

ML Lab Research Proposal

Non-Sinusoidal DTLMUW Waveform Discovery via Variational Autoencoder and Bayesian Optimization for Turbulent Drag Reduction

Team: Adhiraj Mann (solo)
Date: May 24th, 2026
Project: VAE-BO Waveform Optimizer for DTLMUW Drag Reduction
Target Compute: 1x AMD Instinct MI300X GPU (of available 8), approx. 2-week window

1. PROBLEM STATEMENT

Dolphin-skin-inspired Downstream-Traveling Longitudinal Micro-Ultrasonic Waves (DTLMUWs) have recently demonstrated drag reductions of up to 90% on airfoil surfaces by inducing a dynamic Stokes boundary layer that disrupts near-wall turbulent structures. However, virtually all existing DTLMUW research assumes the wall oscillates in a purely sinusoidal waveform—a mathematically convenient but physically arbitrary choice.

Prior DNS studies of spanwise wall oscillations have shown that non-sinusoidal waveforms can outperform the sinusoid when actuator constraints prevent reaching optimal operating conditions (Cimarelli et al., 2013), yet no work has applied machine learning to discover optimal non-sinusoidal shapes for out-of-plane (DTLMUW-type) oscillations. The combinatorial space of possible waveform shapes is far too large to explore exhaustively with expensive DNS simulations. This project addresses that gap: we propose a generative ML pipeline that learns the structure of the waveform shape space from existing DNS data and proposes novel, physically plausible non-sinusoidal DTLMUW waveform candidates predicted to outperform the sinusoidal baseline—reducing the number of expensive DNS evaluations required by orders of magnitude.

2. PROPOSED APPROACH

I propose a two-stage pipeline combining a Variational Autoencoder (VAE) with a Bayesian Optimization (BO) surrogate in latent space.

Stage 1 - VAE: Learning the Waveform Shape Space

A VAE is trained on a combined dataset of (a) real DNS waveforms from published literature with known drag reduction labels, and (b) synthetically generated waveforms produced via Fourier series superposition with randomized harmonic amplitudes and phase shifts. The VAE encoder compresses each 64-point wall velocity time series into a low-dimensional latent vector (8-16 dimensions); the decoder reconstructs the waveform from those coordinates. The forced continuity of the latent space ensures that coordinates between known waveforms decode into physically plausible interpolated shapes.

Stage 2 - Surrogate Predictor + Bayesian Optimization

A Gaussian Process Regression (GPR) surrogate is trained to map latent coordinates to predicted drag reduction % using only the labeled DNS subset. Bayesian optimization then navigates the learned latent space using an Expected Improvement acquisition function, identifying coordinates predicted to maximize drag reduction per unit actuator power. Promising coordinates are decoded back into waveform shapes—novel non-sinusoidal profiles never explicitly present in the training data—which constitute the paper's proposed candidates for future DNS validation.

Why This Approach Over Alternatives

- **Plain Neural Network Regressor:** Could predict DR% for known waveforms but cannot generate new ones.
- **GAN / Diffusion Model:** Could generate novel waveforms but requires far more training data than DNS studies provide and is harder to interpret.
- **VAE + BO Stack:** Uniquely suited to small, labeled datasets (tens to low hundreds of DNS samples), produces interpretable latent representations, and has strong precedent in engineering design optimization. Applying it to out-of-plane DTLMUW oscillations—rather than the in-plane spanwise oscillations covered by all prior ML work—is the primary novelty of this paper.

3. DATASET

Subset	Source	Approx. Size	Labels?	Role
Labeled DNS non-sinusoidal	Cimarelli et al. (2013), Physics of Fluids	~90-150 waveforms (9 families x amplitude/period grid)	Yes (DR% and net power saving%)	Train predictor + VAE
Labeled DNS sinusoidal	Gatti & Quadrio (2016) parametric study	~200-500 samples (subset extractable from paper)	Yes (DR%)	Train predictor + VAE
Synthetic unlabeled	Generated via Fourier superposition (randomized harmonics/phases)	~10,000 waveforms	No	VAE shape-space training only

Raw size for all subsets is essentially negligible, only a couple of MB, since the data consists of 1D physics simulations rather than raw sensor recordings.

One Row of the Dataset

Each labeled row contains: 64 normalized wall velocity values ($w_1 \dots w_{64}$) sampled across one oscillation cycle; physical parameters amplitude A^+ , period T^+ , Reynolds number Re ; waveform family label; source reference; and output targets DR% and net power saving%. Unlabeled synthetic rows contain only the 64 velocity values and physical parameters; DR% fields are null.

Preprocessing

- Normalize each waveform so $\max |w| = 1.0$ to ensure amplitude-invariant shape encoding.
- Filter synthetic waveforms exceeding realistic actuator acceleration limits.
- Train/validation/test split of 70/15/15 applied to labeled rows only; synthetic waveforms used exclusively for VAE training.

4. EVALUATION METRICS

Category	Metric	Target / Interpretation
VAE reconstruction	Mean Absolute Error (MAE) on held-out waveforms	Low MAE confirms decoder accurately reconstructs unseen shapes
VAE reconstruction	Latent space smoothness (nearest-neighbor interpolation quality)	Interpolated waveforms should be physically coherent, not noisy

DR% predictor	MAE and R ² on test set	R ² > 0.85 required before BO is trusted
DR% predictor	95% confidence interval calibration	GPR uncertainty estimates should be well-calibrated for BO to work
Generated waveforms	Predicted DR% vs. sinusoidal baseline	Primary claim: at least one novel waveform exceeds best sinusoidal DR%
Generated waveforms	Net power saving% of proposed candidates	Proposed waveforms must have positive net power saving to be practically relevant
Ablation	VAE latent dimension sensitivity (4, 8, 16, 32)	Identifies optimal compression level for this dataset size

Note: Because proposed waveforms are not validated with new DNS in this study, the headline scientific claim is framed as 'predicted to outperform' rather than 'demonstrated to outperform'. DNS validation is explicitly scoped as future work.

5. COMPUTE REQUIREMENTS

Phase	Hardware	Est. Wall-Clock	Notes
Synthetic data generation (10,000 Fourier waveforms)	CPU only	< 1 minute	Pure NumPy - no GPU needed
VAE training (~10,200 waveforms, 64-point series, latent dim 8-16)	1x AMD Instinct MI300X	Minutes to ~1 hour	Very lightweight 1-D time series VAE; 192 GB HBM3 far exceeds requirements
GPR surrogate training (~200-500 labeled samples)	1x AMD Instinct MI300X	< 5 minutes	GPR on small tabular data is computationally trivial
Bayesian optimization loop (~500-1,000 acquisition evaluations)	1x AMD Instinct MI300X	< 1 hour	Each evaluation queries the surrogate, not DNS; extremely fast
Latent dimension ablations (4 runs, 5 seeds)	1x AMD Instinct MI300X	2-4 hours total	Parallelizable across seeds
Total	1x AMD Instinct MI300X	~1 wall-clock day	Well within 2-week window; remaining 7 GPUs available for extensions

Software Dependencies

- ROCm 6.x + PyTorch ROCm wheel
- PyTorch, scikit-learn (GPR), NumPy, SciPy
- No CUDA-only kernels - all operations use standard torch and numpy
- No Docker required; pure Python venv

The AMD Instinct MI300X's 192 GB HBM3 is substantially more than required for this workload. The extra headroom allows exploration of larger VAE architectures or higher latent dimensions as a stretch goal without memory constraints.

6. EXPECTED DELIVERABLES

- **Trained VAE Model Checkpoint:** Encoder/decoder weights, latent space visualization (2-D PCA projection of all waveforms colored by DR%), and reconstruction error table across waveform families.
- **Trained GPR Surrogate:** Complete weights with calibration plot and R^2 /MAE on held-out test set.
- **Novel Waveform Proposals:** A set of 10-20 candidate waveforms identified by the BO optimizer, reported as 64-point time series with predicted DR% and net power saving%, compared against the best sinusoidal baseline.
- **Ablation Results Table:** Latent dimension sensitivity and impact of synthetic data augmentation on VAE reconstruction quality.
- **Research Paper Draft:** Written in standard ML conference format, covering methodology, results, and discussion of proposed waveforms as candidates for DNS validation.
- **Public Code Repository:** Fully reproducible, with data generation scripts, training scripts, and a README with MI300X/ROCm setup instructions.

Intentionally not promised: DNS validation of proposed waveforms (future work) and SOTA drag reduction claims. The goal is a clean, reproducible proof-of-concept demonstrating the pipeline's ability to propose physically grounded novel candidates.

7. RISK & MITIGATION

Risk	Likelihood	Mitigation
Labeled dataset too small (~200 samples) for GPR to generalize reliably	Medium	Add physics-informed features (acceleration parameter $A+/T+$, Stokes penetration depth) as additional input dimensions to the predictor. These are known to correlate with DR% from theory, giving the GPR structure to lean on beyond raw waveform shape.
VAE latent space loses smoothness with small dataset - novel coordinates decode to noise	Medium	Increase synthetic waveform augmentation to 50,000 samples. Apply beta-VAE regularization (increase KL divergence weight) to enforce smoother latent structure at the cost of slightly higher reconstruction error.
All BO-proposed waveforms cluster near the sinusoidal optimum - no genuine novelty	Low-Medium	Add diversity constraint to BO acquisition function penalizing proposals too close to known waveforms in latent space. Reframe paper as methodology demonstration rather than novel waveform discovery if necessary.
Cimarelli data primarily covers in-plane (spanwise) oscillations, not out-of-plane	High (known limitation)	Explicitly acknowledge in paper that training data is from spanwise oscillations and frame as a cross-domain transfer study. This is a limitation but also part of the paper's novelty argument.
Compute overrun or environment setup issues on AMD Instinct MI300X	Low	Entire pipeline (excluding DNS) runs comfortably on CPU in 1 day. MI300X is used for speed, not necessity. ROCm compatibility verified via standard PyTorch ROCm wheel with no CUDA-specific dependencies.

ML Lab Research Proposal

HF-Bench: Classical ML vs. Deep Learning for 30-Day Heart Failure Readmission Prediction on MIMIC-IV

Team: Anthony Nguyen (solo)

Date: May 24th, 2026

Project: HF-Bench: Classical ML vs. Deep Learning for 30-Day Heart Failure Readmission Prediction

Target Compute:

1x AMD Instinct MI300X (deep learning arms); CPU-only for classical ML arms. Est. total wall-clock: 2-3 days.

1. PROBLEM STATEMENT

Heart failure (HF) is one of the leading drivers of preventable hospital readmission in the United States, with roughly 20-25% of patients returning within 30 days of discharge and associated costs exceeding \$26 billion annually. Accurate risk stratification at the point of discharge could enable targeted interventions—closer follow-up, medication adjustment, home health referrals—that reduce readmissions and improve patient outcomes.

A substantial body of literature applies machine learning to this problem, with most published models relying on classical approaches: logistic regression, random forest, or gradient-boosted trees (XGBoost/LightGBM). Deep learning methods—particularly attention-based architectures (TabTransformer, FT-Transformer) and purpose-built tabular DL models (TabNet)—have shown promise on general tabular benchmarks but remain underexplored in the clinical HF readmission setting. Critically, most existing comparisons are conducted on single-institution datasets, rarely report subgroup performance across demographic groups, and do not systematically evaluate the interpretability-performance tradeoff that clinicians actually care about. This project fills that gap: we conduct the first systematic head-to-head benchmark of classical ML versus deep learning for 30-day HF readmission prediction on MIMIC-IV, with explicit evaluation of fairness across demographic subgroups and clinical interpretability via SHAP attribution.

2. PROPOSED APPROACH

We benchmark five model families on an identical MIMIC-IV HF cohort, held to a strict temporal train/validation/test split to simulate real deployment:

Model	Family	Hardware	Why included
Logistic Regression	Classical	CPU	Clinical standard; maximally interpretable baseline
XGBoost	Classical (GBDT)	CPU	Current SOTA on tabular EHR; sets the bar DL must beat
LightGBM	Classical (GBDT)	CPU	Faster GBDT alternative; cross-checks XGBoost results
TabNet	Deep Learning	1x AMD Instinct MI300X	Attention-based tabular DL with built-in feature selection
FT-Transformer	Deep Learning	1x AMD Instinct MI300X	Feature Tokenizer + Transformer; SOTA on tabular DL benchmarks

All models are trained on identical feature sets. Hyperparameters are tuned via Optuna (100 trials per model) on the validation set. Feature importance is assessed for all models using SHAP (TreeExplainer for classical models, GradientExplainer for DL models), enabling a direct interpretability comparison across paradigms. Subgroup analyses by race/ethnicity, sex, and age group (≥ 65 vs < 65) are conducted for all five models. This surfaces not only which model performs best overall, but which model performs most equitably across patient populations—a contribution largely absent from prior benchmarks.

The core research questions are: (1) Does deep learning outperform classical ML on this task? (2) If so, at what cost to interpretability and fairness? (3) Which model a clinician should actually deploy?

3. DATASET

All data comes from MIMIC-IV v2.2, a publicly available de-identified EHR dataset from Beth Israel Deaconess Medical Center, accessed via PhysioNet credentialed access (no DUA beyond PhysioNet credentialing, typically approved in 1-7 days).

Cohort definition: Adults (≥ 18 years) with primary ICD-10 diagnosis of heart failure (I50.x), discharged alive following an inpatient hospitalization, with at least one prior admission in MIMIC-IV (required for historical feature computation). Estimated cohort size is ~15,000-25,000 admissions. The outcome (label) is binary readmission within 30 days of discharge (1 = readmitted, 0 = not).

MIMIC-IV Module	Features Extracted
admissions / patients	Age, sex, race/ethnicity, insurance type, admission source, length of stay
diagnoses_icd	Comorbidities: diabetes, CKD, COPD, AFib, hypertension (one-hot); Charlson Comorbidity Index
labevents	BNP/NT-proBNP, creatinine, sodium, hemoglobin, eGFR at discharge + missingness flags
prescriptions	Loop diuretic dose, ACE inhibitor/ARB/ARNI, beta-blocker, total medication count
chartevents	Discharge weight, systolic BP, heart rate, O2 saturation
Engineered features	Admissions in prior 6 months, BNP delta (admission to discharge), days since last admit

Preprocessing

- Median imputation for missing labs + binary missingness indicators.
- One-hot encoding for categorical variables.
- StandardScaler normalization (fit on train only, applied to val/test).
- SMOTE oversampling on the training fold only to balance classes.
- Temporal train/val/test split: 70% / 15% / 15% sorted strictly by admission date.

4. EVALUATION METRICS

Experiment	Metric
Primary benchmark (all 5 models)	AUROC, AUPRC primary performance metrics; target AUROC ≥ 0.75
Clinical operating point	Sensitivity + specificity at 80% specificity threshold; PPV, NPV
Calibration	Brier score + calibration plot (reliability diagram) for all models

Fairness / subgroup	AUROC stratified by race/ethnicity, sex, age ≥ 65 vs < 65 for all 5 models
Interpretability	Mean SHAP feature ranking; top-10 feature table; SHAP summary plot per model
DL vs classical gap	Delta AUROC (FT-Transformer - XGBoost); paired bootstrap significance test
Ablation	All models trained without engineered features vs full feature set; delta AUROC

Seeds used: 42, 123, 7, 0, 256. All metrics are reported as mean \pm 95% CI across 5 seeds. Hyperparameter tuning is performed on the validation set only; final numbers are reported on the held-out temporal test set.

5. COMPUTE REQUIREMENTS

Classical ML arms (logistic regression, XGBoost, LightGBM) run entirely on CPU and require no special hardware. Deep learning arms (TabNet, FT-Transformer) use the AMD Instinct MI300X for training and hyperparameter search.

Phase	Hardware	Estimated Wall-Clock
MIMIC-IV data pull + preprocessing (mimic-code SQL + pandas)	CPU only	~2-4 hours (one-time)
Classical ML training + Optuna tuning (100 trials each)	CPU only	~1-2 hours total
TabNet training + Optuna tuning (100 trials, 5 seeds)	1x AMD Instinct MI300X	~4-6 hours
FT-Transformer training + Optuna tuning (100 trials, 5 seeds)	1x AMD Instinct MI300X	~6-8 hours
SHAP attribution (all 5 models, test set)	CPU / AMD Instinct MI300X	~1-2 hours
Subgroup analysis + figure generation	CPU only	~1 hour
Total	1x AMD Instinct MI300X node	~1.5-2 days wall-clock with buffer

Hardware & Storage Headroom

GPU memory: TabNet and FT-Transformer on a ~20,000-patient tabular dataset fit comfortably within 16 GB VRAM. The AMD Instinct MI300X (192 GB HBM3) is far above what is strictly required, but allows large Optuna search spaces and full-batch training without minibatch approximation artifacts. Storage requires ~5-10 GB for relevant MIMIC-IV tables + preprocessed parquet cache.

Software Dependencies

- Python 3.10+, pandas, numpy, scipy, scikit-learn, xgboost, lightgbm
- pytorch-tabnet, pytorch-frame (FT-Transformer implementation)
- shap, optuna, imbalanced-learn, matplotlib, seaborn
- ROCm 6.x + official PyTorch ROCm wheel (for MI300X)
- mimic-code SQL scripts for cohort extraction (MIT-licensed)
- No Docker required; standard Python venv. No CUDA-only kernels.

6. EXPECTED DELIVERABLES

By the end of the two-week window, we will provide:

- Reproducible cohort extraction pipeline (SQL + Python) from MIMIC-IV.
- Trained and tuned models for all 5 model families with full cross-validation results.
- Head-to-head benchmark table: AUROC, AUPRC, Brier score, sensitivity, specificity for all models.
- Subgroup fairness table: AUROC by race/ethnicity, sex, and age group for all 5 models.
- SHAP summary plots and top-10 feature tables for all models (classical TreeExplainer + DL GradientExplainer).
- ROC and precision-recall curves overlaid for all 5 models.
- Draft methods + results sections formatted for npj Digital Medicine.

Longer-term deliverables: Full manuscript submission to npj Digital Medicine, PLOS ONE, or JAMIA Open; Abstract submission to AMIA Annual Symposium or AHA Scientific Sessions; and a public GitHub repo under MIT license.

We are intentionally not promising that deep learning beats XGBoost—a negative result (classical ML matches or outperforms DL on this task) is equally publishable and arguably more clinically useful. The goal is a clean, honest, reproducible benchmark with equity findings.

7. RISK & MITIGATION

Risk	Likelihood	Mitigation
MIMIC-IV PhysioNet credentialing delayed	Medium	Apply immediately (typically 1-7 days). Use MIMIC-IV demo (100-patient public subset, no credentialing) to build and validate full pipeline in parallel. No time lost.
DL models underperform XGBoost on this task	Medium	This is a valid, publishable finding. 'Classical ML matches DL on tabular EHR data with better interpretability' is a useful clinical result. Frame as an honest benchmark. XGBoost arm is the safety net.
FT-Transformer or TabNet fails to converge / hyperparameter sensitivity	Medium	Isolate DL training behind unit-tested modules. Fall back to MLP (simpler DL baseline) if attention-based models misbehave. ROCm compatibility tested in smoke run before full training.
SHAP GradientExplainer is slow or unstable on DL models	Medium	Use SHAP KernelExplainer as fallback (model-agnostic, slower but reliable). Limit explainability analysis to a random 500-patient subset of the test set if runtime is prohibitive.
Subgroup sample sizes too small for reliable AUROC estimates	Low	Report CIs explicitly. Collapse granular race categories to broader groups if $n < 100$. Flag small-n subgroups rather than omitting them.
Class imbalance (~20% positive) degrades minority-class recall	Low	SMOTE on training fold + <code>class_weight='balanced'</code> for classical models + <code>pos_weight</code> in DL loss. Report AUPRC alongside AUROC. Tune threshold to clinical operating point rather than defaulting to 0.5.

I will check in at the 1-week mark with classical ML results and Stage-1 DL training numbers so we can decide together whether to push for the full 5-model benchmark or trim to the strongest 3.

Research Proposal

CC-FNO-E: Conservation-Constrained Fourier Neural Operators for Regime-Generalizable Fluid Dynamics

Team: AI Research Collaborative (Group)

Date: May 24, 2026

Project:

CC-FNO-E: Conservation-Constrained Fourier Neural Operators for Regime-Generalizable Fluid Dynamics

Target Compute: 4x AMD Instinct MI300X GPUs (single node), approx. 1 week

1. PROBLEM STATEMENT

Neural operators have emerged as highly efficient surrogates for partial differential equation (PDE) solvers. However, their accuracy degrades sharply when they are evaluated outside the training distribution of physical parameters. In fluid dynamics, achieving out-of-distribution (OOD) generalization across different Reynolds number regimes in 2D incompressible Navier-Stokes equations remains a core challenge.

Existing standard Fourier Neural Operators (FNOS) fail to respect fundamental physics, leading to severe error accumulation and instability when transitioning between laminar and turbulent flows. This project tests whether enforcing hard conservation constraints and physical consistency can yield near-regime-invariant accuracy and stable autoregressive rollouts without requiring target-regime training labels.

2. PROPOSED APPROACH

We are building CC-FNO-E and its variant CC-FNO-E-PF, which introduce hard conservation constraints into the Fourier Neural Operator architecture:

- Leray Projection Layer:** Enforces divergence-free velocity fields at every spectral layer of the network.
- Dissipation-Aware Scaling Constraint:** Bounds kinetic energy at the output to act as an amplitude regularizer.
- Pushforward Training (CC-FNO-E-PF):** Combines conservation constraints with multi-step pushforward training to drastically close the accuracy-stability Pareto gap during long rollouts.

This stack is selected over a standard FNO because the energy constraint provides excellent long-term stability, while the combination of both constraints yields the best overall per-step accuracy.

3. DATASET

The datasets consist of numerical simulations of 2D incompressible Navier-Stokes equations across varying fluid regimes:

Regime Category	Corpus / Configuration	Size / Properties	Role
In-Distribution (Train)	2D Navier-Stokes Simulation Data (Re \in {500, 1000})	~50 GB	Used strictly for training the base operators.
Out-of-Distribution (Eval)	Laminar OOD Evaluation (Re = 100)	Tabular / Mesh Tensor	Downstream evaluation of zero-shot transfer accuracy.

Out-of-Distribution (Eval)	Turbulent OOD Evaluation ($Re = 2000$)	Tabular / Mesh Tensor	Downstream evaluation of zero-shot stability.
----------------------------	--	-----------------------	---

Preprocessing

- Grids are downsampled to a uniform spectral mesh size (e.g., 64×64 or 128×128).
- Data is stored as memory-mapped tensors for optimized data-parallel loading.
- Total on-disk footprint is estimated around 120 GB, well within standard storage limits.

4. EVALUATION METRICS

Experiment Type	Primary Metric	Target / Interpretation
OOD Regime Accuracy	Single-step relative L^2 error across all Reynolds numbers	Measures generalization across unseen fluid regimes
Long-term Rollout Stability	Percentage drift over 100-step rollouts	Enforces non-divergent physical bounds over time
Autoregressive Horizon	Maximum rollout horizon achieved while maintaining stability	Targeting 90-95 steps before error breaks physics bounds
Statistical Significance	Wilcoxon signed-rank test ($p = 0.031$)	Evaluated over 5 random seeds to guarantee consistency

5. COMPUTE REQUIREMENTS

Estimates are based on initial single-GPU benchmarks running standard FNO blocks with custom projection layers, scaled up to a multi-GPU AMD node configuration.

Phase	Hardware	Estimated Wall-Clock	Notes / Details
Phase 1: CC-FNO-E Base Training	4x AMD Instinct MI300X	~2 days	Training across $Re \in \{500, 1000\}$ with mixed-precision
Phase 2: Pushforward Fine-tuning (CC-FNO-E-PF)	4x AMD Instinct MI300X	~2 days	Multi-step lookahead backpropagation to close the stability gap
Phase 3: OOD Sweeps & Ablations	2x AMD Instinct MI300X	~2 days	Sweeping 10 model variants over 5 random seeds across 4 Reynolds numbers

Dependencies & Memory Allocation

- PyTorch + ROCm Toolkit compatibility framework.
- torchharmonic or custom FFT-based spectral layers.
- Custom Leray projection and dissipation scaling modules implemented in pure PyTorch for high portability.

6. EXPECTED DELIVERABLES

1. **Model Checkpoints:** Converged weights for both CC-FNO-E and CC-FNO-E-PF variants.

2. **Benchmark Headline Table:** Relative L^2 error comparison showing a reduction from the standard FNO error (0.039–0.105) down to the target conservation-constrained error (0.013–0.019).
3. **Ablation Analysis:** Clear data separating the distinct impacts of the energy constraint (amplitude regularization) versus the Leray projection.
4. **Conference Draft:** A fully compiled manuscript formatted for the International Conference on Machine Learning (ICML).

7. RISK & MITIGATION

Risk	Likelihood	Mitigation
Pushforward training instability	Medium	Fall back to incremental horizon scheduling (start with 2-step lookahead, scale up to 5+ steps dynamically).
High computational overhead from spectral Leray projections	Low-Medium	Optimize via inline execution; if needed, compute projections every N blocks rather than at every single layer.
Energy constraint acts strictly as a dampener rather than preserving true forcing dynamics	Medium	Calibrate the dissipation-aware scaling factor using a small validation sweep to ensure fluid structures remain sharp.

Research Proposal

Airfoil Generation via SE(3)-Equivariant Diffusion on Dual-Quaternion Frame Graphs

Team: Rishi Shah (solo)
Date: May 24, 2026
Project: Airfoil Generation via SE(3)-Equivariant Diffusion on Dual-Quaternion Frame Graphs
Target Compute: 2x AMD Instinct MI300X, approx. 2 weeks

1. PROBLEM STATEMENT

This project proposes a novel generative framework for aerodynamic shape design that represents an airfoil as a graph of local geometric frames. Each local frame is encoded as a dual quaternion, allowing the airfoil to be modelled as a sequence of rigid transformations that capture local position, orientation, curvature, and surface evolution.

Rather than performing diffusion in Euclidean coordinate space, the model performs diffusion on a structured product space: the Lie group SE(2) governs pose (position and in-plane orientation), while a Euclidean fiber encodes higher-order local geometry (signed curvature, curvature rate, and local half-thickness). This fiber-bundle decomposition keeps group operations mathematically valid while making full use of all six degrees of freedom available in a dual-quaternion parametrisation. The denoising network is designed to be SE(2)-equivariant and operates on two complementary edge features: relative SE(2) frame transformations and Euclidean fiber differences. This allows the model to learn geometric relationships directly from the underlying transformation structure. The central hypothesis is that learning distributions over local geometric transformations and intrinsic curvature will produce smoother, more physically realistic airfoils than learning distributions over raw coordinates. The framework scales naturally to full 3-D wing surfaces.

2. PROPOSED APPROACH

A 2-D airfoil curve lives in the $z=0$ plane. A local Frenet-Serret frame requires exactly three numbers: $(x, y, \theta) \in SE(2)$. A unit dual quaternion carries six independent degrees of freedom; the three unused dimensions are repurposed as a geometric fiber attached to each surface frame:

Extra DOF	Encoded Quantity	Aerodynamic Significance
Out-of-plane rotation 1	Signed curvature $K(s)$	Pressure gradient, separation onset
Out-of-plane rotation 2	Curvature rate $K'(s)$	Smoothness, transition prediction
z-translation	Local half-thickness $t(s)$	Thickness distribution, drag

The resulting node representation lives in $SE(2) \times \mathbb{R}^3$. Message passing employs Frenet-Serret edge transformations. Diffusion is performed in the Lie algebra by linearization around the group identity, admitting closed-form score matching. A geometric consistency constraint loss regularizes the network by penalizing frames whose curvature encoding is inconsistent with the implied frame rotation rate.

The denoising network is implemented as an SE(2)-equivariant graph transformer with dual edge-processing heads (Pose head + Fiber head). The model is conditioned on Reynolds number (Re), angle of attack (α), target lift (C_l), and target drag (C_d) via FiLM or cross-attention layers. Airfoil reconstruction solves a global least-squares

integration block to recover positions without accumulating sequential drift, enforcing trailing-edge closure as a hard equality constraint.

3. DATASET

The framework utilizes a combination of open aerodynamic source profiles and large-scale synthetic data augmentation:

Corpus / Dataset	Source	Raw Size / Sample Count	Role
UIUC Airfoil Database	Public Aerodynamic Repo	~1,600 unique airfoils	Base seed configurations
Augmented Perturbations	CST coefficient variations ($\pm 20\%$)	~50,000 - 90,000 shapes	Synthetic training expansion
Aerodynamic Targets	XFOIL simulations over (Re , AoA) grid	Fully Labeled Map	Surrogate network training & conditioning

Preprocessing & Normalization

- **Canonical Frame Normalization:** Chord length scaled to 1.0; leading edge translated to origin; trailing edge anchored at (1, 0).
- Each airfoil is re-sampled along its arc-length into N points (typically $N=128$) to extract positions, tangents, normals, curvatures, and thickness metrics.
- Samples with self-intersections or non-closing trailing edges are strictly rejected.

4. EVALUATION METRICS

Category	Metric	Description / Goal
Geometry	Curvature smoothness	L^2 norm of $K''(s)$; minimizing non-physical kinks
Geometry	Surface continuity	G1 and G2 continuity checks along the entire perimeter
Geometry	Self-intersection / Closure	Fraction of self-intersections (target 0); Trailing edge distance error
Aerodynamics	Lift & Drag Error	Absolute error vs. target C_l and C_d coefficients
Aerodynamics	Lift-to-drag-ratio	C_l/C_d optimization performance at design operating point
Generative	Diversity & Novelty	Spread in shape space; Fraction of unique samples not in train set

5. COMPUTE REQUIREMENTS

The model leverages multi-accelerator compute to handle graph message passing and coordinate-free diffusion processing blocks efficiently.

Phase	Hardware	Estimated Wall-Clock	Notes
Model Pretraining & Augmentation	2x AMD Instinct MI300X	~5 days	Joint SE(2) and fiber diffusion transformer training

Aerodynamic Surrogate Training	1x AMD Instinct MI300X	~1 day	Separate training of the physics guide network
Guided Denoising Sweeps & Eval	1x AMD Instinct MI300X	~2 days	Generating, filtering, and post-processing candidate airfoils

Software Dependencies

- ROCm 6.x + PyTorch ROCm wheel environment
- NumPy, SciPy, scikit-learn, and custom FFT / Lie algebra utilities
- XFOIL standard bindings for automated aerodynamic validation sweeps
- No Docker required; standard Python venv execution setup

6. EXPECTED DELIVERABLES

- Converged weights for the SE(2)-equivariant graph transformer and the separate neural aerodynamic surrogate.
- An expanded, cleaned, and labeled training database of 50,000–90,000 airfoil profiles complete with XFOIL evaluation logs.
- A set of 10-20 novel high-efficiency candidate airfoil geometries meeting target C_l and C_d conditions.
- A benchmark headline table comparing geometric quality and lift-to-drag performance against coordinate-diffusion and CST baselines.
- A fully compiled conference draft targeted at the International Conference on Machine Learning (ICML).

7. RISK & MITIGATION

Risk	Likelihood	Mitigation
Log-map singularity near high-curvature leading edge	Medium	Arc-length re-sampling to guarantee consecutive frames differ by $< \pi/2$; apply smooth clamping function $\tanh(\omega /\pi) \times \pi$ to rotation magnitudes.
Pose and fiber representations become geometrically inconsistent	Medium	Enforce a geometric consistency loss regularizer linking rotational twist to integrated curvature during training.
Surrogate guidance diverges at high noise levels	High	Apply noise annealing—only activate surrogate guidance in the final 30% of denoising steps where the geometry is recognizable.

ML Lab Research Proposal

Param-Mix: Parameter-Space Interpolation for Mixed-Data Optimization in Large Language Models

Team:	Olivia (solo)
Date:	May 24, 2026
Project:	Param-Mix: Parameter-Space Interpolation for Mixed-Data Optimization in LLMs
Target Compute:	2x AMD Instinct MI300X GPU accelerators, approx. 2 weeks

1. PROBLEM STATEMENT

Optimizing data mixtures for large language models remains computationally prohibitive because identifying the best mixture ratios typically requires repeated full-scale training runs guided by heuristic tuning. Existing methods are primarily optimized for language modeling loss such as perplexity, but reductions in perplexity do not reliably translate to improved downstream reasoning, coding, and mathematical performance.

This project investigates whether parameter-space interpolation can serve as a high-fidelity proxy for mixed-data optimization by exploiting the shared first-order optimization dynamics between independently trained domain experts and mixed-domain training trajectories.

2. PROPOSED APPROACH

We propose a framework that approximates mixed-data training using linear interpolation between independently trained domain-specific expert models. Given interpolation coefficients $\alpha \in \Delta^{K-1}$, the merged parameters are constructed as:

$$\Theta_{merge}(\alpha) = \Theta_{base} + \sum_k \alpha_k (\Theta_k - \Theta_{base})$$

This formulation allows efficient exploration of the mixture simplex without retraining full models. The method is motivated by a second-order Taylor expansion analysis showing that mixed-data optimization and parameter interpolation share identical first-order gradient terms, while discrepancies are confined to second-order curvature interactions involving Hessian-gradient products $H_k g_j$. We additionally fit LightGBM regressors to model the performance surface across downstream benchmark domains and optimize a utility function over predicted benchmark scores.

3. DATASET

We use a large-scale mid-training corpus partitioned into four primary domains: mathematics, code, supervised fine-tuning (SFT) data, and web/other data. The benchmark suite spans reasoning, mathematical problem solving, coding, language understanding, and professional knowledge tasks, including GSM8K, MATH, HumanEval, MBPP, GPQA, MMLU, ARC, BBH, and TriviaQA.

Preprocessing

- Domain clustering and token filtering to enforce structural purity.
- Hierarchical decomposition into fine-grained sub-clusters (e.g., formal mathematics, synthetic study notes, code repositories, long chain-of-thought supervision, and academic papers).
- Normalization of text distributions across all training segments.

4. EVALUATION METRICS

Category / Experiment	Metric	Target / Interpretation
Proxy Validation	Spearman rank correlation coefficient (ρ)	Target $\rho > 0.9$ between predicted interpolation performance and full training trajectories
Downstream Capability	Average benchmark accuracy	Evaluated across 5 key capability domains: math, code, knowledge, language, and reasoning
Interference Analysis	Hessian interaction matrices & task-vector cosine similarities	Quantify cross-domain interference and optimization orthogonality to identify breakdown points

5. COMPUTE REQUIREMENTS

The project requires approximately 2x AMD Instinct MI300X GPU accelerators with a combined memory budget near 384GB HBM memory to support distributed sparse Mixture-of-Experts mid-training experiments.

Phase	Hardware	Estimated Wall-Clock	Notes / Scope
Domain-Specific Expert Training	2x AMD Instinct MI300X	~7-9 days	Multiple 5B-25B token expert training runs with shared initialization
Interpolation Sweeps & Surface Fitting	2x AMD Instinct MI300X	~2-3 days	Exploring mixture simplex and training LightGBM performance regressors
Benchmark Evaluation & Verification	1x AMD Instinct MI300X	~1-2 days	Validation verification runs on sampled mixture configurations

Software Dependencies

- PyTorch, DeepSpeed distributed framework, FlashAttention
- Distributed checkpoint interpolation utilities and custom Hessian-vector product modules
- LightGBM-based regression models for downstream capability prediction
- ROCm 6.x + official PyTorch ROCm wheel environment

6. EXPECTED DELIVERABLES

- A working framework for efficient data mixture optimization using parameter interpolation without full retraining loops.
- Trained domain-specialized expert model checkpoints and a final optimized mid-trained language model.
- Comprehensive benchmark evaluation reports and detailed visualization maps of capability landscapes projected over the mixture simplex.
- Empirical analysis of first-order versus second-order optimization behavior during interpolation-based search sweeps.
- Reproducible scripts for interpolation search, utility optimization, and downstream benchmark testing.

7. RISK & MITIGATION

Risk	Likelihood	Mitigation
Higher-order curvature interactions cause parameter interpolation to diverge from mixed-data training	Medium	Constrain expert training to short horizons with constant learning rates, preserve shared initialization across all experts, and empirically validate via Hessian interaction analysis.
Domain-specific task vectors do not remain sufficiently orthogonal, reducing rank correlation	Medium	Monitor cosine similarity measurements continuously and perform rapid full-training verification runs on sampled mixture boundaries.

ML Lab Research Proposal

Hybrid-Weather: Combining RNN/LSTM and Gradient Boosting for Extreme Weather Prediction

Team: Team B - Anubhav Chatterjee & Arya Addagarla
Date: April 19, 2026
Project: Hybrid-Weather: Combining RNN/LSTM and Gradient Boosting for Extreme Weather Prediction
Target Compute: 1x AMD Instinct MI300X GPU, approx. 2 weeks

1. PROBLEM STATEMENT

Accurately predicting extreme weather events (e.g., heatwaves, storms) remains difficult due to the separation of short-term sensor data and long-term climate patterns in existing models. This project aims to improve prediction accuracy by combining both data types into a unified AI model. Better forecasts can reduce economic damage and improve public safety.

2. PROPOSED APPROACH

We will develop a hybrid model that combines a recurrent neural network (RNN/LSTM) for short-term temporal patterns with a gradient boosting model for long-term climate trends. The outputs will be fused using a meta-model (ensemble method).

This approach is chosen because LSTMs handle time-series dependencies well, while boosting models capture structured, non-linear relationships. The ensemble allows leveraging the unique strengths of both paradigms over traditional single-architecture setups.

3. DATASET

We will use publicly available weather datasets such as NOAA or Kaggle weather data, including temperature, humidity, wind speed, and precipitation over time. The dataset will include multiple years of hourly or daily observations.

Preprocessing

- Handling missing data points via statistical imputation or backward filling.
- Standard normalization across continuous feature columns.
- Feature engineering, including rolling averages, historical lag variables, and cyclic time feature encoding.

4. EVALUATION METRICS

Experiment Category	Metric	Target / Interpretation
Prediction Accuracy	Mean Absolute Error (MAE) & Root Mean Squared Error (RMSE)	Evaluates the continuous track precision of the model
Extreme Event Detection	Precision, Recall, and F1-score	Ensures high sensitivity for identifying rare, high-impact weather anomalies

5. COMPUTE REQUIREMENTS

Training is highly efficient and optimized for standard multi-core setups, utilizing high-performance acceleration where necessary.

Phase	Hardware	Estimated Wall-Clock	Notes
Model Architecture Training & Tuning	1x AMD Instinct MI300X GPU	1-3 hours	Fits comfortably inside a fraction of the HBM memory buffer

Software Dependencies

- Python 3.10+, PyTorch or TensorFlow framework core
- scikit-learn, pandas, NumPy, and gradient boosting bindings
- ROCm compatibility setup for execution on AMD hardware infrastructure

6. EXPECTED DELIVERABLES

- A fully trained and optimized hybrid LSTM-Gradient Boosting model.
- A short research report summarizing data preparation methods, architectural choices, and evaluation results.
- A simple demo notebook containing prediction visualizations overlaid on top of genuine test data sequences.

7. RISK & MITIGATION

Risk	Likelihood	Mitigation
Insufficient improvement over baseline architectures due to limited data or time constraints	Medium	Implement a simpler baseline model (e.g., linear regression or single ML model) for immediate comparison. If the hybrid model underperforms, we will refine engineering features or reduce model complexity to ensure stable results.

ML Lab Research Proposal

Quake-GNN: Multimodal Graph Neural Network for High-Reliability Earthquake Prediction

Team: Team B - Arya Addagarla & Anubhav Chatterjee
Date: April 24, 2026
Project: Quake-GNN: Multimodal Graph Neural Network for High-Reliability Earthquake Prediction
Target Compute: 1x AMD Instinct MI300X GPU, approx. 2 weeks

1. PROBLEM STATEMENT

We are addressing the unpredictability of earthquakes and how incorporating sensor data and analyzing it with neural network-based predictive models can yield high-reliability predictions. This matters because living in California, a place with relatively high seismic activity, keeps civilians at greater risk of the next deadly earthquake.

2. PROPOSED APPROACH

We are using a Multimodal Graph Neural Network. This treats every seismic station, GPS point, and fiber-optic cable as a connected node in a massive web, similar to how the brain connects different senses. We chose this over traditional methods because it doesn't just see a line from data; instead, it tracks how that vibration moves across a map.

By requiring the signal to travel at the actual speed of a tectonic wave, we can instantly tell the difference between a real earthquake and a truck driving past a single sensor.

3. DATASET

We are using a multi-stream physics data repository incorporating explicit spatial relationships:

- Raw vibrations from the **Stanford Earthquake Dataset**.
- Satellite movement logs from active **GPS stations**.
- High-speed telemetry from localized **fiber-optic cable arrays**.

Preprocessing

- Use the R programming language to explore and clean raw data streams.
- Enforce strict temporal synchronization to ensure every data point matches a shared universal clock.
- Construct graph adjacency matrices based on geographical coordinates of sensor nodes.

4. EVALUATION METRICS

Experiment Category	Metric	Target / Interpretation
Predictive Accuracy	Overall Percentage Accuracy	Success threshold established at > 90% accuracy
Anomalous Identification	F1-Score	Balances system sensitivity to small quakes while avoiding false alarms ('crying wolf')
Spatio-Temporal Localization	Epicenter distance error & Warning lead time	Quantifies the extra warning time delivered before shaking starts

5. COMPUTE REQUIREMENTS

To handle this massive amount of data, we utilize a high-end accelerator configuration. This specialized hardware allows the AI to think about multiple data streams at the same time rather than one by one.

Phase	Hardware	Estimated Wall-Clock	Notes / Purpose
Data Engineering & Exploration	CPU Core Allocation	~4-6 hours	Math exploration and matrix tuning using R
Multimodal GNN Training	1x AMD Instinct MI300X GPU	~48 hours	Heavy AI lifting using PyTorch graph loaders

Software Dependencies

- R environment for initial data cleaning and cleaning analytics.
- PyTorch framework core + PyTorch Geometric or related extensions for graph operations.
- ROCm execution environment configured for AMD MI300X hardware allocation (minimum 24GB VRAM target).

6. EXPECTED DELIVERABLES

- A functional Multimodal Graph Neural Network model optimized for seismic time-series analysis.
- An exploratory data analysis pipeline implemented in R for parsing structural sensor feeds.
- An end-to-end training and inference pipeline mapping live sensor streaming to real-time warning indices.

7. RISK & MITIGATION

Risk	Likelihood	Mitigation
Discovery of illusory correlations when tracking patterns in highly noisy data streams	Medium	Run the AI to isolate real predictors, then perform cross-validation against historic real data blocks to ensure correlation validity.
Regular ambient tremors (e.g., localized vehicle traffic noise) contaminating input sensors	High	Incorporate physical speed propagation constraints; require signal tracking to match true tectonic wave velocities to filter out non-seismic anomalies.

ML Lab Research Proposal

CLD-Trans: Causal-Lagged Dynamic Transformer for Biosignals

Team: Ankur (solo)
Date: April 25, 2026
Project: CLD-Trans: Causal-Lagged Dynamic Transformer for Biosignals
Repo: <https://github.com/coolguy46/CLD-Trans>
Target Compute: 8x AMD Instinct M1300X (single node), approx. 2 weeks

1. PROBLEM STATEMENT

Existing biosignal foundation models (BIOT, BENDR, NeuroLM, ECG-FM) treat EEG/ECG channels as either independent or coupled by a static attention pattern, even though clinicians know that signals propagate through the body in continuous time—seizures spread from a focal lead to its neighbors over hundreds of milliseconds, and ECG depolarization reaches V6 a few milliseconds after V1. Existing causal-discovery methods (DYNOTEARS, Rhino) recover lagged graphs but assume integer-sample lags and acyclic dependencies, which is biologically wrong. I want to test whether a model built around continuous-valued, possibly cyclic propagation lags can recover clinically meaningful structure (e.g. seizure focal leads) with no task labels.

2. PROPOSED APPROACH

I am building CLD-Trans, which combines three components under a single non-Gaussian Lagged-Delay Structural Equation Model (LD-SEM) objective:

1. A Motif VQ-VAE that tokenizes raw waveforms into a 512-entry codebook of physiological primitives (spike-wave, QRS, K-complex, etc.).
2. A differentiable fractional-lag operator (FFT phase shift, `modules/fractional_delay.py`) that learns sub-sample per-channel-pair delays τ_{ij} in the range 0 to τ_{\max} .
3. A graph-conditioned Neural ODE that integrates latent state through the inferred time-varying adjacency $A(t)$.

I picked this stack over a plain Transformer FM because the lag operator gives me an interpretable, identifiability-friendly target—the closest thing in the literature is Rhino, but Rhino assumes integer lags and acyclic graphs, neither of which holds for EEG/ECG. The LD-SEM objective is also the same object I want to use in the identifiability theorem on the theory side, so method and theory share one optimization target.

3. DATASET

All datasets are public and pulled from PhysioNet open-data S3 buckets via `scripts/download_datasets_aws.sh`. No DUA, no PHI handling.

Pretraining (no labels used):

Corpus	Source	Raw Size	Notes
MIMIC-IV-ECG v1.0	<code>s3://physionet-open/mimic-iv-ecg/1.0/</code>	approx 90 GB	approx 800k 12-lead, 10 s, 500 Hz ECGs

EEG Motor Movement/Imagery (EEGMMIDB)	s3://physionet-open/eegmmidb/1.0.0/	approx 3 GB	109 subjects, 64-ch, 160 Hz
---------------------------------------	-------------------------------------	-------------	-----------------------------

Downstream evaluation:

Dataset	Source	Raw Size	Use
CHB-MIT	PhysioNet	approx 40 GB	Zero-shot focal-lead localization (headline) + few-shot
PTB-XL	PhysioNet	approx 3 GB	Few-shot 5-superclass arrhythmia
Sleep-EDF	PhysioNet	approx 8 GB	Few-shot 5-stage sleep scoring

Preprocessing

- Resample to a per-modality common rate (160 Hz for EEG, 500 Hz for ECG).
- Bandpass filter (0.5-40 Hz EEG, 0.05-150 Hz ECG), z-score per channel.
- Cache fixed-length windows as memory-mapped npy shards for fast data-parallel loading.

Realistic on-disk footprint: approx 145 GB raw + roughly the same again for the npy window cache, so I plan for approx 300 to 400 GB of scratch, well under a 1 TB budget.

4. EVALUATION METRICS

Experiment	Metric
Zero-shot focal-lead localization (CHB-MIT, headline)	top-1/top-3/top-5 accuracy of argmin over i of (sum over j of τ_{ij}) vs. clinician annotation
Few-shot CHB-MIT seizure detection	AUROC, AUPRC at 1 percent/10 percent/100 percent label budgets
Few-shot PTB-XL arrhythmia	macro-AUROC over 5 superclasses
Few-shot Sleep-EDF	macro-F1 and Cohen's kappa over 5 stages
Synthetic LD-SEM identifiability	tau recovery MAE, edge-support F1 vs. ground truth
Ablations (no-VQ, no-lag, integer-lag, no-ODE)	Same primary metric per dataset, 5-seed mean plus or minus 95 percent CI

Seeds: 42, 123, 7, 0, 256.

5. COMPUTE REQUIREMENTS

Estimates below are scaled from a measured single-GPU smoke run (python main.py mode=stage1 train.max_steps=1) on a development GPU. Padded by 25% for M1300X.

Phase	Hardware	Estimated Wall-Clock
Stage 1 Motif VQ-VAE + LD-SEM pretraining (bf16, DDP across 8 GPUs, eff. batch 256, approx 7.7k steps/epoch, approx 60 epochs)	8x M1300X	approx 3 to 4 days
Stage 2 Graph-ODE fine-tune	8x M1300X	approx 1 day

Downstream eval (3 datasets x 3 budgets x 5 seeds)	1 to 2 M1300X per run	approx 1.5 days aggregate
Ablations + synthetic identifiability sweeps	1 to 8 M1300X	approx 2 days
Total	8x M1300X node	approx 8 to 10 wall-clock days inside a 2-week window

GPU memory: the largest activation tensor is the dense C-by-C-by-T lag tensor for 64-channel EEG plus the ODE adjoint state. On a single 192 GB M1300X this should fit without activation checkpointing; on an 80 GB H100 it would force checkpointing or tensor parallelism. That is the main reason I am asking for M1300X specifically rather than just 'any 8-GPU node'.

Storage: approx 400 GB scratch (datasets + npy cache + checkpoints).

Software / Dependencies

- ROCm 6.x + the official PyTorch ROCm wheel.
- pip install -r requirements.txt (stock PyTorch, torchdiffeq, numpy, scipy, mne, wfdb, hydra-core).
- AWS CLI for the one-time PhysioNet download.
- No CUDA-only kernels (no flash-attn v2, no xformers, no custom Triton CUDA kernels). FFT lag op is plain torch.fft.
- No Docker required; pure Python venv.

6. EXPECTED DELIVERABLES

- A trained Stage-1 motif VQ-VAE checkpoint on EEGMMIDB + MIMIC-IV-ECG, plus the trained Graph-ODE on top.
- Headline experiment table: zero-shot focal-lead localization on CHB-MIT vs. BIOT, BENDR, EEG-GCNN, DYNOTEARS, Rhino baselines.
- Few-shot transfer numbers on CHB-MIT, PTB-XL, Sleep-EDF at 1%/10%/100% label budgets.
- Synthetic LD-SEM identifiability table (tau recovery error and edge-support F1 with and without the non-Gaussianity assumption).
- A NeurIPS-format draft with plots and tables generated end-to-end by scripts/make_figures.sh, plus per-seed JSON metrics in results/ and logs in logs/.
- The full repo remains MIT-licensed and public, with an M1300X/ROCm-specific section in the README and the reproducibility statement.

I am intentionally not promising 'SOTA on every benchmark'—the goal is a clean, reproducible test of the identifiability + zero-shot focal-lead hypothesis, with the few-shot numbers as supporting evidence.

7. RISK & MITIGATION

Risk	Likelihood	Mitigation
Stage-1 pretraining doesn't converge in 60 epochs on the larger MIMIC-IV-ECG corpus	Medium	Early-stop on validation reconstruction; the headline experiments only need a converged motif codebook, not a fully converged loss curve. Can also subset MIMIC-IV-ECG to approx 200k recordings without changing the story.

ROCm-specific PyTorch op gap (e.g. torch.fft edge case, torchdiffq adjoint quirk)	Medium	The lag op and ODE solver are isolated behind two modules with unit tests; CPU fallback exists for the FFT operator, and I can swap dopri5 for rk4 if the adjoint misbehaves on ROCm.
Identifiability theorem doesn't go through cleanly in time	Medium-High	Fallback: empirical lag-recovery on synthetic LD-SEM + zero-shot CHB-MIT result still stand on their own as an empirical paper. The theory becomes 'supporting' rather than headline.
Zero-shot focal-lead result is weak	Medium	Fall back to a self-supervised framing: pretrain + lag-only readout with light fine-tuning on a few seizure subjects. Still novel relative to BIOT/BENDR.
Wall-clock overrun on the M1300X node	Low-Medium	Stage-1 schedule has a train.max_steps cap; can drop to 3 seeds instead of 5 and skip the 100 percent label condition without losing the core claims.
Dataset download stalls from PhysioNet S3	Low	Download script supports --only <dataset> for retries; CHB-MIT and EEGMMIDB are small enough to re-fetch quickly.