Research Statement: Scaling (Scientific) Machine Learning

Yiping Lu (Northwestern University) yiping.lu@northwestern.com

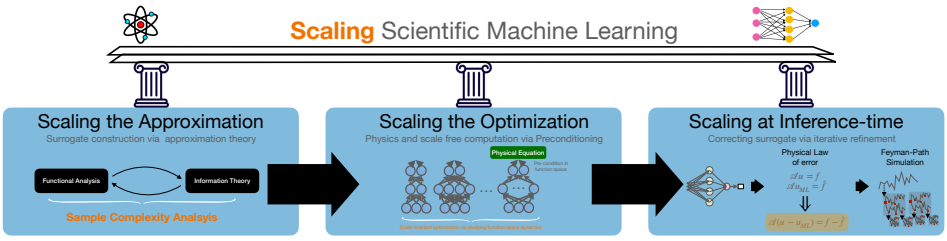
Machine learning has yet to meet the precision required for rigorous scientific and engineering applications, where success depends on combining principled model-based structure—such as physical laws, financial constraints, and operational models—with scalable data-driven inference. At the same time, recent advances in large language models reveal that model performance improves *predictably* as we scale training resources. Empirically, the generalization risk—measured as validation loss, perplexity, or prediction error—follows a power-law decay with respect to data size and compute:

$$\text{Risk} \approx \frac{C_\alpha}{\text{Data}^\alpha} + \frac{C_\beta}{\text{Compute}^\beta} + \varepsilon_{\text{floor}}, \tag{1}$$

where:

- Risk is the model's error on the target distribution.
- Data denotes the effective number of training examples (e.g., number of high-quality text tokens).
- Compute quantifies the training budget (e.g., total FLOPs).
- $\alpha, \beta > 0$ are the *scaling exponents* that determine how efficiently adding more data or more compute reduces error and $C_\alpha, C_\beta > 0$ are problem-dependent constants.
- $\varepsilon_{\text{floor}}$ is the *irreducible error floor*, representing noise, ambiguity, or intrinsic difficulty of the task.

The predictability in (1) fundamentally changes how we design learning systems: instead of guessing whether larger models will help, we can *forecast* performance and allocate resources optimally. Yet, despite the success of scaling laws in language models, current *scientific and engineering* machine learning systems do not exhibit the same predictable improvement when scaled. These systems must integrate *model-based knowledge*—such as physical laws, financial constraints, or operational models—with *data-driven statistical learning*. However, without a reliable scaling law, increasing model size or data does not guarantee improved accuracy or stability; performance often plateaus due to optimization difficulty, stiff constraints, or limited signal in the data.



This gap motivates my research: I aim to build the next generation of scientific machine learning systems whose accuracy scales *predictably* with resources, just as modern language models do. By combining principled model-based structure with scalable data-driven inference, my work seeks to ensure that **"more compute, more data, more model capacity" reliably leads to better predictive accuracy**. Specifically, this vision crystallizes into the following research thrusts.

- **Sample Complexity of Machine Learning with Model-based Structure**. Direct application of machine learning methods is often highly data-intensive. However, data may not always be available due to privacy concerns, changing environments, or other limitations. In contrast, structural models typically encode fundamental laws—often expressed as high-dimensional differential equations—which are generally difficult to solve. Machine learning offers a promising approach to solving these structural models, with the potential to break the curse of dimensionality via Monte Carlo methods. Moreover, structural models are transparent and reliable for counterfactual prediction. A central question is: how can we combine structural modeling with machine learning to simultaneously improve sample efficiency and overcome the curse of dimensionality in high-dimensional problems?
- **Scaling Language Model Optimizer**. Large-scale neural networks often lead to ill-posed optimization problems and complex loss landscapes. A key question is how optimization behaves as network width and depth scale to infinity—i.e., in the mean-field or infinite-width limit—and how to design optimizers whose performance and convergence are independent of network size.
- **Inference-time Scaling Via Stochastic Simulation**. I also investigate a new scaling paradigm called inference-time scaling, which aims to improve model performance via allocating more resource at inference rather than during training. The key insight is that, at inference, we have access to structural models, simulation techniques, or partial observations that were not fully utilized during training. By leveraging

structural models and simulation-based techniques, we aim to correct model outputs during inference, enabling faster and more optimal convergence. The goal is to construct inference-time scaling methods that achieve both efficiency and accuracy.

Below are the directions I am pursuing to build such scalable systems—capable of reaching the level of precision demanded by science, finance, and operations research.

Scaling the Approximation: Statistical Complexity Scaling an ML system first requires a scalable approximation ansatz. Scaling SciML requires identifying the right function spaces—along with their intrinsic complexity—to ensure models remain computationally efficient while expressive enough to capture the underlying physics and structure of scientific systems.

- **Scaling Physics-inspired Architecture Design** My initial research interpreted popular neural networks as numerical discretizations of (stochastic) differential equations. The connection between differential equation and neural network reveals a promising but underexplored opportunity—the integration of physics priors into the ansatz space to enhance approximation quality. My research introduces a unifying dynamical-systems viewpoint for modern neural architectures. In one of the first works to make this connection explicit [20], we show that a ResNet block is exactly the forward Euler discretization of an ODE, where depth corresponds to time and skip connections implement the identity-plus-residual update; stability heuristics such as step-size tuning or Lipschitz regularization naturally emerge from numerical time–stepping. Extending this idea to CNNs, we interpret convolutions as finite-difference stencils approximating differential operators. By enforcing moment-matching constraints on filters [13, 12], our PDE-Net simultaneously (i) predicts the dynamics of complex systems and (ii) discovers the underlying governing PDE. More recently, we further generalize this ODE–PDE perspective to Transformers: viewing tokens as particles evolving in continuous "layer time," self-attention becomes the velocity field of a mean-field interacting particle system [17, 7]. This dynamical interpretation explains emergent clustering, attractors, and token collapse, and provides principled tools—ODE theory, stability analysis, and mean-field PDEs—for understanding depth, robustness, and generalization in large-scale language models.
- **Optimizing Physics-Inspired Neural Architectures via Optimal Control** Building on this viewpoint, we developed new optimization methods by formulating the training of infinitely deep networks as an optimal control problem [29, 19], and further established a new generalization theory based on this continuous-depth perspective [4].
  - *Fast Training via Pontryagin's Maximal Principle.* A key perspective in my research is to interpret deep neural networks as *discretized dynamical systems*. A residual network corresponds to a time-discretized ODE, where the weights $\{\theta_t\}$ act as control variables that steer the state $x_{t+1} = f(x_t, \theta_t)$. Training therefore becomes an **optimal control problem**. The Pontryagin Maximum Principle (PMP) provides: (i) a forward state equation, (ii) a backward costate (adjoint / backpropagation) equation, and (iii) a Hamiltonian maximization rule to update θ_t. [29] shows that applying PMP leads to training updates that maximize the Hamiltonian at each layer, resulting in an optimizer that is only a slight modification of standard SGD. This perspective enables principled optimizer design that combines control-theoretic updates, structure, and preconditioning.
  - *Mean-Field Limit of ResNet.* We extend the differential-equation viewpoint by deriving a *mean-field limit* for deep residual networks [19]. When both the number of residual blocks and the width grow to infinity under proper scaling, the ResNet converges to a continuous-time "shallow ensemble" model in which every local minimizer becomes global. This reduces the analysis of deep ResNets to classical mean-field theory for two-layer networks and yields one of the first global convergence guarantees for multilayer architectures without assuming convexity. Moreover, this continuum limit enables us to analyze how different **width–depth scaling laws** [5] affect the speed of convergence toward the infinite-width/depth regime, providing principles for designing extremely deep yet optimizable architectures.
- **Sample Complexity of Scaling Scientific Machine Learning** We develop optimal sample complexity results for scientific machine learning [16, 8], enabling scalable learning as data increases.
  - **Statistical and Computational Analysis for ML Based PDE Solver** In [16, 15], we focus on a prototype elliptic PDE $\mathcal{L}u = f$. We aim to build an estimaotr for u from random observations $\{(x_i, f(x_i) + \eta)\}_{i=1}^n$ of right hand side function f. We establish the information theoretical lower bounds for learning the equation's solution from sampled data and the first matching upper bound for both (modified version of) Deep Ritz Method (DRM) and Physics Informed Neural Network (PINN). We observed that DRM enlarge the variance of sampling a high-frequency single and a modification is needed to achieve optimal rate. In [15], we explain an implicit acceleration of using a Sobolev norm as the objective function for training. While DRM and PINN can achieve statistically optimality, the proper number of epochs of DRM is larger than the number of PINN when both the data size and the hardness of tasks increase in low dimension.
  - **Statistical Analysis for Operator Learning** In [8], we consider the optimal learning rate for learning a linear operator between two infinite dimensional Hilbert spaces. We provided a novel lower bound

to the literature and showed that multi-level machine learning is essential to achieve an optimal learning rate. This example showed a fundamental difference between infinite dimension machine learning and finite dimension one both in sample complexity and algorithmic design.

– **Statistical Analysis for Unsupervised Learning** One of the most powerful tools in unsupervised learning—including spectral clustering, diffusion maps, and many manifold-learning methods—is to compute eigenvectors of differential operators (e.g., the Laplace–Beltrami operator). Despite their widespread use, existing work rarely analyzes the optimal statistical rate for estimating these eigenvectors from randomly sampled data. In this research thread, we develop new estimators of the Laplacian based on graph Laplacians with higher-order kernels and an RKHS-based approximation of Laplace operators on manifolds. Our theoretical analysis [22] shows that these constructions achieve strictly improved, fully adaptive, and in fact minimax-optimal convergence rates for eigenvector estimation—closing the gap between empirical graph-based methods and the information-theoretic limits of manifold learning.

In parallel, we establish a novel connection between feature geometry and hypothesis space complexity [18], bridging functional analysis and information theory. This provides a theoretical foundation for scaling laws and guides the search for efficient and expressive ansatz spaces in scientific machine learning. We are currently establishing optimality results in PDE simulation, eigenvalue problems, and operator learning, which in turn inspire the development of new algorithms guided by these optimality principles.

Scaling the Optimization: Structure-aware Geometry As networks become wider, the geometry of the loss landscape changes in ways that often worsen its smoothness properties—making gradients sharper and optimization more sensitive to step sizes. A central question is therefore how the landscape geometry scales with network width, and how this scaling translates into computational effort. Empirically, the optimal learning rate depends strongly on model width [26, 27, 28]: a rate tuned for a network with 512 hidden units can lead to divergence or severe slow-down when width increases to 2048. This sensitivity reveals that standard optimizers do not naturally respect architectural scaling. To fully harness the benefits predicted by scaling laws, my research aims to design scaling-aware optimizers whose tuning—especially the learning rate—varies only weakly with width. In other words, I seek algorithms whose optimal hyperparameters transfer robustly across model scales.

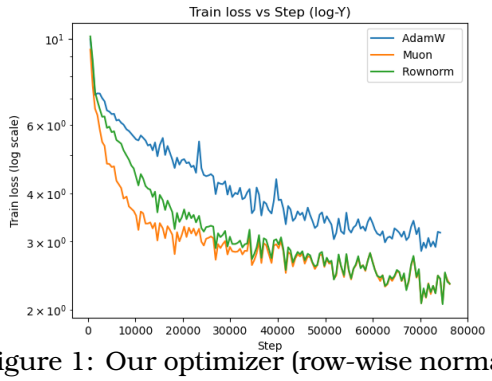


Figure 1: Our optimizer (row-wise normalization only) has the same per-step computational cost as `Adam` on **GPT2-Large**, yet converges significantly faster per iteration, matching `MUON`'s convergence rate without requiring spectral computations.

- **Selecting the Descent Geometry**:We answer this question in the affirmative by viewing existing neural network optimizers, including `SignSGD` [1], `AdamW` [10, 14] and `MUON`[3, 9], in a unified framework as instances of steepest descent under different norms. pecifically, we consider the optimization problem $\min_{\mathbf{W}} f(\mathbf{W})$, where $\mathbf{W}$ denotes the network parameters. Steepest descent can be defined with respect to an arbitrary norm $\|\cdot\|$ with dual $\|\cdot\|_*$. At iterate $\mathbf{W}_k$, a steepest–descent direction is any $\mathbf{D}_k \in \arg\min_{\|\mathbf{D}\|\le 1} \langle \nabla f(\mathbf{W}_k), \mathbf{D}\rangle = -\partial\|\nabla f(\mathbf{W}_k)\|_*$, where $\partial\|\cdot\|_*$ denotes the subdifferential of the dual norm, which is known as `linear minimization oracle` (LMO). In [23], we consider the steepest descent under $(p,\texttt{mean}) \to (q,\texttt{mean})$ geometry $\|\mathbf{D}\|_{(p,\texttt{mean})\to(q,\texttt{mean})} := \sup_{\|\boldsymbol{x}\|_{(p,\texttt{mean})}=1} \|\mathbf{D}\boldsymbol{x}\|_{(q,\texttt{mean})}$, where $\|\boldsymbol{x}\|_{(p,\texttt{mean})} := \left(\frac{1}{n}\sum_{i=1}^n |\boldsymbol{x}_i|^p\right)^{1/p} = n^{-1/p}\|\boldsymbol{x}\|_p$. First of all, we want closed form computable linear minimization oracle which focus our attention to

Norm	LMO	Norm	LMO	Norm	LMO
$(2,\texttt{mean})\to(2,\texttt{mean})$	`MUON`	$(p,\texttt{mean})\to\infty$	Row Normalization	$(1,\texttt{mean})\to(p,\texttt{mean})$	Column Normalization

Under $(p,\texttt{mean}) \to (q,\texttt{mean})$ geometry, the L-smoothness constant is *width-insensitive* precisely when $q \ge 2p$. Otherwise, any residual width dependence is governed by the term $\frac{2}{q}-\frac{1}{p}$, which determines the rate at which the smoothness constant increases with the layer width w. In particular, $(1,\text{mean})\to(p,\text{mean})$ with $p\ge 2$ and $(q,\text{mean})\to\infty$ both yield smoothness that does not grow with width. By contrast, $(2,\text{mean})\to(2,\text{mean})$ (`MUON`) exhibits L-smoothness scaling as $\sqrt{w}$.

The mean-norm operator allows us to express the geometry of a weight matrix $\mathbf{W}\in\mathbb{R}^{w\times d}$ in terms of classical $\ell_1\to\ell_p$ matrix norms. In particular, for the $(1,\text{mean})\to(p,\text{mean})$ operator norm we have $\|\mathbf{W}\|_{(1,\text{mean})\to(p,\text{mean})} = w^{\,1-1/p}\,\|\mathbf{W}\|_{\ell_1\to\ell_p}$, which implies that a gradient update must be scaled by $w^{-1+1/p}$

to keep the step size invariant under width scaling. Similarly, for the $(q, \text{mean}) \to \infty$ operator norm we obtain $\|\mathbf{W}\|_{(q,\text{mean})\to\infty} = w^{1/p} \|\mathbf{W}\|_{\ell_1\to\ell_q}$, so the corresponding update requires a scaling of $w^{-1/q}$. Because smoothness (L–smoothness) of the loss requires $p > 2$, the exponent $1/p$ becomes small and thus the scaling factor $w^{-1/q}$ produces *larger update magnitudes* than $w^{-1+1/p}$. Consequently, the $(q, \text{mean}) \to \infty$ geometry yields stronger and more effective updates, especially in wide networks, while still maintaining width-invariant optimization behavior. In Figure 1, on **GPT-2 Large**, our optimizer achieves `MUON`-level convergence while having the same per-step compute as `Adam` (row-wise normalization only), resulting in faster iteration-wise progress without spectral computations.

- **Understanding Decoupled Weight Decay**: We study the role of decoupled weight decay through the lens of the Norm-Constrained Linear Minimization Oracle (LMO) framework. When the constraint set is chosen as a norm ball that matches the geometry of the LMO oracle, an interesting phenomenon emerges: decoupled weight decay effectively enforces all iterates to remain within a fixed-radius norm ball—equivalently, it transforms the original problem into a constrained optimization problem whose radius is governed by the decay rate. In contrast, applying standard L_2 regularization (defined in Euclidean geometry) introduces a geometric mismatch between the regularizer and the LMO geometry. This misalignment degrades the smoothness properties of the network and exacerbates the width-dependent scaling behavior. Understanding and correcting such geometric inconsistencies is key to designing optimizers that scale reliably with model size.

Algorithm 1: Weight Decay (WD)

Input: Initial parameters θ_0, learning rate η, weight decay λ

for $t = 0, 1, 2, \ldots, T$ **do**

Compute gradient of loss: $g_t = \nabla_\theta \mathcal{L}(\theta_t)$

Weight Decay: $g_t = g_t + \lambda \theta_t$

Parameter update: $\theta_{t+1} = \theta_t - \eta\, \texttt{LMO}(g_t)$

Algorithm 2: Decoupled WD

Input: Initial parameters θ_0, learning rate η, weight decay λ

for $t = 0, 1, 2, \ldots, T$ **do**

Compute gradient of loss: $g_t = \nabla_\theta \mathcal{L}(\theta_t)$

Parameter update: $\theta_{t+1} = \theta_t - \eta\, \texttt{LMO}(g_t)$

Decoupled Weight Decay: $\theta_{t+1} = (1 - \eta\lambda)\, \theta_{t+1}$

Applications:

- **Large Scale Neural Physic Simulator** Larger PINNs are generally harder to train, which hinders their scaling behavior. In [11], we address this challenge by introducing a scale-aware preconditioner that unlocks the scaling behavior of PINN training and achieves machine-precision accuracy. With the scaling law unlocked, we are now scaling PINN computation to 3D turbulence with complex boundaries—settings where traditional finite element and spectral methods often fail.
- **(Online) Low-Rank Approximation via Randomized Preconditioning** In many scientific and engineering applications, the central computational task is to extract the dominant eigenmodes of a (potentially infinite-dimensional) operator—for example, the generator of a dynamical system, or the Hamiltonian in quantum chemistry.
  In [24], we introduce a randomized preconditioning strategy for computing eigenvectors of a large operator $A \in \mathbb{R}^{n\times n}$ using only matrix–vector products. Given a target eigenvalue λ, we study the fixed-point iteration $x_{k+1} = (\lambda I - \hat{A})^{-1}(A - \hat{A})x_k$, where $\hat{A}$ is a randomized low-rank *sketch* of A, constructed from a small number of matrix–vector evaluations. The sketch $\hat{A}$ acts as an effective preconditioner for the shifted-inverse problem: we prove that every eigenvector v satisfying $Av = \lambda v$ is a fixed point of the iteration, *regardless of how $\hat{A}$ is chosen*. Moreover, the quality of $\hat{A}$ directly controls the convergence speed: a more accurate sketch reduces the spectral radius of $(\lambda I - \hat{A})^{-1}(A - \hat{A})$, accelerating contraction toward the desired eigenvector, analogous to classical preconditioned Krylov methods. When $\hat{A}$ is chosen to be low rank, $(\lambda I - \hat{A})$ can be inverted efficiently using the Sherman–Morrison–Woodbury formula, enabling each iteration to be performed using only cheap linear algebra without ever forming or factorizing the full operator A.
  Leveraging this property, we extend the method to an *online* setting: when A represents the Koopman generator of a dynamical system, streaming trajectory data continuously updates $\hat{A}_t$, and the same preconditioned iteration produces the eigenmodes (dynamic modes) *in real time*. This yields an online version of Dynamic Mode Decomposition (DMD) that computes the eigenmodes of the generator directly from data, rather than requiring a full reconstruction of the operator. Finally, this framework naturally applies to quantum systems. I also extend this preconditioned solver—via quantum embedding—toward large-scale quantum chemistry, where the "wavefunction" is precisely the eigenfunction of interest.

Scaling at Inference Time: Simulation-based Calibration Inspired by the recent progress of inference-time scaling in large language models—where models improve simply by allocating more computation at inference (e.g., generating multiple reasoning paths or performing deeper search) without changing model parameters—we advocate a similar principle for scientific simulation: think longer, perform better. Instead of retraining or enlarging the model, we leverage additional computation during inference to refine, correct, and calibrate the model's prediction. In language models, inference-time scaling means letting the model "think longer" (sample more trajectories, perform self-consistency, use Monte-Carlo search), which naturally produces higher-quality outputs. Analogously, in simulation, we let the simulator "work longer" by exploring multiple futures and resampling them according to scientifically grounded criteria. This perspective shifts the focus from model capacity to compute allocation, establishing inference-time computation as a new scaling axis alongside data and parameter scaling. A natural research question comes out

How can we improve the accuracy and reliability of machine-learned simulator at inference time—without any fine-tuning or retraining?

In [6], we improve the accuracy and reliability of a machine-learned (semi-linear) PDE solver at inference time through iterative refinement—without any retraining or fine-tuning—simply by allocating more computation during inference. We introduce a new PDE, the Law of Defect, which quantifies the residual error in the learned surrogate model. Using the Feynman–Kac representation, we launch Monte-Carlo trajectories to estimate this defect and apply corresponding corrections to the model's prediction. From a decision-theoretic perspective, a correct decision should incur zero Bellman error; our algorithm leverages this fact by using the future Bellman error discovered along simulated trajectories to correct the current decision, enabling inference-time correction. Rather than committing to the surrogate model's initial output, the method adaptively improves the solution using compute-scaling rollouts, yielding higher accuracy and greater reliability at deployment time. Here are some application examples:

- ***Online Debiasing Neural Controller.*** Building on our recent framework for physics-informed inference-time refinement [6], we propose an online debiasing neural controller that treats the learned policy not as an oracle but as a proposal to be calibrated. At each space–time point, the controller only needs access to the local solution of the HJB equation to propose a control; rather than executing it blindly, we perform short-horizon rollouts under the learned dynamics to forecast the downstream value and compute the Bellman residual as a principled error signal. This residual exposes where the controller's value approximation violates optimality and, crucially, how to fix it. We then correct the control online by solving a lightweight simulation-time calibration problem: particle rollouts propagate multiple futures, Bellman-error identifies bias, and a small compute budget adjusts the action before it is committed. This design inherits the advantages of inference-time scaling—leveraging extra compute at decision time to buy accuracy—while adding closed-loop reliability: as real observations arrive, a data-assimilation step updates the trajectory weights so that the controller remains robust to model drift and partial observability. In benchmarks on high-dimensional control/PDE tasks, this "simulate-then-correct" loop consistently reduces sub-optimality without retraining, offering a practical path to safe, precise neural control.
- ***Posterior Sampling of Diffusion Model.*** Recent diffusion and flow-based generative models can be interpreted as simulating a stochastic dynamical system. However, existing "guidance" techniques—such as classifier guidance, classifier-free guidance, and energy-based steering—modify the drift of the diffusion process without correcting the induced measure mismatch, often leading to biased and unstable sampling. In our recent work [21], we propose **URGE** (Unbiased Resampling via Girsanov Estimation), the first *inference-time scaling* method that provides unbiased sampling when guiding a pretrained diffusion model toward a task-specific objective. The key idea is to view generation as sampling a path measure and continuously evaluate a "Bellman style error" that detects when the model's predicted trajectory deviates from the desired distribution. Instead of retraining or modifying the score network, URGE performs *particle filtering* (Sequential Monte Carlo) on diffusion trajectories: particles that accumulate higher reward are resampled, while low-performing particles are pruned. This yields an online, inference-time correction loop—"detect error, resample, correct"—that preserves unbiasedness of the target posterior distribution.
  This framework turns diffusion models into a flexible *world simulator*: they can simulate forward dynamics while continuously assimilating new observations through particle filtering. This makes it possible to integrate real-time data streams into a diffusion-based simulator for decision making or forecasting, enabling applications such as model-predictive control, weather prediction, robotics, and scientific data assimilation—all without retraining the base diffusion model.
- ***Backtracking Language Model Reasoing via metropolis hastings.*** A major challenge in test-time scaling for multi-step reasoning is that current methods rely on a verifier (or reward/value model) to guide intermediate steps. However, verifier errors accumulate: a small overestimate early in the chain can

push the reasoning path into an irrecoverable failure mode. I develop a new inference-time correction mechanism [25] that treats reasoning as sampling from a posterior distribution and performs *Metropolis–Hastings backtracking* to retroactively fix earlier mistakes. The key insight is to use the **Bellman error**—the mismatch between the predicted value of the current step and the expected value of its continuation—to *detect when the verifier is wrong*. If the Bellman error indicates overconfidence, we reject the step and roll back to an alternative reasoning trajectory, analogous to backtracking in dynamic programming. The result is an inference-time decoding procedure that suppresses error propagation and ensures that cumulative reasoning quality depends primarily on later (more informed) evaluations of the verifier, rather than compounding errors across the entire chain. Empirically, this leads to significant gains in complex reasoning tasks, even with imperfect reward models.

- ***Statistical Optimality.*** Statistically, in the most stylized setting—computing the mean of an unknown distribution—our analysis in [2] proves that this inference-time correction strategy is minimax optimal, achieving the best possible estimation accuracy given a fixed computational budget, so long as the system does not exhibit rare-event behavior that induces infinite variance. Building on this theoretical foundation, we extend the framework to large-scale machine learning systems. These results reveal a unifying principle: instead of spending more compute in training, we can spend it strategically during inference, correcting errors adaptively based on the real-time signal. I am now pushing this program further—generalizing the statistical theory, developing new sequential estimators for path-space objectives such as diffusion models and scientific simulators, and designing inference-time algorithms that are both theoretically optimal and practically scalable.

My long-term research agenda is to establish a theory of scalable scientific machine learning grounded in optimization, complexity, and resource allocation. I view scaling not as an engineering artifact, but as a fundamental operations research problem: given limited computational budget, data, and model capacity, how do we allocate resources to achieve the maximal statistical gain? This perspective leads to a unifying framework where (i) approximation theory determines what can be learned, (ii) optimizer geometry determines how efficiently we approach the optimum, and (iii) inference-time computation provides an adaptive correction mechanism to guarantee reliability when models deviate from optimality. By treating compute, data, and simulation as decision variables in an optimization problem with provable statistical objectives, I aim to build ML systems whose accuracy is not only high, but predictable and certifiable. Ultimately, I seek to develop a mathematical foundation in which scientific ML behaves like a well-designed OR system—scalable, resource-efficient, and performance-guaranteed.

References

[1] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar. signsgd: Compressed optimisation for non-convex problems. In *International conference on machine learning*, pages 560–569. PMLR, 2018.

[2] J. Blanchet, H. Chen, Y. Lu, and L. Ying. When can regression-adjusted control variate help? rare events, sobolev embedding and minimax optimality. *Advances in Neural Information Processing Systems*, 36:36566–36578, 2023.

[3] D. E. Carlson, E. Collins, Y.-P. Hsieh, L. Carin, and V. Cevher. Preconditioned spectral descent for deep learning. *Advances in neural information processing systems*, 28, 2015.

[4] Y. Chen, F. Liu, Y. Lu, G. G. Chrysos, and V. Cevher. Generalization of scaled deep resnets in the mean-field regime. *arXiv preprint arXiv:2403.09889*, 2024.

[5] L. Chizat. The hidden width of deep resnets: Tight error bounds and phase diagrams. *arXiv preprint arXiv:2509.10167*, 2025.

[6] Z. Fan, Y. Sun, S. Yang, and Y. Lu. Physics-informed inference time scaling via simulation-calibrated scientific machine learning. *arXiv preprint arXiv:2504.16172*, 2025.

[7] B. Geshkovski, C. Letrouit, Y. Polyanskiy, and P. Rigollet. A mathematical perspective on transformers. *Bulletin of the American Mathematical Society*, 62(3):427–479, 2025.

[8] J. Jin, Y. Lu, J. Blanchet, and L. Ying. Minimax optimal kernel operator learning via multilevel training. *arXiv preprint arXiv:2209.14430*, 2022.

[9] K. Jordan, Y. Jin, V. Boza, J. You, F. Cesista, L. Newhouse, and J. Bernstein. Muon: An optimizer for hidden layers in neural networks. *Cited on*, page 10, 2024.

[10] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[11] J. Lai, S. Wang, Y. Lu, and C. Wang. Unveiling the scaling law of pinns through scale aware preconditioning. *submitted*, 2025.

[12] Z. Long, Y. Lu, and B. Dong. Pde-net 2.0: Learning pdes from data with a numeric-symbolic hybrid deep network. *Journal of Computational Physics*, 399:108925, 2019.

[13] Z. Long, Y. Lu, X. Ma, and B. Dong. Pde-net: Learning pdes from data. In *International conference on machine learning*, pages 3208–3216. PMLR, 2018.

[14] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[15] Y. Lu, J. Blanchet, and L. Ying. Sobolev acceleration and statistical optimality for learning elliptic equations via gradient descent. *Advances in Neural Information Processing Systems*, 35:33233–33247, 2022.

[16] Y. Lu, H. Chen, J. Lu, L. Ying, and J. Blanchet. Machine learning for elliptic pdes: Fast rate generalization bound, neural scaling law and minimax optimality. *arXiv preprint arXiv:2110.06897*, 2021.

[17] Y. Lu, Z. Li, D. He, Z. Sun, B. Dong, T. Qin, L. Wang, and T.-Y. Liu. Understanding and improving transformer from a multi-particle dynamic system point of view. *arXiv preprint arXiv:1906.02762*, 2019.

[18] Y. Lu, D. Lin, and Q. Du. Which spaces can be embedded in l_p-type reproducing kernel banach space? a characterization via metric entropy. *arXiv preprint arXiv:2410.11116*, 2024.

[19] Y. Lu, C. Ma, Y. Lu, J. Lu, and L. Ying. A mean field analysis of deep resnet and beyond: Towards provably optimization via overparameterization from depth. In *International Conference on Machine Learning*, pages 6426–6436. PMLR, 2020.

[20] Y. Lu, A. Zhong, Q. Li, and B. Dong. Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations. In *International conference on machine learning*, pages 3276–3285. PMLR, 2018.

[21] C. Wang, W. Wang, Y. Ren, Y. Ying, Lexing, J. Blanchet, and Y. Lu. Simple unbiased inference-time scaling for diffusion models via sequential monte carlo on path measures. 2025.

[22] W. Wang, Y. Zhu, R. Yang, and Y. Lu. Spectral convergence of high-order graph laplacians under smooth densities. 2025.

[23] R. Xu, J. Li, and Y. Lu. On the width scaling of neural optimizers under matrix operator norms i: Making operators play nice together. 2025.

[24] R. Xu and Y. Lu. What is a sketch-and-precondition derivation for low-rank approximation? inverse power error or inverse power estimation? *arXiv preprint arXiv:2502.07993*, 2025.

[25] C. W. Yang, Y. Zhu, and Y. Lu. Inference-time scaling with metropolis–hastings backtracking. 2025.

[26] G. Yang and E. J. Hu. Tensor programs iv: Feature learning in infinite-width neural networks. In *International Conference on Machine Learning*, pages 11727–11737. PMLR, 2021.

[27] G. Yang, E. J. Hu, I. Babuschkin, S. Sidor, X. Liu, D. Farhi, N. Ryder, J. Pachocki, W. Chen, and J. Gao. Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer. *arXiv preprint arXiv:2203.03466*, 2022.

[28] G. Yang, J. B. Simon, and J. Bernstein. A spectral condition for feature learning. *arXiv preprint arXiv:2310.17813*, 2023.

[29] D. Zhang, T. Zhang, Y. Lu, Z. Zhu, and B. Dong. You only propagate once: Accelerating adversarial training via maximal principle. *Advances in neural information processing systems*, 32, 2019.