabilistic decision logic. By combining complementary signals that capture loss of dynamical stability, increasing model uncertainty, and structural changes in data and learned representations, the framework enables early, interpretable, and statistically robust detection of regime drift to support timely model recalibration.

### ****Tipping Point–Inspired Resilience Metrics****

This category captures early signs of loss of dynamical stability, motivated by concepts from tipping point science. These metrics are applied to model outputs, residuals, and, where applicable, latent representations, without implying the existence of physical tipping points in the underlying system.

Rolling-window estimates of variance and lag-1 autocorrelation are used to detect increasing fluctuation amplitude and memory, consistent with reduced resilience. Higher-order moments, including skewness and kurtosis, are monitored to identify asymmetric error growth and heavy-tailed behavior indicative of excursions toward unstable regions.

### ****Performance and Uncertainty-Based Metrics****

This category focuses on direct indicators of model reliability under deployment. Prediction residuals are monitored using adaptive thresholds derived from rolling statistics, such as exceedance of multiples of the rolling standard deviation. Persistent threshold violations or clustered spatial–temporal error patterns are treated as evidence of localized or systemic degradation.

For probabilistic models, predictive uncertainty is explicitly tracked using measures such as variance or entropy. Sustained increases relative to historical baselines are interpreted as growing epistemic uncertainty and declining confidence in model predictions.

### ****Distributional and Representation-Level Drift Detection****

To detect drift that may not immediately manifest in errors or uncertainty, the framework includes diagnostics targeting statistical and structural changes in data and learned features.

Distributional drift is assessed by comparing incoming input and output distributions to reference distributions using nonparametric statistical tests, such as the Kolmogorov–Smirnov test. Statistically significant deviations indicate changes in the data-generating process that may compromise model validity.

In parallel, drift indicators are applied in learned representation spaces, such as latent embeddings from diffusion models or bottleneck features from convolutional architectures. Deviations from the training manifold in latent space provide early detection of regime mismatch in high-dimensional settings where marginal statistics may be insufficient.

### ****Probabilistic Integration and Decision Logic****

The final category integrates all diagnostic signals into a robust decision-making layer. Rather than relying on single-metric thresholds, the framework employs Bayesian inference to combine evidence from resilience, performance, uncertainty, and distributional indicators with prior knowledge and alternative explanations (e.g., data quality changes).

Model recalibration or adaptation is triggered through composite gating mechanisms, implemented either as Boolean rules (requiring concurrent exceedance across multiple metric families) or as score-based aggregation into a unified drift risk index. This integration reduces false positives due to noise or transient anomalies and ensures that interventions are driven by consistent, multi-source evidence.

|  |  |  |
| --- | --- | --- |
| Metric / Indicator | Category | Applied To |
| Rolling variance | Resilience | Outputs, residuals |
| Lag-1 autocorrelation | Resilience | Residuals, latent |
| Skewness / kurtosis | Resilience | Residuals |
| RMSE / bias | Performance | Outputs |
| Threshold exceedance | Performance | Residuals |
| Predictive variance | Uncertainty | Outputs |
| KS / energy distance | Distributional | Outputs |
| Maximum mean discrepancy | Distributional | Multivariate output fields |
| Latent Mahalanobis | Representation | Latent space |
| Reconstruction error | Representation | Latent space, outputs |
| Composite drift index | Integration | All |

Table 1. Exemplary Regime Drift Diagnostics Across Metric Categories.

#### Proposed Workflow for Developing a Drift Monitoring System

This research proposes a three-phase workflow to translate generic drift metrics into practical tools for monitoring groundwater model reliability.

* Phase 1: Candidate Identification & Method Selection: The first step involves curating a specific subset of indicators from the broader regime drift metrics that are physically and statistically relevant to subsurface flow and transport dynamics.
* Phase 2: Simulation-Based Robustness Testing: Once candidates are selected, they are subjected to rigorous stress testing using synthetic test cases derived from physics-based model simulations.
  + Scenario Generation: A suite of test cases is generated representing common subsurface regime shifts, such as rapid aquifer depletion, saltwater intrusion fronts, or changes in recharge patterns. These simulations explicitly model the transition from stable to unstable regimes.
  + Reliability Analysis: The selected candidates are applied to these simulations to quantify their sensitivity (true positive rate) and specificity (true negative rate).
  + Addressing Ambiguity: To counter statistical pitfalls like the "prosecutor’s fallacy" and the non-specificity of generic signals, a Bayesian evaluation framework is applied. This step assesses how well each indicator performs in the presence of realistic hydrogeological noise and variable data quality, filtering out methods that generate excessive false alarms.
* Phase 3: Operational Implementation & Recalibration Triggers: The final phase synthesizes the evaluation results to establish a standardized protocol for model maintenance.
  + Selection of Best Performers: The indicators that demonstrate high robustness and lead time in Phase 2 are adopted as the core monitoring suite.
  + Defining Triggers: Quantitative thresholds for these indicators are established to serve as formal "recalibration triggers." When a composite index of these signals (e.g., rising autocorrelation combined with latent space drift) exceeds a critical value, it automatically flags the groundwater model as unreliable.
  + Community Recommendation: The findings are distilled into a set of best practices for the subsurface modeling community, recommending specific drift monitoring workflows to preemptively identify when data-driven or conceptual models require retraining or structural adjustment.

Additional metrics:

* + - Residual/uncertainty-based signals such as dynamic error thresholds (e.g., residuals > 2× rolling standard deviation or spatial-temporal clustering of misfit) and inflation in predictive variance or entropy relative to historical baselines.
    - Statistical comparison of incoming inputs/outputs against reference distribution (e.g., Kolmogorov Smirnov test), where significant drift (e.g., p < 0.01) would activate recalibration.
    - Combined/composite gating logic by integrating multiple signals using Boolean or score-based rule to trigger model updates while reducing false positive and improving lead time.

1. Continual Learning **(**Xuehang**)**

Continual learning for scientific surrogate models focuses on updating a model in response to gradual system change while preserving valid historical behavior. Mature approaches fall into two widely adopted categories: (1) regularization-based techniques, which constrain parameter updates to avoid overwriting previously important knowledge, and (2) replay-based techniques, which maintain stability by including a curated set of historical samples during updating. Elastic Weight Consolidation (Kirkpatrick et al., 2017) is a prototypical regularization strategy, penalizing deviations from weights identified as influential in previous training. Replay-based strategies are widely applied across engineering and environmental forecasting domains to reduce the risk of catastrophic forgetting when conditions shift incrementally (De Lange et al., 2021). Within the environmental modeling literature, sequential fine-tuning has repeatedly been shown to outperform static surrogates in non-stationary systems such as energy load forecasting (Hurtado et al., 2023) and long-horizon operational forecasting (Sayed et al., 2025). These same studies emphasize that frequent retraining is unnecessary—and often destabilizing—when well-designed continual learning mechanisms are in place. This aligns with findings in hydrologic surrogate modeling, where modest periodic updates informed by new hydrologic states and compact replay buffers maintain performance while avoiding the computational cost of full retraining (e.g., Asher et al., 2015; Wang et al., 2023). Taken together, the literature supports a conservative, stability-focused approach to continual learning that fits DOE operational requirements.

Fine-tuning is initiated only after documented indicators of drift demonstrate meaningful degradation—such as sustained increases in predictive error, divergence between modeled and observed temporal trends, or reduced empirical coverage of uncertainty intervals (e.g., under coverage in UQ metrics). Once triggered, the update process proceeds through several structured steps.

* Assemble a compact update dataset consisting of (i) targeted physics simulations representing the changed regime, (ii) recent validated field observations, and (iii) a replay sample of historical cases, typically 70–75% of the batch, consistent with replay ratios shown effective in continual learning applications (De Lange et al., 2021).
* Freeze early encoder layers to preserve structural representations learned from the original physics-informed domain, while updating later layers using a small learning rate and a regularization penalty such as EWC or L2 anchoring, following recommendations from continual learning studies in dynamic engineering systems (Hurtado et al., 2023).
* Monitor validation performance and stop early once improvements plateau, following best practices for stability in sequential fine-tuning.
* Recalibrate predictive uncertainty using held-out data or ensemble-based UQ methods to ensure that nominal intervals align with empirical coverage, consistent with uncertainty calibration practices in hydrology and geosciences (e.g., Zhu et.al, 2019; Wen et al., 2021).
* Prepare a governance package including pre-/post-update performance metrics, drift indicators, and updated uncertainty diagnostics. The surrogate is promoted to operational status only if acceptance criteria are met; otherwise, the system reverts to the prior model.

Asher, M. J., Croke, B. F., Jakeman, A. J., & Peeters, L. J. (2015). A review of surrogate models and their application to groundwater modeling. Water Resources Research, 51(8), 5957–5973. <https://doi.org/10.1002/2015WR016967>

De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., & Tuytelaars, T. (2021). A continual learning survey: Defying forgetting in classification tasks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(7), 3366–3385. <https://doi.org/10.1109/TPAMI.2021.3057446>

Zhu, Y., Zabaras, N., Koutsourelakis, P.-S., & Perdikaris, P. (2019). Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. Journal of Computational Physics, 394, 56–81. <https://doi.org/10.1016/j.jcp.2019.05.024>

Hurtado, J., Salvati, D., Semola, R., Bosio, M., & Lomonaco, V. (2023). Continual learning for predictive maintenance: Overview and challenges. Intelligent Systems with Applications (arXiv:2301.12467).

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., et al. (2017). Overcoming catastrophic forgetting in neural networks. Proceedings of the National Academy of Sciences, 114(13), 3521–3526. <https://doi.org/10.1073/pnas.1611835114>

Sayed, A. N., Himeur, Y., Varlamis, I., & Bensaali, F. (2025). Continual learning for energy management systems: A review of methods and applications, and a case study. Applied Energy, 384, Article 125458. <https://doi.org/10.1016/j.apenergy.2025.125458>

Wen, G., Hay, C., & Benson, S. (2021). CCSNet: : A deep learning modeling suite for CO2 Storage. Advances in Water Resources, 149, 104009. <https://doi.org/10.1016/j.advwatres.2021.104009>

### Active Learning (Xuehang)

Active learning identifies new physics simulations that provide the highest expected improvement in surrogate accuracy, stability, or uncertainty reduction. Mature approaches include uncertainty sampling, which selects simulations at locations with high predictive variance (Settles, 2009), and error-based sampling, which targets regions where model–observation residuals are large or persistent (Cohn et al., 1996). These strategies have been widely used in groundwater inversion, contaminant transport modeling, and CO₂ storage forecasting, where simulation costs are high and surrogate models must be continually refined (Wang et al., 2023). Recent hydrologic literature emphasizes the value of selecting simulations near the “vicinity of the solution manifold,” where small corrections yield substantial improvements in predictive fidelity. Diversity constraints—such as spatial dispersion, stratigraphic coverage, or forcing variability—are also well-established and shown to prevent sampling redundancy while enhancing surrogate generalization (Wen et al., 2021). These approaches are simple, interpretable, and well aligned with DOE’s operational context, where transparency and governance are essential.

Active learning is invoked only when a quarterly update is justified by monitoring diagnostics. Once activated, the process progresses through the following structured steps.

* Generate a candidate pool of operationally relevant simulation scenarios representing the current regime and near-term management options (e.g., pumping schedules, well reconfigurations).
* Score candidates using uncertainty and error metrics, including predictive variance, credible interval width, and residual patterns relative to recent field observations, consistent with methods applied in hydrologic surrogate refinement (Wang et al., 2023).
* Apply diversity constraints to ensure selected scenarios span hydraulic gradients, hydrostratigraphic units, and forcing conditions rather than clustering in redundant areas.
* Select 6–10 high-value simulations that collectively maximize expected information gain, consistent with batch active learning recommendations in environmental modeling (Vesselinov et al., 2017).
* Document simulation rationale, including expected uncertainty reduction and operational relevance.
* Execute selected simulations and incorporate results directly into the fine-tuning batch described in Section 3.2.3, ensuring integration with continual learning and drift mitigation mechanisms.

Cohn, D., Ghahramani, Z., & Jordan, M. (1996). Active learning with statistical models. Journal of Artificial Intelligence Research, 4, 129–145. <https://doi.org/10.1613/jair.295>

Settles, B. (2009). Active learning literature survey. University of Wisconsin–Madison, Computer Sciences Technical Report 1648.  
<https://minds.wisconsin.edu/handle/1793/60660>

Wang, J., Chang, H., & Zhang, D. (2023). Inverse Modeling for Subsurface Flow Based on Deep Learning Surrogates and Active Learning Strategies. Water Resources Research, 59(5), e2022WR033644. [https://doi.org/10.1029/2022WR033644](https://doi.org/10.1029/2022WR033644" \t "_new)

Wen, G., Hay, C., & Benson, S. (2021). CCSNet: : A deep learning modeling suite for CO

Storage. Advances in Water Resources, 149, 104009. [https://doi.org/10.1016/j.advwatres.2021.104009](https://doi.org/10.1016/j.advwatres.2021.104009" \t "_new)

### Multi-fidelity Data Fusion (Jason, Xuehang):

Multi-fidelity modeling enhances surrogate accuracy and computational efficiency by combining inexpensive low-fidelity simulations with high-fidelity simulations and field observations. Two mature strategies dominate the literature: (1) autoregressive Gaussian-process and co-Kriging models, which treat high-fidelity outputs as corrected versions of low-fidelity predictions (Kennedy & O’Hagan, 2000; Le Gratiet, 2013), and (2) neural discrepancy models, which learn nonlinear mappings between fidelity levels (Perdikaris et al., 2017, CMAME). In groundwater flow and reactive transport applications, low-fidelity simulations (e.g., coarse-grid PFLOTRAN or MODFLOW models) provide broad characterization of system variability, while fine-grid simulations or field observations anchor localized accuracy (Asher et al., 2015). Recent advances include physics-aware multi-fidelity networks (Wen et al., 2021) and transfer learning approaches in which surrogates are pre-trained on large low-fidelity ensembles and fine-tuned using a smaller high-fidelity subset (Tang et al., 2022), achieving strong performance with orders of magnitude reduction in simulation cost. These methods integrate naturally with active learning and continual learning, enabling scalable refinement even under evolving operational regimes.

The multi-fidelity fusion procedure integrates coarse simulations, fine simulations, and field data through a structured workflow.

* Identify fidelity levels and align datasets, ensuring spatial, temporal, and unit consistency across LF, HF, and observational sources.
* Pre-train surrogate models on low-fidelity simulations, exploiting their broad coverage of the parameter and operational space.
* Introduce high-fidelity data—simulation or observational—and train a discrepancy model or co-Kriging structure so that predictions inherit LF global structure and HF local accuracy (Kennedy & O’Hagan, 2000; Perdikaris et al., 2017).
* Quantify fidelity-conditioned uncertainty, expanding intervals where only LF information is available and narrowing them where HF evidence supports confidence, consistent with UQ strategies in Wen et al. (2021).
* Validate fused predictions against withheld HF runs or independent observational periods to confirm improved performance over single-fidelity surrogates.
* Integrate the fused surrogate into the continual learning loop, enabling more accurate drift detection, better-informed active learning selection, and more stable fine-tuning updates.

Asher, M. J., Croke, B. F., Jakeman, A. J., & Peeters, L. J. (2015). A review of surrogate models and their application to groundwater modeling. Water Resources Research, 51(8), 5957–5973. <https://doi.org/10.1002/2015WR016967>

Kennedy, M. C., & O’Hagan, A. (2000). Predicting the output from a complex computer code when fast approximations are available. Journal of the Royal Statistical Society: Series B, 62(3), 425–464. <https://doi.org/10.1111/1467-9868.00293>

Le Gratiet, L. (2013). Multi-fidelity Gaussian process regression for computer experiments. Computational Statistics & Data Analysis, 66, 147–159. <https://doi.org/10.1016/j.csda.2013.03.018>

Perdikaris, P., Raissi, M., Damianou, A., Lawrence, N., & Karniadakis, G. E. (2017). Nonlinear information fusion algorithms for data-efficient multi-fidelity modeling. Proceedings of the Royal Society A: Mathematical, Physica. [https://doi.org/10.1098/rspa.2016.0751](https://doi.org/10.1098/rspa.2016.0751" \t "_blank)

Tang, M., Ju, Y., & Durlofsky, L. J. (2022). Multi-fidelity surrogate modeling for CO₂ storage optimization under uncertainty. International Journal of Greenhouse Gas Control, 118, 103692. <https://doi.org/10.1016/j.ijggc.2022.103692>

Wen, G., Hay, C., & Benson, S. (2021). CCSNet: Deep learning surrogate for coupled flow–geomechanics in geologic carbon storage. Advances in Water Resources, 149, 104009. <https://doi.org/10.1016/j.advwatres.2021.104009>

Geneva, N., & Zabaras, N. (2020). *Multi-fidelity generative deep learning turbulent flows*. Foundations of Data Science, 2(4), 391–428. [https://doi.org/10.3934/fods.2020019](https://doi.org/10.3934/fods.2020019" \t "_new)) (not used)

**3.2.6 Fully Retrain (Xuehang)**

Full retraining is reserved for circumstances in which the existing surrogate is no longer representative of the physical system or operational domain. Concept drift literature emphasizes that full retraining is appropriate when foundational assumptions underlying the surrogate no longer hold, such as regime shifts, major forcing changes, or redefinition of the underlying physics model (Parisi et al., 2019; Sayed et al., 2025). In groundwater modeling, full retraining is often required after recalibration of the numerical model, introduction of new hydrogeologic interpretations, or major reconfiguration of pumping and boundary conditions (Laloy et al., 2017). Studies in multi-fidelity and physics-informed ML similarly reinforce that when high-fidelity simulations deviate substantially from prior domains, incremental updates cannot maintain integrity, and a new baseline surrogate is needed (Wen et al., 2021). Retraining provides a clean, traceable reset that aligns with DOE’s emphasis on lifecycle documentation and reproducibility.

The retraining process proceeds through a formal set of steps.

* Verify retraining triggers, including major operational changes, substantive updates to the physics model (e.g., revised geologic model or reaction network), or repeated fine-tuning failures indicating that the training domain has materially shifted.
* Redesign the training domain, defining updated ranges of hydrofacies, operational scenarios, and boundary forcings consistent with new conceptual understanding.
* Generate a new ensemble of high-fidelity simulations, supplemented where appropriate by low-fidelity runs to maintain broad coverage and accelerate sampling.
* Integrate recent field observations, anchoring the surrogate to empirical system behavior.
* Train a new surrogate from scratch, including UQ calibration, validation against benchmark scenarios, and documentation of performance improvements relative to the previous baseline.
* Designate the new model as the operational baseline, while archiving the prior surrogate, training data, and diagnostic artifacts to preserve full scientific and regulatory traceability.

Levy, S., Hunziker, J., Laloy, E., Irving, J., & Linde, N. (2022). Using deep generative neural networks to account for model errors in Markov chain Monte Carlo inversion. Geophysical Journal International, 228, 1098–1118.

<https://doi.org/10.1093/gji/ggab391>

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., & Wermter, S. (2019). Continual lifelong learning with neural networks: A review. Neural Networks, 113, 54–71. <https://doi.org/10.1016/j.neunet.2019.01.012>

Sayed, A. N., Himeur, Y., Varlamis, I., & Bensaali, F. (2025). Continual learning for energy management systems: A review of methods and applications, and a case study. Applied Energy, 384, Article 125458. <https://doi.org/10.1016/j.apenergy.2025.125458>

Wen, G., Hay, C., & Benson, S. (2021). CCSNet: Deep learning surrogate for coupled flow–geomechanics in geologic carbon storage. Advances in Water Resources, 149, 104009. <https://doi.org/10.1016/j.advwatres.2021.104009>

# Validation & Metrics (should we move this to 3.0?)

* **Test scenarios: Test scenarios could include locating and optimizing pumping wells to capture/remediate a contaminant plume, surrogate‑based well placement, monitoring frameworks require spatial‑temporal data (chemicals, heads, flows),and model outputs and decision‑making metrics (e.g., monitoring optimization, remediation performance).**
* **Primary metrics:**
  + **Accuracy: e.g. reduction in contaminant mass or concentration at target wells; mean drawdown head error; RMSE of modeled plume extent vs observed.**
  + **Uncertainty: e.g., predictive interval coverage for contaminant concentration; mass discharge uncertainty; confidence in capture zones.**
  + **Operations/efficiency: e.g., pumping cost (energy/volume), duration of treatment, number of wells, monitoring frequency.**
* **Acceptance thresholds (example): e.g., ≥90% of contaminant mass removed within 10 years; prediction interval coverage of 95% intervals capturing true concentrations ≥ 90%; pumping cost < X $/m³; monitoring network cost reduction by ≥30% without compromising remediation success?**

# Software Architecture

XS Thoughts. This still reads like product/software plan in great details, but our deliverable if more like a framework design package (we should have high level things like: proposed framework, components, metrics, workflow, and implementation roadmap, and demonstrable recalibration loop). For this one I will reduce it to framework-design level, and put details into appendix

There are three key aspects to consider for this project:

1. Software Operational Capability
2. Software Audience and Usability
3. Software Maintenance and Development

The operational capability of the software refers to both the end-goal of creating a functional recalibration framework and also to the engineered tools working in-tandem to facilitate the framework’s functionality. Because many framework tools and algorithms have general applicability to scientific problems, it would be worthwhile to share these tools with the greater scientific community alongside the framework itself.

With a strong characterization in place to define what the uncertainty and recalibration framework must be able to calculate and how it will be able to do it, the primary consideration rests with who will use and operate the framework, and how they will do so. The target audiences must fall in-line with the scope of the deliverable for this project, and the software structure must be designed to facilitate meaningful access to those user groups.

For the maintenance and development of the project we will have to consider not only the first initial push of development of all of these tools, but how these tools and the framework will be potentially developed after the initial project scope. We will set-up guidelines to ensure that the final deliverable project and its sub-components have the necessary self-evaluation systems in-place to indicate to future researchers and engineers if specific software changes will affect the scientific or numerical integrity the generated results.

## Software Requirements

We will subdivide our potential userbase/stakeholders into two distinct categories (in decreasing technical ability):

1. Engineering Capable Users
   1. Machine Learning Engineers
   2. Software Engineers
   3. Data Scientists
2. Analysis Capable Users
   1. Analysts
   2. Administrators

These two classes represent the two distinct audiences for this software and their respective needs regarding the software itself.

#### Requirements for Engineering Users

These users are more interested in the technical aspects of the project framework. They will be interested in modifying and testing the framework for unique scientific modeling evaluation characteristics. These users will also be interested in using or evaluating the individual sub-components which allow the framework to function. Therefore, we must consider how we are able to provide the software meaningfully as a whole and individual tools.

**Requirements for Analysis Users**

For analysis capable users we must prioritize delivery and ingestion of complex information in a way that is palatable/easy to understand to those stakeholders. These users are not the intended audience for any code or programming related functionality. Rather, these users must simply be able to use the software and extract meaningful insights from it.

For this requirement, it is strongly recommended that there exists a small Graphic User Interface (GUI) component to the project. This GUI must serve the purpose of providing a method for users to interact with the software framework to upload new measurements and download

## System Architecture

The overall system and the majority of its components will be developed in contemporary **Python**, as this has been the language of choice for research and commercial development of numerical and deep learning modeling. If a major dependency is not developed in Python, we will provide Python-bindings for that dependency to ensure mutual compatibility in the software ecosystem.

Because the functionality of this project requires fundamental neural network model-level analysis and tooling, we will structure our project to use **NumPy** for CPU-bound computational tasks, and **Torch** with **PyTorch** bindings for GPU-bound tasks. These two numerical processing frameworks are mutually interoperable and will provide the greatest overlap between the targeted domain for the framework and the existing tooling of its active users.

To facilitate modularity in the overall software structure, we break up the scope of the architecture into three primary hierarchical levels.

1. Independent Numerical Processing Packages
2. The Monitoring and Recalibration Framework Package
3. The Operational Monitoring and Recalibration Software System

### Independent Software Packages and the M&R Software Package

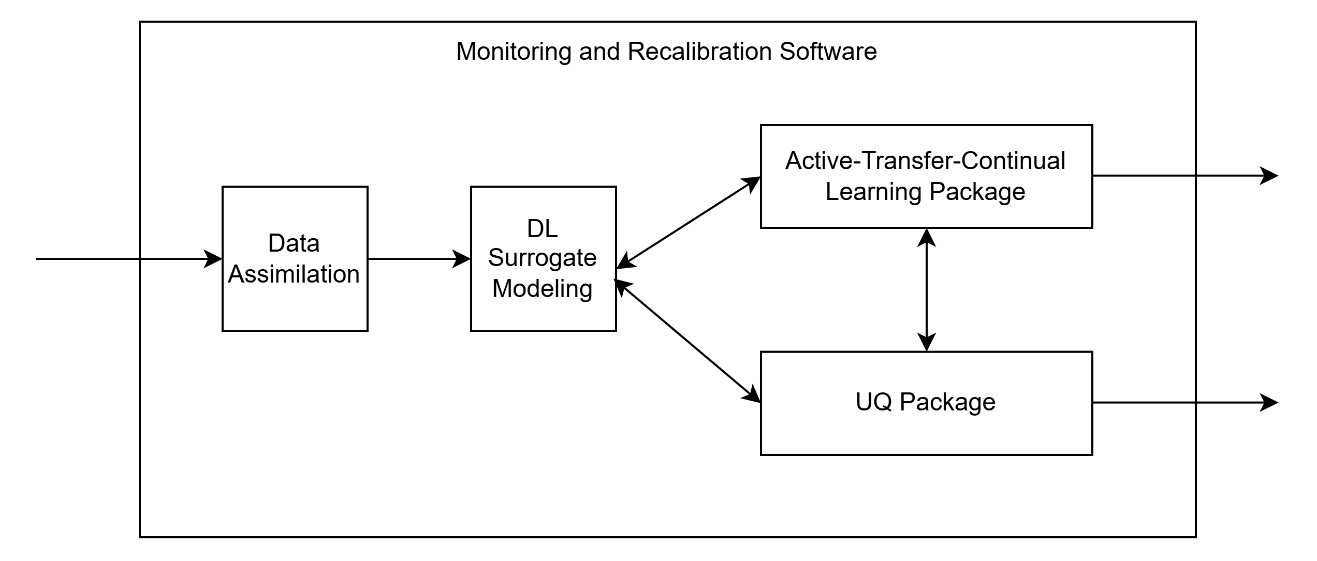
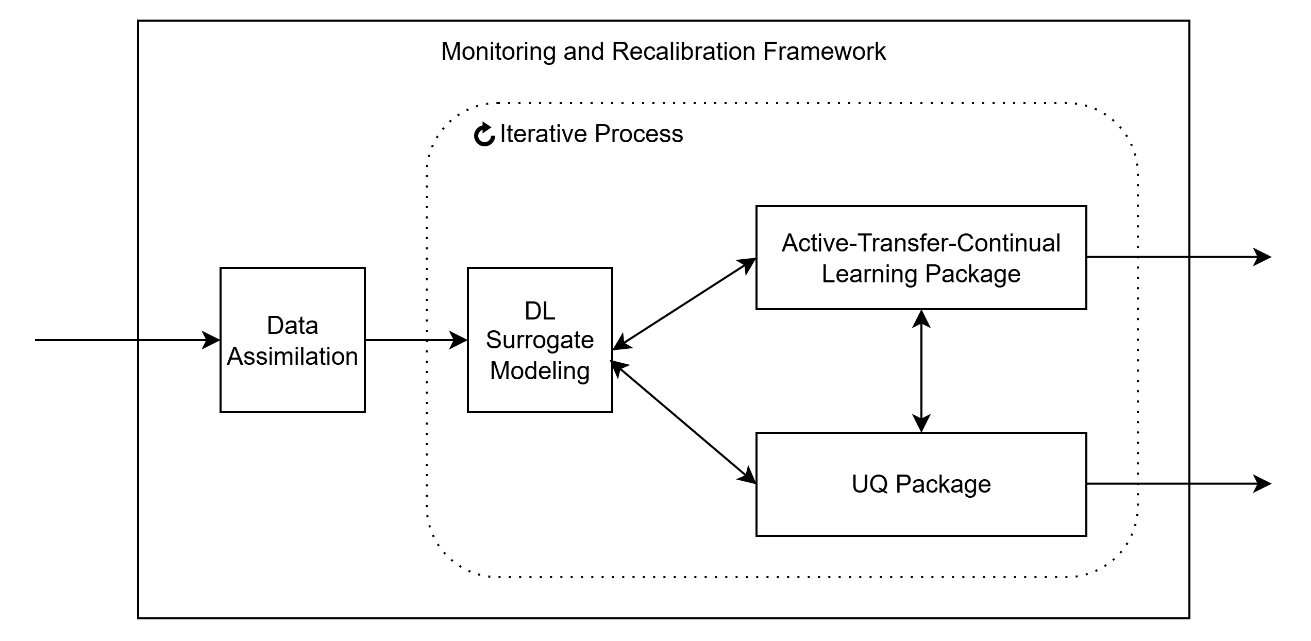
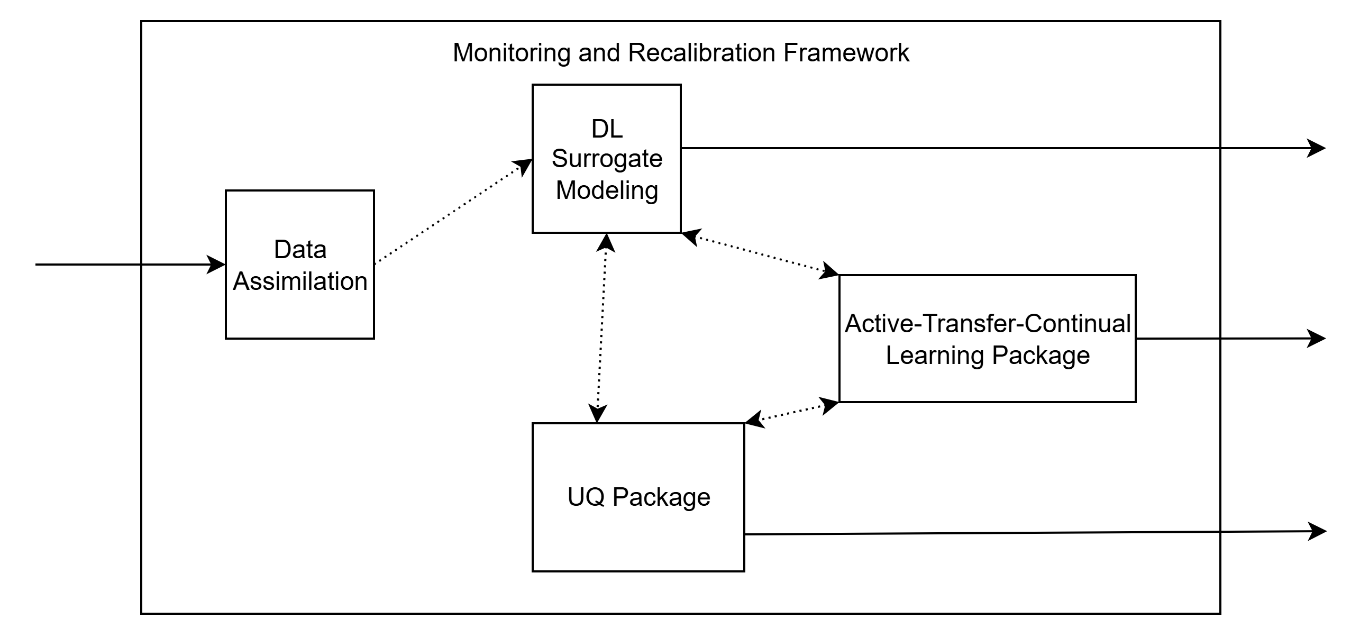
  

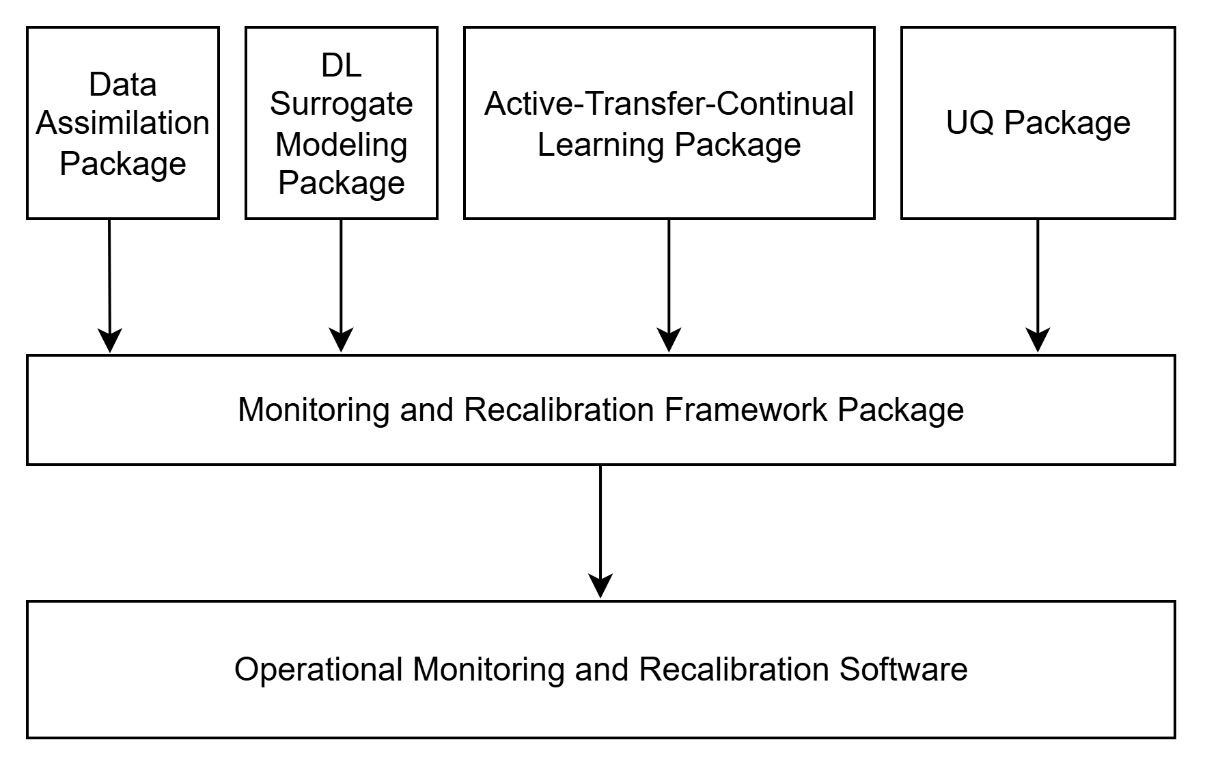
Figure 5 Framework composition of package interaction control flow graph.

This work will focus and encapsulation of independent subsystems and subcomponents. The purpose of this process is to standardize the interfaces between each subsystem, such that if a specific subsystem were to be replaced or modified then the framework will continue to function without needing to modify a connected component.

The process of maintaining standardized interfaces allows us to use subsystems as modular components. This modular aspect allows the components to be distributed and provided individually, and will also facilitate independent version control for separate development streams.

These modular components will be made available to the framework itself and to the greater scientific community as **Python packages** using the native Python package manager **pip**. This will allow the packages to be shared globally if we register the packages with the **PyPi** distribution service (i.e. to allow `pip install pnnl-uq-torch`), or installed locally by sharing the source code to be installed in the user’s environment (i.e. to allow `pip install -e ./path/to/source`).

As the relevant subsystems are developed and provided as packaged utilities, the overall framework will be built with those packages as enumerated dependencies. This compositional development will allow any changes upstream to subsystem functionality to be readily available downstream in the framework itself. Furthermore, this will allow the computational framework to be encapsulated as a Python package in itself because the core framework is the numerical tool under development. The systems for the orchestration of aggregation, distribution, and evaluation of long-term data will be handled by the On-line Operational Software System.



Diagram

AI-generated content may be incorrect.

A picture containing diagram

AI-generated content may be incorrect.

Figure 6 Software and Package Dependency Graph

### Operational System Architecture

With the numerical processing framework conceptualized, we now have the necessary components and usability requirements to develop a persistent and usable software system. Persistent software for this use is defined as software that maintains its own state throughout the intended interaction life time. In terms of operation of the Monitoring and Recalibration Software, this means defining an initial monitoring area and being able to upload new sets of measurements to which the software framework will respond by running its internal routines which update its state, reports on its current state, and becomes ready for further measurement ingestion. This persistent approach is contrary to typical single-shot experimental approaches in which the measurement dataset corpus is statically defined then iterated or sampled through independent configurations or dataset compositions.

The concept of persistence does not mean that the software provided must be continuously-online throughout its lifetime. Though such an always-online system is desirable for efficient time-to-use by end-users; such a system can be developed to be activated ad-hoc when a researcher has a new set of measurements to evaluate, just maintaining state between operational runs.

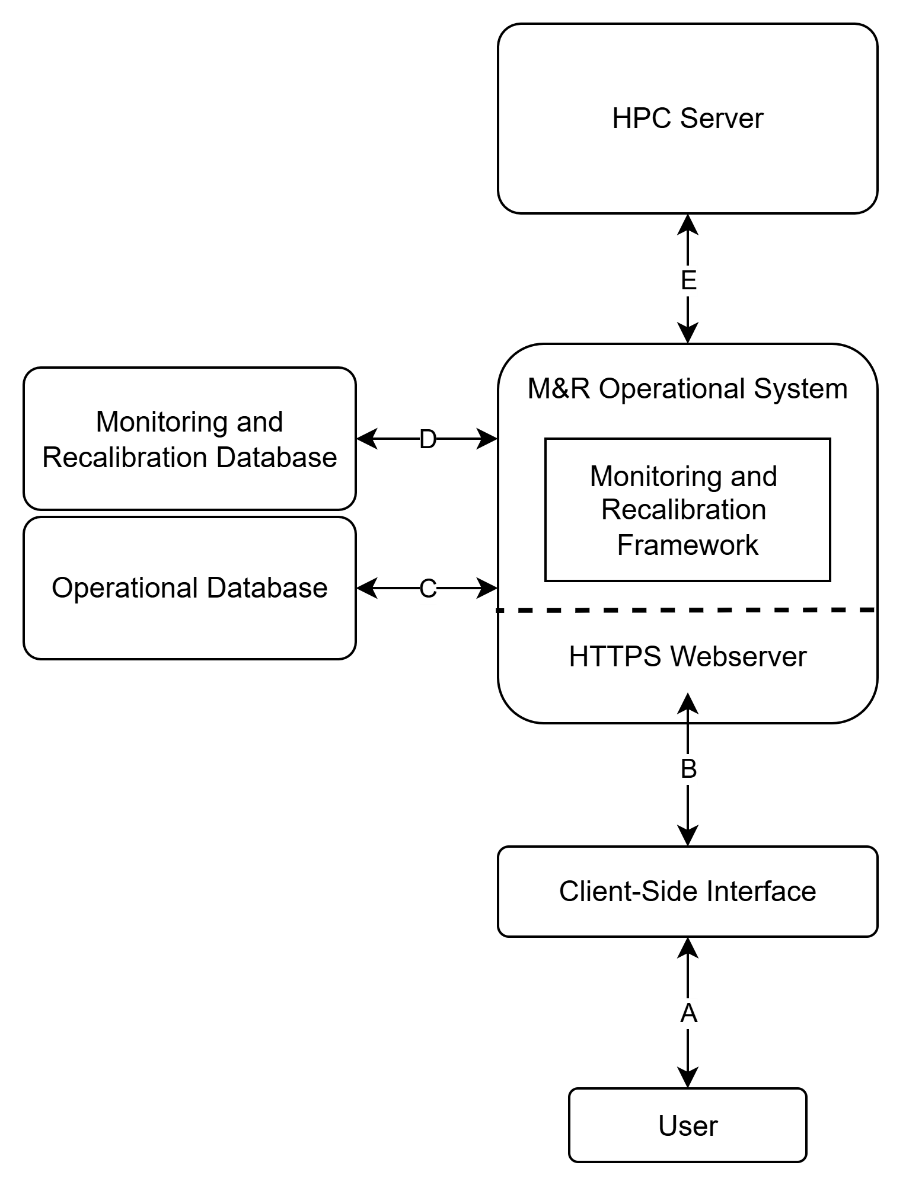


Figure 7: Control Flow Diagram for Waterlevel Monitoring and Recalibration System. User interacts with the GUI interface (A). The GUI acts as a visual representation for the state of the system and provides the ability to interact with the system (B). The monitoring and recalibration software (by schedule or manual initiation) upload new measurements from the client to the operational database (C), orchestrate processing in a generic HPC connection for model training and evaluation(E), then upload those results to the monitoring database (D).

A high-level diagram of the monitoring and calibration software framework is provided in Figure 1. The primary consideration here is that all boxed components must function as isolated services, independent of the activity or availability of the other services. This means that errors or failure in one subsystem do not affect the operation of other subsystems. It also means that each of these subsystems can be generically hosted on local and online resources, allowing for dynamic configuration of the system.

In this architecture, the Operational Monitoring and Recalibration Software provides user access via an HTTPS Webserver. The client-side interface (a GUI application) interacts with the software via the webserver and acts as a pure representation of the internal state of the application. The client-side application can be either a Web App or a Desktop GUI, depending on the desired course of development. The desktop application can be built using PySide6 upon the Qt framework, or as an Electron application (a brower+Web App desktop software bundle).

### Basic User Interface

## Software Development Guidelines

### Version Control

Version Control will be facilitated by using a commercial git vendor, which has more general capability than using open-source git in isolation. Version control will be applied independently to the individual software packages, as well as to the overall framework package. Independent versioning of the framework subsystems and of the framework itself will allow independent development to occur simultaneously.

The primary choices for development are **GitLab** and **GitHub**. GitHub provides an offsite software repository hosting service with privacy controls dictated by the subscribing organization, to which GitLab allows for organization-hosted software repositories with greater privacy and distribution control vested with the organization. Both of these options allow for public access and distribution once the framework has reached sufficient maturity and readiness, which means the open and public-facing packages will be available for greater community engagement in a structured and moderated environment.

### Testing and CI/CD

Continuous Integration and Continuous Distribution (CI/CD) is an industry development paradigm that focuses on being able to move the newest software modifications quickly to dependencies and end-users while ensuring stability of the changed software and prevents interruptions of service. For the software framework CI/CD is not particularly relevant for fast distribution, but the tools in its ecosystem will facilitate scientific integrity and numerical stability of the software.

The two main version control providers, GitHub and GitLab, provide a robust set of tools to evaluate software changes from static source code and its dynamic runtime. Specifically, utilizing the services’ git-actions allows each software package to define what needs to be compiled, evaluated, and regression tested. Such testing asserts and validates that the input-output interfaces remain consistent.

Because the software framework has a strong focus on encapsulation and modularization of the subsystem components as python packages, it becomes more straightforward to develop the necessary tests and evaluation routines for each package individually across known testing conditions and parameterizations for its own implementation. This way the Monitoring and Recalibration framework package does not need to test the individual subsystems it incorporated, rather just to evaluate well-enumerated tests of their connections as they are intended to interact within the framework.

### Operational Software Packaging and Distribution

## Not Considered

Using JAX as the GPU backend.

High-Level Graphical User Interface Tooling with Webserver and Webpage (in the appendix?)

### Framework

The fundamental purpose of the system is host GIS-based numerical and deep neural network (DNN) models, ingest new sensor measurements, evaluate and compare the quality of the modeling results, and be able to initiate model recalibration automatically and through manual initiation.

The key performance indicators (KPIs / metrics) for the modeling results must be easily viewable and explainable in-context to the users. In addition to being viewable, the model metrics will be used to trigger automatic calibration of model and model components. Those recalibrations must have a traceable history and meaningful annotations with respect to the metric(s) which resulted in recalibration.aComponent EnumerationaThis tool must be able to handle enumeration of multiple different DNN models, performance metrics, and model training procedures. This is important because the initial push of models developed for this task will not be the final iteration said models, nor their training schemes. Customers will also find it desirable to develop and utilize custom or external models for their use case.

With the expectation of change it is paramount that the Application Programming Interface (API) is consistent with multiple models, and the packaged models will use this interface to enforce compliance. Incompatibilities which arise in this type of system indicate insufficient generic model handling, which can be addressed during the development period.eWithout tying the software to a specific neural network framework (such as Torch, TensorFlow, JAX . . .) open standards like ONNX can facilitate model ingestion and computation. eThe same needs are exhibited and addressed by allowing a robust Metric API for facilitating internal and custom key performance metrics explaining data and modelling quality, as well as being used as recalibration triggers.

#### Hosting

Because the software system is designed for industrial applications, it requires significant computational resources to function. This means that it is unlikely that monolithic software running on a single device is desirable by organizations, and therefore must be designed with modular functional components. Specifically training DNN models and running numerical simulations require High Performance Computing resources

With that information we structure the software architecture to utilize three distinct hosting paradigms: hCloud Hosting via AWS, GCP, and AzurehOn-Premises Hosting

* Mixed-Hosting (On-Premises Components + Cloud)

Because these web-based technologies aim to solve internet engineering problems, they all share similar functionality and interfaces. The hosting problem can be solved by enumerating general compositors for the distinct possible configurable connections, which is made possible by the same IP/Web-Based protocol shared by these systems.

Because the system configuration at each host may be different, it is important to provide a system that is as mutually compatible as possible. With tools such as Docker, we are able solve the majority of this problem by creating “system images” of a fully functional instance of that software which can be distributed and activated on disparate technical systems. An example of such successful use-case is the Open-Source software CVAT, which is a self-hosted web-based image annotation system, which ships its operational software as a docker image. Software Usability For the usability component, we will subdivide our potential userbase/stakeholders into two distinct categories (in decreasing technical ability):

1. Engineering Capable Users
   1. Machine Learning Engineers
   2. Software Engineers
   3. Data Scientists
2. Analysis Capable Users
   1. Analysts
   2. Administrators

These two classes represent the two distinct audiences for this software and their respective needs regarding the software itself.

#### Requirements for Engineering Users

For engineering capable users, the most valuable information is well-stated documentation on the API endpoint and Custom Model/Metric development, as well as descriptive error handling to indicate whether new development errors arise from their implementations or from the software itself.

**Setup and Configuration Usability**

Setup and configuration usability should focus on providing a seamless experience for engineering users through detailed installation guides, step-by-step configuration tutorials, and prebuilt templates. The software should include system diagnostics tools to test compatibility with user hardware or hosting environments. Furthermore, default configurations should allow inexperienced users to easily deploy a minimally functional system, while advanced configuration options must be readily accessible for expmrt users. w

**Model and Metric Development Usability**

#### *For model and metric development usability, thorough documentation is key, outlining how custom DNN models and metrics can be integrated within the system. The development environment should include example scripts and templates to simplify implementation, alongside access to debugging tools that streamline the process of testing and validating custom developments.*

#### Requirements for Analysis Users

For analysis capable users we must prioritize delivery and ingestion of complex information in a way that is palatable/easy to understand to those stakeholders. This would form the bulk of the necessary GUI work for the software itself, which would come in the form of dashboards for the geographic area under monitoring.

**Necessary tools would be:**

1. Utility to upload new measurements
   1. individually and in bulk via CSV/Excel sheet
   2. error detection such as incorrect formatting
   3. validation of new sensors in measurement data.
2. Utility to export data and results to documents
   1. Modeling figures
   2. Measurement histories
   3. Modelling tables
3. Dashboard to view current modeling results
   1. Interact with modeling results
   2. Interact with modeling metrics
   3. Evaluate at a different time period
   4. Manually trigger recalibration
4. Dashboard to view measurement history
   1. View measurements and measurement locations
   2. View metrics on measurement history
5. Dashboard to view model comparisons
   1. Compare DNN models and numerical models
   2. View comparisons from different time periods

#### **UI and Dashboards for Client Side Interface**

### Software Maintenance

Software maintenance involves ensuring the adaptability and robustness of the program as both internal and external requirements evolve. As the software is changed to meet new requirements, it is important that there are no ill effects downstream of the updated or novel implementation. Doing so will help facilitate future development and maintain morale of the developers performing modifications. Developing a software test system to identify runtime issues that may not be evident in cursory testing (or that may be missed by out-of-scope quality assurance evaluation).tKey areas include implementing version control systems to track changes over time and automating regression testing to validate numerical reproducibility after every modification. Performing regression testing after software or model changes will indicate whether there are incorrect changes to model performance (such as a speed-performance increasing change which inadvertently changes the accuracy in the modeling results).t

### System Architecture

Diagram

AI-generated content may be incorrect.

Figure 8: Control Flow Diagram for Waterlevel Monitoring and Recalibration System. User interacts with the GUI interface (A). The GUI acts as a visual representation for the state of the system and provides the ability to interact with the system (B). The monitoring and recalibration software (by schedule or manual initiation) upload new measurements from the client to the operational database (C), orchestrate processing in a generic HPC connection for model training and evaluation(E), then upload those results to the monitoring database (D). tA high level diagram of the monitoring and calibration software framework is provided in Figure 1. The primary consideration here is that all boxed components must function as isolated services, independent of the activity or availability of the other services. This means that errors or failure in one subsystem does not affect the operation of other subsystems. It also means that each of these subsystems can be generically hosted on local and online resources, allowing for dynamic configuration of the system.

In this architecture, the Monitoring and Recalibration Software provides user access via an HTTPS Webserver. The client-side interface (a GUI application) interacts with the software via the webserver and acts as a pure representation of the internal state of the application. The client-side application can be either a Web App or a Desktop GUI, depending on the desired course of development. The desktop application can be built using PySide6 upon the Qt framework, or as an Electron application (a brower+Web App desktop software bundle).

# Prototype & Schedule

* **What the demo shows:**
  + Side-by-side physics vs surrogate.
  + Uncertainty bands.
  + Small batch of targeted physics runs.
  + Fine-tune.
  + Improved calibration
* **Architecture:** containerized pipeline (optional).
* **Milestones (need or not?)**
* **Roles**

Prototype Surrogate Architecture (Ross and Yilin)

This is the place that we can start to insert the method we used for the surrogate.

1) The FNO method for the 2D cross section model

2) U-Net Model for P&T

https://www.sciencedirect.com/science/article/pii/S0309170825001162

## U‑FNO Surrogate for 2D cross‑section Transport Prototype

For the prototype, we use a U‑Net Fourier Neural Operator (U‑FNO) surrogate to emulate a 2D PFLOTRAN flow and transport model on a fixed grid. The purpose is to keep PFLOTRAN as the authoritative simulator, but train a surrogate model that can adequately reproduce the spatiotemporal concentration fields in milliseconds of wall-clock time, so we can support rapid scenario testing and automated recalibration workflows.

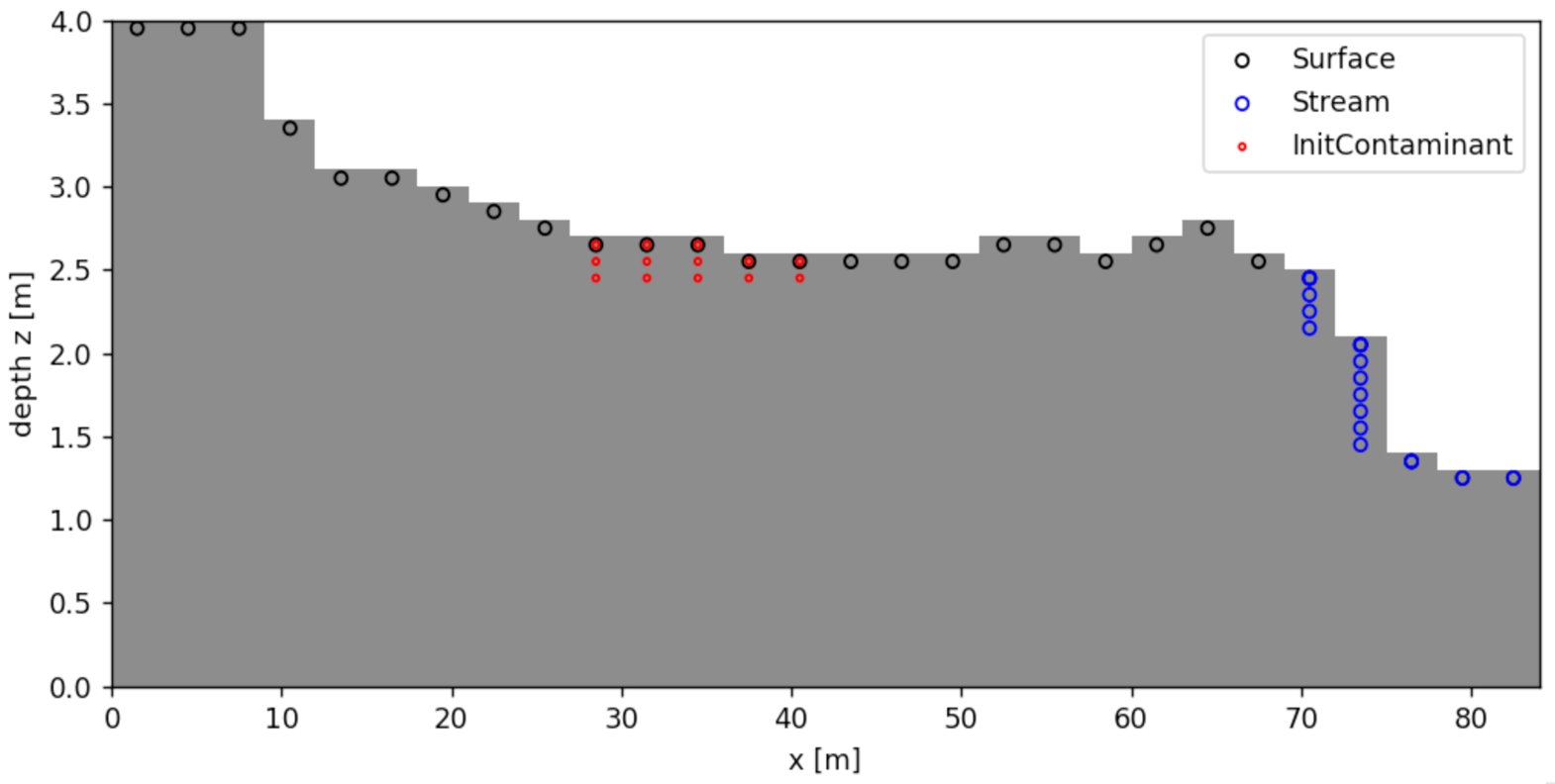
This U‑FNO approach is adapted from the GeoFUSE (Geological Forecasting using Surrogates and the Ensemble Smoother with Multiple Data Assimilation) workflow in Jiang et al. (2025), where the original application was seawater intrusion in a tidally influenced stream–floodplain cross section. In our prototype, we keep the same grid discretization and overall model configuration, but replaced salinity with a contaminant concentration field. Practically, the surrogate architecture and training pipeline remain the same. We focus on tracking the evolution of an initially contaminated plume as it responds to time-varying flow field induced by upland and stream water-level fluctuations prescribed at the boundary.

### PFLOTRAN Baseline Model

Jiang et al. (2025) define a 2D vertical cross section from upland to stream and simulate subsurface flow and transport using PFLOTRAN. PFLOTRAN solves variably saturated Richards equation and multicomponent reactive transport. The setup uses an 84 m × 4 m domain discretized on a 28 × 40 grid. Flow is driven by Dirichlet pressure boundary conditions on the upland and stream sides, with time‑varying stream stage updates (tidal influence) and time‑varying upland pressure updates. A seepage boundary is applied at the stream and a no‑flow boundary at the bottom.

For surrogate training, it is important that the simulator provides a consistent mapping from geologic properties to dynamic state trajectories across a fixed set of boundary conditions. Jiang et al. treat permeability and porosity as uncertain spatial fields and generate heterogeneous realizations. Porosity is assumed to share the same spatial distribution pattern as permeability, and Gaussian statistics are specified for both fields.

We adapt this method for the contaminant prototype. We keep the same grid and overall boundary condition structure but treat the transported scalar as a contaminant concentration rather than salinity. Recharge rate of 10 mm/yr is imposed on the ground surface. Contaminant is initially emplaced within the top 30 cm at the center of the domain. [Placeholder for the transport source term]...



Chart

AI-generated content may be incorrect.

### U‑FNO Surrogate and Training

The U‑FNO design is intended to capture both global, long‑range dependencies (handled efficiently in Fourier space), and local spatial structure (preserved by a U‑Net path embedded into the operator block). The combination of global and local features is especially relevant for transport problems where the solution contains both broad hydraulic controls and localized fronts or plume features.

Jiang et al. (2025) express the forward surrogate simulator as a mapping from the geological model to the dynamic states (pressure and transported scalar) over time. The surrogate approximates this mapping with a neural operator . In their implementation, the geological model input is repeated across all time steps to form the network input , then lifted to a higher dimensional representation via a learned linear and fully‑connected mapping, passed through multiple operator layers, and finally projected back to the output field.

Diagram

AI-generated content may be incorrect.

The architecture specification from Jiang et al. (2025) includes an input tensor that spans space and time, padding/depadding, three Fourier layers, three U‑Fourier layers, and linear projection layers. Training is done by minimizing the mismatch between PFLOTRAN outputs and surrogate predictions, including an additional weighting term that emphasizes cells that matter most in the modeled cross section.

Their optimization uses Adam, with a reported configuration that includes:

* ~126 million parameters,
* learning rate = 0.001 with stepwise decay,
* 200 epochs and batch size 4,
* and separate networks trained for pressure and salinity.

For our prototype, we keep this same training philosophy. The main practical differences are (1) the transported scalar (contaminant instead of salinity) and (2) the specific initial/boundary setup used to generate the PFLOTRAN training pairs.

Chart, histogram

AI-generated content may be incorrect.

Chart

AI-generated content may be incorrect.

## U‑Net Architecture for 2D Pump‑and‑Treat Plume Prediction

For the pump‑and‑treat (P&T) prototype, we use a U‑Net convolutional neural network as a fast surrogate to predict how a contaminant plume moves under different pumping configurations. The goal is to replace repeated, much slower physics‑model runs during screening with a surrogate that produces plume forecasts in milliseconds. The modified U-Net surrogate prototype takes gridded inputs describing the current plume and the P&T forcing, and it outputs the plume at the next time step. The approach below follows the 2D method introduced by Song et al. (2025).

A picture containing diagram

AI-generated content may be incorrect.

### 2D Training Data Generation

Song et al. (2025) demonstrate the method on a 2D synthetic confined‑aquifer plume migration problem intended to mimic a P&T setting. Key elements of the 2D setup are as below.

* Spatial grid: a 2D domain discretized on a 128 × 100 mesh with 1 m × 1 m cells.
* Initial plume: a circular plume placed near the center of the domain, with concentration highest at the center and decreasing toward the edges.
* P&T operation: 1-5 extraction wells placed within a central well field region, with pumping rates sampled across a range to generate diverse drawdown patterns and plume responses.
* Ensemble generation: 10,000 physics simulation runs to create a training library that spans different well configurations and pumping intensities.
* Output snapshots: plume fields are saved at several times (i.e, 0, 1, 2, 3, 4 days).

Chart

AI-generated content may be incorrect.

### Architectural Modifications for Improving Spatial Pattern Reconstruction

A practical challenge in P&T modeling is that well information is sparse. A few wells with coordinates and pumping rates must inform a full spatial plume map. Song et al. found that feeding raw well point information directly to a CNN can lead to unstable learning and less accurate plume predictions. The workaround is simple and effective for a prototype. Instead of giving the network only well locations and pumping rates, compute a continuous well drawdown map over the whole grid using a classic analytical well equation (i.e., Thiem equation) and superposition across all wells (Thiem, 1906). This drawdown map is not treated as a perfect flow solution due to the simplified assumptions. It is used as a physics‑inspired transform that spreads well influence across space in a physically meaningful way, which CNN learns well. In practice, each training sample includes two spatial channels:

* Plume concentration map at the previous time.
* A drawdown field computed from the current pumping configuration.

The standard U-Net structure down-samples to learn features at multiple scales, then up-samples while re‑injects fine spatial detail through skip connections. Our modified U-Net architecture implements the following modifications:

* Encoder-decoder with skip connections: preserves sharp spatial features such as plume fronts and edges while still learning broader plume-scale context.
* Strided convolutions: downsampling is performed using strided convolutions to better preserve spatial details than max pooling, which can discard information at plume edges.
* Bounded activation for concentration outputs: a customed Leaky Tanh is used so outputs remain bounded and transitions are smoother at plume edges. Negative concentrations predictions are clipped out in post-processing.

### Training Workflow

The training follows a standard supervised surrogate workflow:

* Dataset assembly: physics runs produce plume snapshots. Adjacent time pairs are converted into training samples.
* Data preparation: inputs and outputs are padded and normalized so the network trains stably and predictions can be compared consistently across runs.
* Train/validate/test split: data are split into separate subsets (70/15/15) to evaluate generalization to unseen well configurations.
* Loss and optimizer: model weights are learned using the Adam optimizer and a mean squared error (MSE) loss, which directly penalizes plume map differences.
* Early stopping: training stops when validation performance no longer improves, preventing overfitting and preserving the best checkpoint.

Once trained, the U‑Net is used in an autoregressive rollout for multiple time steps, where the predicted plume is fed back as the next input state and the process repeats.

1. Predict plume at time *t + Δt* from plume at *t* plus well‑impact map.
2. Feed that predicted plume forward as the next input state.
3. Repeat to generate a multi‑step plume forecast under a chosen P&T scenario.

This is the mechanism we will use in the prototype demo to show physical and surrogate models side‑by‑side under alternative pumping designs.

# Roadmap and Impact (Xuehang)

* **Operational impact:** faster scenario iteration; fewer unnecessary simulator runs; “explainable updates” to increase regulator confidence.
* **Scaling path:** additional constituents (CTET, TCE), other EM sites, alignment with digital-twin initiatives.
* **Open-source code:** code repo, containers, demo datasets,

# References

Anshuman, A., Eldho, T.I., 2023. A parallel workflow framework using encoder-decoder LSTMs for uncertainty quantification in contaminant source identification in groundwater. Journal of Hydrology 619, 129296.

Arshadi, M., De Paolis Kaluza, M.C., Miller, E.L., Abriola, L.M., 2020. Subsurface source zone characterization and uncertainty quantification using discriminative random fields. Water Resources Research 56, e2019WR026481.

Asher, M.J., Croke, B.F., Jakeman, A.J., Peeters, L.J., 2015. A review of surrogate models and their application to groundwater modeling. Water Resources Research 51, 5957-5973.

Bear, J., Cheng, A.H.-D., 2010. Modeling groundwater flow and contaminant transport. Springer.

Chen, Y., Liu, G., Huang, X., Chen, K., Hou, J., Zhou, J., 2021. Development of a surrogate method of groundwater modeling using gated recurrent unit to improve the efficiency of parameter auto-calibration and global sensitivity analysis. Journal of Hydrology 598, 125726.

Chen, Y., Liu, G., Huang, X., Meng, Y., 2022. Groundwater Remediation Design Underpinned By Coupling Evolution Algorithm With Deep Belief Network Surrogate. Water Resources Management 36, 2223-2239.

Doherty, J., 2004. PEST model-independent parameter estimation user manual. Watermark Numerical Computing, Brisbane, Australia 3338, 3349.

Du, J., Shi, X., Mo, S., Kang, X., Wu, J., 2022. Deep learning based optimization under uncertainty for surfactant-enhanced DNAPL remediation in highly heterogeneous aquifers. Journal of Hydrology 608, 127639.

Evensen, G., 2009. Data assimilation: the ensemble Kalman filter. Springer.

Fienen, M.N., Doherty, J., Hunt, R., Reeves, H., 2010. Using prediction uncertainty analysis to design hydrologic monitoring networks: Example applications from the Great Lakes water availability pilot project, Scientific Investigations Report.

Graham, G.H., Chen, Y., 2020. Bayesian inversion of generative models for geologic storage of carbon dioxide. arXiv preprint arXiv:2001.04829.

Guo, Z., Ma, R., Zhang, Y., Zheng, C., 2021. Contaminant transport in heterogeneous aquifers: A critical review of mechanisms and numerical methods of non-Fickian dispersion. Science China Earth Sciences 64, 1224-1241.

Hammond, G.E., Lichtner, P.C., Mills, R., 2014. Evaluating the performance of parallel subsurface simulators: An illustrative example with PFLOTRAN. Water resources research 50, 208-228.

Harbaugh, A.W., 2005. MODFLOW-2005, the US Geological Survey modular ground-water model: the ground-water flow process. US Department of the Interior, US Geological Survey Reston, VA, USA.

Hendricks Franssen, H.-J., Kinzelbach, W., 2008. Real‐time groundwater flow modeling with the ensemble Kalman filter: Joint estimation of states and parameters and the filter inbreeding problem. Water resources research 44.

Hill, M.C., Tiedeman, C.R., 2007. Effective groundwater model calibration: with analysis of data, sensitivities, predictions, and uncertainty. John Wiley & Sons.

Huling, S.G., 1991. Dense nonaqueous phase liquids. Superfund Technology Support Center for Ground Water, Robert S. Kerr ….

Jiang, S., Liu, C., Dwivedi, D., 2025. GeoFUSE: An Efficient Surrogate Model for Seawater Intrusion Prediction and Uncertainty Reduction. Water Resources Research 61, e2024WR038898.

Jiang, Z., Tahmasebi, P., Mao, Z., 2021. Deep residual U-net convolution neural networks with autoregressive strategy for fluid flow predictions in large-scale geosystems. Advances in Water Resources 150, 103878.

Johnson, V.M., Rogers, L.L., 2000. Accuracy of Neural Network Approximators in Simulation-Optimization. Journal of Water Resources Planning and Management 126, 48-56.

Ko, N.-Y., Lee, K.-K., Hyun, Y., 2005. Optimal groundwater remediation design of a pump and treat system considering clean-up time. Geosciences Journal 9, 23-31.

Kontos, Y.N., Kassandros, T., Perifanos, K., Karampasis, M., Katsifarakis, K.L., Karatzas, K., 2022. Machine learning for groundwater pollution source identification and monitoring network optimization. Neural Computing and Applications 34, 19515-19545.

Kovachki, N., Li, Z., Liu, B., Azizzadenesheli, K., Bhattacharya, K., Stuart, A., Anandkumar, A., 2023. Neural operator: Learning maps between function spaces with applications to pdes. Journal of Machine Learning Research 24, 1-97.

Laloy, E., Hérault, R., Jacques, D., Linde, N., 2018. Training-Image Based Geostatistical Inversion Using a Spatial Generative Adversarial Neural Network. Water Resources Research 54, 381-406.

Lauzon, D., 2024. A U-Net architecture as a surrogate model combined with a geostatistical spectral algorithm for transient groundwater flow inverse problems. Advances in Water Resources 189, 104726.

Li, X., Peng, C., Zhao, Y., Xia, X., 2025. A Hybrid DSCNN-GRU Based Surrogate Model for Transient Groundwater Flow Prediction. Applied Sciences 15, 4576.

Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., Anandkumar, A., 2020. Fourier neural operator for parametric partial differential equations. arXiv preprint arXiv:2010.08895.

Liu, Q., Gui, D., Zhang, L., Niu, J., Dai, H., Wei, G., Hu, B.X., 2022. Simulation of regional groundwater levels in arid regions using interpretable machine learning models. Science of The Total Environment 831, 154902.

Lu, L., Meng, X., Cai, S., Mao, Z., Goswami, S., Zhang, Z., Karniadakis, G.E., 2022. A comprehensive and fair comparison of two neural operators (with practical extensions) based on FAIR data. Computer Methods in Applied Mechanics and Engineering 393, 114778.

Luo, J., Ji, Y., Lu, W., Wang, H., 2018. Optimal Latin hypercube sampling-based surrogate model in NAPLs contaminated groundwater remediation optimization process. Water Supply 18, 333-346.

Luo, J., Lu, W., 2014. A mixed-integer non-linear programming with surrogate model for optimal remediation design of NAPLs contaminated aquifer. International Journal of Environment and Pollution 54, 1-16.

Luo, J., Lu, W., Xin, X., Chu, H., 2013. Surrogate model application to the identification of an optimal surfactant-enhanced aquifer remediation strategy for DNAPL-contaminated sites. Journal of Earth Science 24, 1023-1032.

Luo, J., Ma, X., Ji, Y., Li, X., Song, Z., Lu, W., 2023. Review of machine learning-based surrogate models of groundwater contaminant modeling. Environmental Research 238, 117268.

Majumder, P., Eldho, T.I., 2020. Artificial Neural Network and Grey Wolf Optimizer Based Surrogate Simulation-Optimization Model for Groundwater Remediation. Water Resources Management 34, 763-783.

Marrel, A., Iooss, B., 2024. Probabilistic surrogate modeling by Gaussian process: A review on recent insights in estimation and validation. Reliability Engineering & System Safety 247, 110094.

Maxwell, R., Condon, L., Kollet, S., 2015. A high-resolution simulation of groundwater and surface water over most of the continental US with the integrated hydrologic model ParFlow v3. Geoscientific model development 8, 923-937.

McCaulou, D.R., 1995. Nonaqueous phase liquids compatibility with materials used in well construction, sampling, and remediation. United States Environmental Protection Agency, Office of Research and ….

Meray, A., Wang, L., Kurihana, T., Mastilovic, I., Praveen, S., Xu, Z., Memarzadeh, M., Lavin, A., Wainwright, H., 2024. Physics-informed surrogate modeling for supporting climate resilience at groundwater contamination sites. Computers & Geosciences 183, 105508.

Mezić, I., 2013. Analysis of Fluid Flows via Spectral Properties of the Koopman Operator. Annual Review of Fluid Mechanics 45, 357-378.

Miele, R., Linde, N., 2025. Diffusion models for multivariate subsurface generation and efficient probabilistic inversion. Computers & Geosciences 207, 106076.

Mo, S., Zhu, Y., Zabaras, N., Shi, X., Wu, J., 2019. Deep Convolutional Encoder-Decoder Networks for Uncertainty Quantification of Dynamic Multiphase Flow in Heterogeneous Media. Water Resources Research 55, 703-728.

Mosser, L., Dubrule, O., Blunt, M.J., 2017. Reconstruction of three-dimensional porous media using generative adversarial neural networks. Physical Review E 96, 043309.

Neuman, S., Wierenga, P.J., Nicholson, T., 2003. A comprehensive strategy of hydrogeologic modeling and uncertainty analysis for nuclear facilities and sites. Division of Systems Analysis and Regulatory Effectiveness, Office of Nuclear ….

Newell, C.J., 1995. Light nonaqueous phase liquids. United States Environmental Protection Agency, Office of Research and ….

Nguyen, T.-U., Suk, H., Liang, C.-P., Ho, Y.-C., Chen, J.-S., 2025. Using Machine Learning to Develop a Surrogate Model for Simulating Multispecies Contaminant Transport in Groundwater. Hydrology 12, 185.

Pfaff, T., Fortunato, M., Sanchez-Gonzalez, A., Battaglia, P., 2020. Learning mesh-based simulation with graph networks, International conference on learning representations.

Pruess, K., Oldenburg, C.M., Moridis, G., 1999. TOUGH2 user's guide version 2.

RMCS, 2017. Hanford 200 Area ZP-1 OU, Washington

Rogers, L.L., Dowla, F.U., 1994. Optimization of groundwater remediation using artificial neural networks with parallel solute transport modeling. Water Resources Research 30, 457-481.

Rogers, L.L., Dowla, F.U., Johnson, V.M., 1995. Optimal Field-Scale Groundwater Remediation Using Neural Networks and the Genetic Algorithm. Environmental Science & Technology 29, 1145-1155.

Rubin, Y., 2003. Applied stochastic hydrogeology. Oxford University Press.

Sanchez-Gonzalez, A., Godwin, J., Pfaff, T., Ying, R., Leskovec, J., Battaglia, P., 2020. Learning to simulate complex physics with graph networks, International conference on machine learning. PMLR, pp. 8459-8468.

Schmid, P.J., 2010. Dynamic mode decomposition of numerical and experimental data. Journal of Fluid Mechanics 656, 5-28.

Sethi, R., Di Molfetta, A., 2019. Mechanisms of Contaminant Transport in Aquifers, in: Sethi, R., Di Molfetta, A. (Eds.), Groundwater Engineering : A Technical Approach to Hydrogeology, Contaminant Transport and Groundwater Remediation. Springer International Publishing, Cham, pp. 193-217.

Shams, R., Alimohammadi, S., Yazdi, J., 2021. Optimizing surfactant-enhanced aquifer remediation based on Gaussian process surrogate model in DNAPL-contaminated sites considering different wells patterns. Groundwater for Sustainable Development 15, 100675.

Siade, A.J., Cui, T., Karelse, R.N., Hampton, C., 2020. Reduced‐dimensional Gaussian process machine learning for groundwater allocation planning using swarm theory. Water Resources Research 56, e2019WR026061.

Šimůnek, J., van Genuchten, M.T., 2016. Contaminant transport in the unsaturated zone: Theory and modeling, The handbook of groundwater engineering. CRC Press, pp. 221-254.

Šimůnek, J., Van Genuchten, M.T., Šejna, M., 2016. Recent developments and applications of the HYDRUS computer software packages. Vadose zone journal 15, vzj2016. 2004.0033.

Song, X., Demirkanli, I., Hou, Z., Lin, X., Karanovic, M., Tonkin, M., Appriou, D., Mackley, R., 2025. Integrating analytical solutions and U-Net model for predicting groundwater contaminant plumes in pump-and-treat systems. Advances in Water Resources 202, 105002.

Taccari, M.L., Nuttall, J., Chen, X., Wang, H., Minnema, B., Jimack, P.K., 2022. Attention U-Net as a surrogate model for groundwater prediction. Advances in Water Resources 163, 104169.

Taccari, M.L., Wang, H., Goswami, S., Florio, M.D., Nuttall, J., Chen, X., Jimack, P.K., 2024. Developing a cost-effective emulator for groundwater flow modeling using deep neural operators. Journal of Hydrology 630, 130551.

Tang, M., Liu, Y., Durlofsky, L.J., 2020. A deep-learning-based surrogate model for data assimilation in dynamic subsurface flow problems. Journal of Computational Physics 413, 109456.

Thiem, G., 1906. Hydrologische Methoden. J. M. Gebhardt's Verlag.

Vali, M., Zare, M., Razavi, S., 2021. Automatic clustering-based surrogate-assisted genetic algorithm for groundwater remediation system design. Journal of Hydrology 598, 125752.

Wang, K., Sun, W., 2018. A multiscale multi-permeability poroplasticity model linked by recursive homogenizations and deep learning. Computer Methods in Applied Mechanics and Engineering 334, 337-380.

Wang, L., Kurihana, T., Meray, A., Mastilovic, I., Praveen, S., Xu, Z., Memarzadeh, M., Lavin, A., Wainwright, H., 2022. Multi-scale Digital Twin: Developing a fast and physics-informed surrogate model for groundwater contamination with uncertain climate models. arXiv preprint arXiv:2211.10884.

Wang, N., Chen, Y., Zhang, D., 2025a. A comprehensive review of physics-informed deep learning and its applications in geoenergy development. The Innovation Energy 2, 100087-100081-100087-100015.

Wang, Z., Chen, Y., Fu, W., Du, M., Chen, G., Ma, X., Zhang, D., 2025b. Generative inverse modeling for improved geological CO2 storage prediction via conditional diffusion models. Applied Energy 395, 126071.

Wang, Z., Chen, Y., Wang, N., Chen, G., Zhang, D., 2025c. Generative subsurface flow modeling with pretrained diffusion model and training‐free knowledge alignment. Geophysical Research Letters 52, e2025GL118000.

Wen, G., Li, Z., Azizzadenesheli, K., Anandkumar, A., Benson, S.M., 2022. U-FNO—An enhanced Fourier neural operator-based deep-learning model for multiphase flow. Advances in Water Resources 163, 104180.

Wen, G., Li, Z., Long, Q., Azizzadenesheli, K., Anandkumar, A., Benson, S.M., 2023. Real-time high-resolution CO 2 geological storage prediction using nested Fourier neural operators. Energy & Environmental Science 16, 1732-1741.

Werner, A.D., Bakker, M., Post, V.E.A., Vandenbohede, A., Lu, C., Ataie-Ashtiani, B., Simmons, C.T., Barry, D.A., 2013. Seawater intrusion processes, investigation and management: Recent advances and future challenges. Advances in Water Resources 51, 3-26.

White, M.D., Oostrom, M., 2003. STOMP subsurface transport over multiple phases version 3.0 User's guide. Pacific Northwest National Lab., Richland, WA (US).

Xie, X., Zhang, X., 2024. Development of a deep surrogate model with spatiotemporal characteristics mining capabilities for the prediction of groundwater level in coastal areas. Journal of Environmental Management 370, 122724.

Xu, R., Zhang, D., 2024. Forward prediction and surrogate modeling for subsurface hydrology: A review of theory-guided machine-learning approaches. Computers & Geosciences 188, 105611.

Xu, Z., Hu, B.X., Xu, Z., Wu, X., 2019. Simulating seawater intrusion in a complex coastal karst aquifer using an improved variable-density flow and solute transport–conduit flow process model. Hydrogeology Journal 27, 1277-1289.

Yan, S., Minsker, B., 2006. Optimal groundwater remediation design using an adaptive neural network genetic algorithm. Water Resources Research 42.

Yan, S., Minsker, B., 2011. Applying Dynamic Surrogate Models in Noisy Genetic Algorithms to Optimize Groundwater Remediation Designs. Journal of Water Resources Planning and Management 137, 284-292.

Yang, L., Wang, X., Mendoza-Sanchez, I., Abriola, L.M., 2018. Modeling the influence of coupled mass transfer processes on mass flux downgradient of heterogeneous DNAPL source zones. Journal of Contaminant Hydrology 211, 1-14.

Yang, M., Annable, M.D., Jawitz, J.W., 2016. Solute source depletion control of forward and back diffusion through low-permeability zones. Journal of Contaminant Hydrology 193, 54-62.

Yang, M., Annable, M.D., Jawitz, J.W., 2017. Forward and back diffusion through argillaceous formations. Water Resources Research 53, 4514-4523.

Yazdi, S.H., Robati, M., Samani, S., Hargalani, F.Z., 2025. Prediction of two groundwater sustainability indicators in semi-arid aquifers using machine learning. Environmental Earth Sciences 84, 294.

Zhang, Y.-F., Thorburn, P.J., 2022. A deep surrogate model with spatio-temporal awareness for water quality sensor measurement. Expert Systems with Applications 200, 116914.

Zheng, C., Bennett, G.D., 2002. Applied contaminant transport modeling. Wiley-Interscience New York.

Zheng, C., Wang, P.P., 1999. MT3DMS: a modular three-dimensional multispecies transport model for simulation of advection, dispersion, and chemical reactions of contaminants in groundwater systems; documentation and user's guide.

Zhu, Y., Zabaras, N., 2018. Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification. Journal of Computational Physics 366, 415-447.

Zhu, Y., Zabaras, N., Koutsourelakis, P.-S., Perdikaris, P., 2019. Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. Journal of Computational Physics 394, 56-81.

# Appendix

To anchor our framework in established practice, we reviewed recent work on physics-trained surrogates and their use in design, forecasting, and uncertainty reduction. Across groundwater contamination studies, surrogate models are used to cut the cost of repeated forward runs for design, inference, and uncertainty analysis. Reviews agree that Latin hypercube sampling is the most common way to generate training or design points, and that artificial neural networks are among the most widely used surrogate families (Luo et al., 2023). No single method dominates across problems, so fit to task and data should guide model choice. This aligns with our plan to generate synthetic training data with a physics model and then train a focused surrogate that predicts the state variables we care about.

Take-aways from recent papers \_

* Physics‑trained surrogates can replace expensive flow and transport in design loops.  
  Majumder and Eldho (2020) built an ANN that emulates an AEM–RWPT transport simulator and coupled it with a Grey Wolf Optimizer for pump‑and‑treat design. They also used kernel density estimation to turn finite particles into smooth concentrations. The surrogate met accuracy targets and made optimization practical.
* Local, cluster‑wise surrogates can help tough optimization landscapes.  
  Vali et al. (2021) proposed an automatic clustering approach that trains multiple local ANN surrogates inside a genetic algorithm. On a remediation case, this reduced forward calls by at least sixty percent while meeting concentration and drawdown constraints.
* Structured learning for reactive transport is feasible.  
  Nguyen et al. (2025) trained feed‑forward ANNs to emulate one‑dimensional multispecies reactive transport. Inputs are Peclet number, decay and retardation factors, space, and time. Outputs are the parent and daughter concentration fields. Once trained, the model reproduces spatial–temporal profiles with large speedups over finite difference runs.
* Probabilistic graphical surrogates support source‑zone UQ.  
  Arshadi et al. (2020) used a discriminative random field model learned from a limited set of field‑scale simulations. The model produces conditional realizations of saturation and concentration that match borehole data and return uncertainty metrics used in decision support.
* End‑to‑end frameworks for state prediction and UQ are emerging.  
  Jiang et al. (2025) introduced GeoFUSE, which couples a U‑Net Fourier Neural Operator surrogate with PCA for geology and ESMDA for data assimilation. On a PFLOTRAN coastal case they report orders‑of‑magnitude speedups and uncertainty reduction while keeping predictive accuracy. Their workflow is reproducible and includes shared code, which we will adapt for synthetic data generation in our project.

Guidance from recent review publications  
Two recent reviews are especially relevant to how we structure the model and training. (Luo et al., 2023) survey 120 surrogate papers in groundwater contaminant modeling and highlight practical choices that echo our table fields: inputs and outputs, sampling plans, dependence handling, model class, and validation. They also note the growing role of deep models in spatiotemporal prediction.   
Complementing that, (Xu and Zhang, 2024) and (Wang et al., 2025a) review “theory‑guided” and physics‑informed deep learning. They outline three ways to embed physics in training: strong‑form PDE residuals with automatic differentiation, weak‑form residuals that help with discontinuities, and discretized‑form residuals that pair naturally with CNNs or operators. Their take‑home message is to pick the physics coupling that matches the data and the numerics of the base simulator. This framing helps us decide when to stay purely data‑driven and when to add soft physics constraints.

Table A1. Machine Learning Based Surrogates Paper Review

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Application | Inputs | Outputs | Sampling methods | Sampling marginals | Sampling dependence | Surrogate models | Software packages | Reference |
| Remediation design | 3D images of k field, initial DNAPL saturation and well locations, injection /extraction rates encoded on the grid. | 3D remaining DNAPL saturation field, cost f1 and remaining area f2, and risk map. | Rejection sampling to condition on sparse borehole data. | ln k assumed lognormal, pumping rates are ranges only. | Spatial dependence in ln k via correlation length and SIP physics. Corss-dependence between k and S\_N0 enforced by RS conditioning. | CNN (U-Net-like) with Residual-in-Residual Dense Blocks (RRDB) to stabilize training. | UTCHEM for forward Surfactant-enhanced aquifer remediation (SEAR) physics. | (Du et al., 2022) |
| Remediation design | 4 PAT well locations (x,y), extraction rates (m3/d), pumping durations (days), and install flag(Xu and Zhang, 2024). | Maximum chloride concentrations at 3 observation wells | Adaptive (PSO-guided, surrogate in the loop) | Values drawn from bounds. | None stated. Discrete choices enforced by thresholding (Wang et al., 2025a) and grid-snapping (x,y) | Deep belief network (stacking Restricted Boltzmann Machines), adaptive retraining during PSO. | MODFLOW (flow) and MT3DMS (transport) | (Chen et al., 2022) |
| Remediation design | Pumping rates (m3/d) for 6 wells for 5 yrs | Plume control well limits: mostly 0.02 g/m3, five most downstream 0.005g/m3, performance well limit 0.005 g/m3 at horizon. Cost. | Adaptive | Decision variable sampled within bounds only. | Not stated. | Feedforward ANN per cluster. One ANN predicts cost, another predicts total violation. | MODFLOW-2000 for flow and MT3DMS for transport. GA optimizer | (Vali et al., 2021) |
| Remediation design | Vector of pumping rate for 5 wells (Case 1: 0-1500 m3/d; Case 2: 0-500 m3/d). | Heads at observation wells and concentrations at monitoring points. | Random pumping rates within bounds. | Pump rates sampled uniformly within bounds. | None. Coupling arises via physics and constraints. | Feedforward neural network with 2 hidden layers, error function is minimized using Levenberg-Marquardt backpropagation algorithm. | Custom Analytic element method (AEM) and Random walk particle tracking (RWPT) and Kernel density estimator (KDE) post-processing to simulate flow and transport and concentrations. | (Majumder and Eldho, 2020) |
| Forward simulation of multispecies reactive transport (1D, homogeneous) | Parent C1: Groundwater seepage velocity and hydrodynamic dispersion coefficient (Peclet number, Pe), attenuation factors for parent (R1), degradation constants (λ₁) for each contaminants, normalized position and time x [0,1], T [0,5]; Daughter C2: Pe, R1, R2, λ₁, λ2, C1, X, and T. | Aqueous concentration of each contaminant across space and time. C1(x,t) and C2(x,t). Reported via errors (MAE/MSE/RMSE) at selected X/T. | Uniform grid based random Monte Carlo subset. | Samples uniformly taken within the min-max ranges. | None. Inputs varied independently. | Feed-forward ANN (two MLPs), 4 hidden layers for parent C1, 6 layers for daughter C2, ReLU activations, Adam optimizer. | Custom FD code. | (Nguyen et al., 2025) |
| NAPL source identification & UQ using physics-trained ML | Permeability field, borehole measurements of NAPL saturation and concentration at several times, per pixel: distances to nearest up/down-gradient boreholes, log k, vertical log k. | 2D fields of saturation and concentrations, uncertainty maps, metrics for total NAPL, total aqueous mass, pool fraction, centers. | Inference sampling: Metropolis-Hastings to draw conditional realizations, when k is uncertain, geostatistical ensemble of permeability then average over them. | No fitted site PDFs for inputs. After classification, dequantization draws from PDFs fitted to category histograms. | Spatial dependence modeled in the DRF via pairwise potentials and shared features. Saturation and concentration are coupled via shared features. | Discriminative Random Field (DRF), trained by psedolikelihood, realizations via Metropolis-Hastings. | M-VALOR for DNAPL infiltration, MT3DMS for transport. | (Arshadi et al., 2020) |
| Forward simulation of seawater intrusion | 2-D heterogeneous log-permeability, porosity fields (Gaussian), boundary time series including stream and upland heads, seepage, no-flow bottom. Monthly observations for 3 wells for 5 years. | Temporal cross sections of pressure and salinity fields. posterior permeability (mean/std) | Prior geomodels from sequential Gaussian simulation, PCA and latent variables to reduce parameters from 1120 to 642, ESMDA to update latent variables (Nr​=400, Na=4, αk=4). | log-k using Gaussian (μ=4.5, σ=1); porosity using Gaussian (μ=0.5, σ=0.05) | Porosity shares spatial pattern with k, PCA basis Φ preserves spatial correlation, assimilation uses cross-covariances | U-FNO trained separately for pressure and salinity | PFLOTRAN for flow and transport, GeoFUSE codebase (U-FNO, ESMDA) | (Jiang et al., 2025) |
| Review papers on source ID, remediation design, coastal aquifers, UQ, monitoring design, parameter inversion | Typical inputs include perm fields, source location, release history, pumping schedule, boundary conditions, and monitoring layouts. | Typical outputs include concentrations, pressure fields, breakthrough curves, error metrics, optimal pumping placement, uncertainty. | LHS, adaptive sampling | Typically uniform, physically informed ranges, Gaussian priors and PCA for high-dimensional fields | Spatial correlation in perm, temporal coupling in time-series data | CNN, U-Net, RNN/LSTM, GAN, | MODFLOW family, PFLOTRAN, TOUGH, | (Luo et al., 2023) (Xu and Zhang, 2024) (Luo et al., 2013; Wang et al., 2025a) |

Table A2. Curated set of representative surrogate-modeling publications for groundwater and subsurface transport, with primary classifications used for **Figure 2-4**.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Year** | **Primary surrogate family** | **Application family** | **Project domain** | **Reference** |
| 1994 | Regression & classical surrogates | Design & optimization | Contaminant hydrogeology / remediation | (Rogers and Dowla, 1994) |
| 1995 | Regression & classical surrogates | Design & optimization | Contaminant hydrogeology / remediation | (Rogers et al., 1995) |
| 2000 | Regression & classical surrogates | Design & optimization | Contaminant hydrogeology / remediation | (Johnson and Rogers, 2000) |
| 2006 | Regression & classical surrogates | Design & optimization | Contaminant hydrogeology / remediation | (Yan and Minsker, 2006) |
| 2011 | Regression & classical surrogates | Design & optimization | Contaminant hydrogeology / remediation | (Yan and Minsker, 2011) |
| 2013 | Regression & classical surrogates | Design & optimization | Contaminant hydrogeology / remediation | (Luo et al., 2013) |
| 2014 | Regression & classical surrogates | Design & optimization | Contaminant hydrogeology / remediation | (Luo and Lu, 2014) |
| 2017 | Generative models | UQ & ensemble generation | Groundwater flow / fluid mechanics | (Mosser et al., 2017) |
| 2018 | Regression & classical surrogates | Design & optimization | Contaminant hydrogeology / remediation | (Luo et al., 2018) |
| 2018 | Regression & classical surrogates | Forward emulation | Groundwater flow / fluid mechanics | (Wang and Sun, 2018) |
| 2018 | Conv encoder-decoder | UQ & ensemble generation | General scientific computing | (Zhu and Zabaras, 2018) |
| 2018 | Generative models | UQ & ensemble generation | Contaminant hydrogeology / remediation | (Yang et al., 2018) |
| 2018 | Generative models | Inverse & calibration | Groundwater flow / fluid mechanics | (Laloy et al., 2018) |
| 2019 | Conv encoder-decoder | UQ & ensemble generation | Groundwater flow / fluid mechanics | (Zhu et al., 2019) |
| 2019 | Conv encoder-decoder | UQ & ensemble generation | CO2 storage / geoenergy | (Mo et al., 2019) |
| 2020 | Operator-learning | Forward emulation | General scientific computing | (Li et al., 2020) |
| 2020 | Regression & classical surrogates | Design & optimization | Groundwater flow / fluid mechanics | (Siade et al., 2020) |
| 2020 | Regression & classical surrogates | Design & optimization | Contaminant hydrogeology / remediation | (Majumder and Eldho, 2020; Tang et al., 2020) |
| 2020 | Conv encoder-decoder | Inverse & calibration | General scientific computing | (Tang et al., 2020) |
| 2020 | Generative models | Inverse & calibration | CO2 storage / geoenergy | (Graham and Chen, 2020) |
| 2021 | Regression & classical surrogates | Design & optimization | Contaminant hydrogeology / remediation | (Shams et al., 2021) |
| 2021 | Regression & classical surrogates | Design & optimization | Contaminant hydrogeology / remediation | (Vali et al., 2021) |
| 2021 | Conv encoder-decoder | Forward emulation | Contaminant hydrogeology / remediation | (Jiang et al., 2021) |
| 2021 | Sequence models | Inverse & calibration | Groundwater flow / fluid mechanics | (Chen et al., 2021) |
| 2022 | Conv encoder-decoder | Design & optimization | Contaminant hydrogeology / remediation | (Du et al., 2022) |
| 2022 | Regression & classical surrogates | Design & optimization | Contaminant hydrogeology / remediation | (Chen et al., 2022) |
| 2022 | Conv encoder-decoder | Forward emulation | Groundwater flow / fluid mechanics | (Taccari et al., 2022) |
| 2022 | Operator-learning | Forward emulation | General scientific computing | (Lu et al., 2022) |
| 2022 | Operator-learning | Forward emulation | General scientific computing | (Wang et al., 2022) |
| 2022 | Operator-learning | Forward emulation | CO2 storage / geoenergy | (Wen et al., 2022) |
| 2022 | Conv encoder-decoder | Inverse & calibration | Groundwater flow / fluid mechanics | (Kontos et al., 2022) |
| 2022 | Conv encoder-decoder | Monitoring & time-series prediction | Groundwater monitoring & forecasting | (Liu et al., 2022) |
| 2022 | Sequence models | Monitoring & time-series prediction | Groundwater monitoring & forecasting | (Zhang and Thorburn, 2022) |
| 2023 | Operator-learning | Forward emulation | General scientific computing | (Kovachki et al., 2023) |
| 2023 | Operator-learning | Forward emulation | CO2 storage / geoenergy | (Wen et al., 2023) |
| 2023 | Conv encoder-decoder | Inverse & calibration | Contaminant hydrogeology / remediation | (Anshuman and Eldho, 2023) |
| 2024 | Operator-learning | Forward emulation | Contaminant hydrogeology / remediation | (Meray et al., 2024) |
| 2024 | Conv encoder-decoder | Inverse & calibration | Groundwater flow / fluid mechanics | (Lauzon, 2024) |
| 2024 | Operator-learning | Inverse & calibration | Groundwater flow / fluid mechanics | (Taccari et al., 2024) |
| 2024 | Sequence models | Monitoring & time-series prediction | Coastal aquifers / seawater intrusion | (Xie and Zhang, 2024) |
| 2025 | Sequence models | Forward emulation | Groundwater flow / fluid mechanics | (Li et al., 2025) |
| 2025 | Regression & classical surrogates | Forward emulation | Contaminant hydrogeology / remediation | (Nguyen et al., 2025) |
| 2025 | Conv encoder-decoder | Inverse & calibration | CO2 storage / geoenergy | (Wang et al., 2025b) |
| 2025 | Generative models | Inverse & calibration | General scientific computing | (Miele and Linde, 2025) |
| 2025 | Operator-learning | Inverse & calibration | Coastal aquifers / seawater intrusion | (Jiang et al., 2025) |
| 2025 | Regression & classical surrogates | Monitoring & time-series prediction | Contaminant hydrogeology / remediation | (Yazdi et al., 2025) |
| 2025 | Generative models | UQ & ensemble generation | Groundwater flow / fluid mechanics | (Wang et al., 2025c) |

Question







# Concepts for High-Fidelity Modeling and Calibration User-Facing Software Interface

Graphical user interface

AI-generated content may be incorrect.

Figure 9: Modeling control panel and landing screen. Buttons transition to dashboards and deeper configuration applications.

Diagram

AI-generated content may be incorrect.

Figure 10: Dashboard for interactive modeling metrics and modeling quality view. Metrics can provide coloring based on their state (good, warning, requires recalibration)

A picture containing chart

AI-generated content may be incorrect.

Figure 11: Dashboard for viewing sensor measurement histories.

**Chart

AI-generated content may be incorrect.**

Figure 12: Dashboard for modeling comparisons.

**Super high-level notes**

Motivation:

* Long-term forecasts will inevitably drift and become useless

Scheduled performance checks and automated recalibration keep error and bias within predefined thresholds, sustaining reliability over multi-year remediation timelines.

The deliverable:

* Framework Design Package – an executive-level document that lays out the proposed framework, components, metrics, and implementation roadmap.
* Showing one full recalibration loop (drift detection → targeted physics runs → fine-tuning) and an interactive dashboard that DOE managers can test-drive (optional).
* Also think about monthly/quarterly updates that we can provide.
* We will have weekly updates (should we keep a confluence page or word documents)

Technical contribution/contents:

* Develop a clear, modular framework: key pillars (Modules) with well-specified subcomponents)
* Illustrate workflow: Use diagrams/flowcharts to show how components interact and when recalibration is triggered.
* Provide concise method summaries: Survey surrogate types, UQ techniques, active-learning strategies, and continual-learning methods (maybe in a comparative table).
* What is the best practice
* Outline a proof-of-concept demonstration plan
* Prototype (illustrative and synthetic)

**Plans/scopes/roles**

|  |  |  |  |
| --- | --- | --- | --- |
| Framework Pillar / Workstream | Lead | Core Responsibilities | Key Outputs |
| Uncertainty Quantification | Jason Hou, TC | - Predictive UQ (Select & implement MC-Dropout ± Deep Ensembles)  - Trigger policy for recalibration | - UQ module |
| Data Assimilation & Integration | TC Chen, Greg, Jason | - Multi-fidelity alignment  - Feature engineering  - Ingest physics-model outputs & field data | - Data assimilation module |
| Active / Continual / Transfer Learning | Xuehang, TC, Jason | - Design drift-detection metrics  - Implement active-learning sampler & fine-tuning loop | - Finetune module |
| Physics & DL Model Development | Ross, Yilin, Eusebius Xuehang | - Maintain baseline physics simulator  - Train first-gen surrogate (e.g., U-Net) | - Modeling Engines |
| Software / Code Framework | Greg, Eusebius, Ross | - Codebase management and pipeline  - Web application?  - Containerize all modules (maybe?) | - Software |