

A SCALABLE COOPERATIVE/COMPETITIVE SPLITTING SCHEME FOR MIXTURE OF EXPERTS MODELS

Anonymous authors

Paper under double-blind review

ABSTRACT

We present a novel probabilistic expectation-maximization scheme for training hierarchical mixture-of-experts models that both exposes and exploits parallelism during training. By replacing the typical categorical distribution used in gating networks with a joint distribution blending cooperative and competitive mechanisms, we obtain a likelihood that encodes both global and local interactions between experts. The application of an M-splitting scheme reveals an M-step that enables the solution of localized, embarrassingly parallel subproblems governing local experts, with deferred corrections accounting for global coupling between experts. When combined with a hierarchical decomposition of nested networks, this yields a fast multi-level training scheme reminiscent of multigrid algorithms, which avoids under-utilization of experts, exposes further GPU parallelism and outperforms standard models on regression tasks. We provide experiments using a scalable GPU implementation that demonstrate rapid convergence and parallel scalability of the iterative scheme, as well as strong localization of the model for non-smooth, high-dimensional regression problems.

1 INTRODUCTION AND PRIOR WORK

Mixture-of-experts (MoE) models employ a gating network to dynamically route specialized sub-networks ("experts") to perform a given task. When the gating network is sparsely activated, the resulting architecture can be highly efficient, as only a small number of experts are evaluated per input. Following their introduction in the 1990s Jacobs et al. (1991); Jordan and Jacobs (1994), MoE models were initially viewed as a means of incorporating modular, specialized components that performed well across multiple tasks French (1999). In recent years, however, they have attracted attention as a scalable approach to training large models Shazeer et al. (2017), with some works successfully training models with over 100 billion parameters Fedus et al. (2022); Du et al. (2022); Lewis et al. (2021); Lepikhin et al. (2020). The recent success of DeepSeek-V2 also relies crucially on MoE architectures Liu et al. (2024). Whereas large monolithic architectures of comparable size are challenging to train, sparsely gated expert models offer a way to combine easier-to-train component networks.

Despite their scalability, training MoE models remains challenging, particularly with respect to ensuring balanced expert utilization and promoting cooperation among experts. Traditional training methods often struggle with these issues, leading to underutilized experts and suboptimal performance.

In this work, we construct a probabilistic framework that admits a scalable expectation-maximization (EM) training scheme for efficiently training individual experts. We introduce a new approach that combines the predictive accuracy of *cooperative* experts, which blend expert predictions, with the computational efficiency of *competitive* experts, which train independently on disjoint subsets of data. This approach integrates several novel technical contributions:

- A novel probabilistic gating function exposes a decomposition of local and global work by using a mean-field assumption to blend cooperative and competitive effects.
- An expectation-maximization (EM) algorithm, solved via a novel splitting scheme, exposes parallelism in local computations amenable to GPU acceleration while preserving cooperation across experts. A theoretical bound establishes a sufficient condition for convergence which are explored experimentally.

- A hierarchical MoE architecture supports a novel multi-level training scheme that ensures full utilization of all experts. This work builds upon the multilevel training framework introduced in Trask et al. (2022) (see Appendix A for distinctions). While the current method inherits this structure, it introduces a cooperative/competitive split that exposes latent GPU-parallelism.
- Benchmarks demonstrate acceleration from these techniques, as well as the importance of cooperative effects in a physics-informed architecture where a purely competitive expert model provides over $10\times$ less accurate predictions.

For simplicity, we develop the method and accompanying theory in the context of supervised regression using simple dense feedforward expert models. However, the proposed scheme is readily compatible with switch transformers and can be integrated into large-scale models without modification. The benchmarks presented in this work demonstrate strong GPU scalability, establishing the feasibility of extending the approach to large transformer-based architectures in future work.

Probabilistic MoE, EM, and cooperative/competitive experts. Early work by Jordan and Jacobs (1994) demonstrated that MoE models admit a probabilistic interpretation, with the gating network defining a categorical distribution upon which experts are conditioned. By maximizing the evidence lower bound (ELBO), they showed that EM can be used to efficiently train experts *competitively*: the M-step yields posterior-weighted, decoupled subproblems for training each expert independently. They also proposed a *cooperative* loss that fully couples experts during training, improving accuracy at the cost of increased computational expense. More recent works have explored the trade-off between cooperative and competitive training objectives Ahn and Sentis (2021); Do et al. (2025). The present work provides a probabilistic unification of both losses under a single ELBO formulation.

Hierarchical models and multigrid. Our cooperative/competitive split builds on the hierarchical MoE regression framework of Trask et al. (2022), which extends Jordan and Jacobs (1994) by incorporating multigrid-inspired training. In numerical linear algebra, multigrid methods enable $O(N \log N)$ solves Brandt (1977); Briggs et al. (2000) and power exascale simulations Ibeid et al. (2020); Falgout et al. (2021). This motivates our hypothesis that multigrid-trained hierarchical models can scale to similarly large architectures. Recent work has explored multigrid-inspired model training Ke et al. (2017); Albergo et al. (2019); Gunther et al. (2020) and ML-enhanced multigrid solvers Oswald et al. (2023); Taghibakhshi et al. (2023).

Splitting schemes in scientific computing. Splitting schemes are an essential algorithmic ingredient in scientific computing, with seminal works by Chorin (1968) and Strang (1968). Our work leverages splitting methods in numerical linear algebra, like Jacobi and Gauss-Seidel, that were among the first iterative methods for solving linear systems (Saad, 2003; Golub and Van Loan, 2013). Within multigrid methods, they serve a critical numerical role of reducing high-frequency errors (Brandt, 1986). Further, exploiting the parallelization of splitting schemes, like Jacobi, allows multigrid methods to run efficiently on supercomputers Adams et al. (2003); Chow et al. (2006).

Scientific machine learning. Beyond scaling large models, MoEs have seen growing use in ML-based physical modeling—e.g., in fluids Sharma and Shankar (2024); Zigon and Zhu (2025), chemistry Shirasuna et al. (2024), and materials Chang et al. (2022). They are particularly effective for enforcing hard constraints on physics residuals Chalapathi et al. (2024); Actor et al. (2024), where accuracy approaching machine precision are often required. While some address this via problem-specific multistage training Wang and Lai (2024); Ainsworth and Dong (2021); Howard et al. (2023), our method aims to achieve such accuracies directly through optimizer design. We demonstrate this by solving a physics-informed neural network with hierarchical cooperative training—without modifying the PDE—yielding accurate predictions. This scales the earlier findings of Cyr et al. (2020), which showed orders-of-magnitude improvements from cooperative MoE losses but required dense $O(N^3)$ linear solves that hinder scalability.

Code is available on the (anonymized) GitHub https://anonymous.4open.science/r/coopcompsplit_neurips2025-3F27/readme.md.

2 TECHNICAL APPROACH

We summarize here the key algorithmic features to exposing and exploiting parallelism, sketched in Figure 1. For further details and distinctions from Trask et al. (2022), see Appendix A.

MoE models admit a probabilistic interpretation where each expert prescribes a conditional distribution and the gating network defines a categorical distribution selecting a given expert. Let the gating variable $Z(x) \sim \text{Cat}(\pi(x))$, where $\pi(x)$ prescribe the probability of a given input x being assigned to the i -th expert model, i.e., $p(Z_1(x) = i_1) = \pi_{i_1}(x; \theta)$, providing the output distribution

$$p(Y_1(x) = y) = \sum_{i_1=1}^{N_1} p(\mathcal{E}_{i_1} = y | Z_1(x) = i_1) p(Z_1(x) = i_1),$$

We assume a mean-field decomposition of the experts into cooperative and competitive components

$$p(\mathcal{E}_{i_1} = y | Z_1(x) = i_1) = \frac{1}{Q_{i_1}} \mathcal{N}(y; \mu_{i_1}, \sigma_{\text{comp}}^2) \mathcal{N}(y; \hat{y}_1, \sigma_{\text{coop}}^2), \quad (1)$$

where Q_{i_1} is the normalizing factor. Each expert prediction μ_{i_1} and the cooperative output blending all experts $\hat{y}_1$ are given by

$$\mu_{i_1}(x) = \mathbf{c}_{i_1}^\top \mathbf{H}_{i_1}(x; \theta) \quad \text{and} \quad \hat{y}_1(x) = \mathbb{E}[Y_1](x) = \sum_{i_1=1}^{N_1} \pi_{i_1}(x; \theta) \mu_{i_1}(x),$$

respectively. The assumed expert form is consistent with a generic hidden architecture consisting of a hidden layer $\mathbf{H}_{i_1}$ composed with a linear layer $\mathbf{c}_{i_1} \in \mathbb{R}^{N_{\text{basis}}}$. The gating distribution $\pi_{i_1}(x; \theta)$ is parameterized by a neural network with a softmax activation at the output layer. This setting encapsulates a broad range of architectures, including switch transformers; for simplicity, in this work, we will consider ResNet architectures.

In general, this architecture requires training of individual experts to prescribe μ_{i_1} , specification of additive noise σ_{comp}^2 and σ_{coop}^2 , and training of the gating network. We will demonstrate a novel expectation maximization strategy that provides a decoupling of the training for individual experts; specifically, we obtain a coupled system of equations for optimal linear layer weights $\mathbf{c}_{i_1}$. To train these individually, we may reinterpret σ_{comp}^2 and σ_{coop}^2 instead as numerical parameters that may be used to control the relative importance of local and global information, selecting the cooperative contribution to be non-zero but sufficiently small that guarantees can be provided for a splitting scheme.

Hierarchical generalization to arbitrary levels. We extend to a multilevel setting by introducing a hierarchy of latent variables Z_n for each level n defined conditionally on the previous $n - 1$ levels

$$p(Z_n = i_n | Z_1 = i_1, \dots, Z_{n-1} = i_{n-1}) = \pi_{i_n},$$

$$p(Z_1 = i_1, \dots, Z_n = i_n) = \prod_{k=1}^n \pi_{i_k} = \tilde{\pi}_{i_n},$$

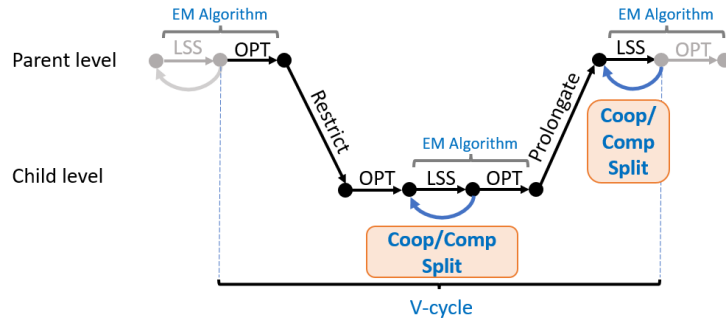


Figure 1: An illustration of the hierarchical training. Details are provided in Appendix A.

$$p(Y_n(\mathbf{x}) = y) = \sum_{I_n} \tilde{\pi}_{I_n} \mathcal{N}(y; \mu_{I_n}, \sigma_{\text{comp}}^2) \mathcal{N}(y; \hat{y}_n, \sigma_{\text{coop}}^2),$$

$$\mu_{I_n}(\mathbf{x}) = \mathbf{c}_{I_n}^\top \mathbf{H}_{I_n}(\mathbf{x}; \theta), \quad \hat{y}_n(\mathbf{x}) = \sum_{I_n} \tilde{\pi}_{I_n}(\mathbf{x}; \theta) \mu_{I_n}(\mathbf{x}).$$

where $I_n = \{i_1, \dots, i_n\}$ is used as a shorthand for all indices up to the current level. In Trask et al. (2022), the authors demonstrate a multi-grid inspired scheme where expectation maximization is used to perform polynomial regression. In the current work, we modify this both by considering arbitrarily deep architectures for experts and using the mean-field distribution in Equation 1. Following Trask et al. (2022), at training time we perform a V-cycle optimization evaluating an EM-step at each level of the hierarchy progressing from coarse to fine, and then applying marginalization of the probabilistic to traverse the hierarchy back from fine to coarse. Details of this and a diagram of training are provided in Appendix A.

When developing a multigrid scheme, a common requirement is that the range of the coarse space is a subset of the fine space, implying that finer scales are well-approximated on coarse-scales. Formally,

$$\text{span}(\mathbf{H}_{i_1, \dots, i_n}) \subseteq \text{span}(\mathbf{H}_{i_1, \dots, i_n, i_{n+1}}). \quad (2)$$

We design architectures that achieve this by constructing children of parent experts which consist of their parents hidden layer stacked with a new hidden layer, so that

$$\mu_{I_{n+1}}(\mathbf{x}) = \mathbf{c}_{I_{n+1}}^\top \begin{bmatrix} \mathbf{H}_{I_n}(\mathbf{x}; \theta) \\ \tilde{\mathbf{H}}_{n+1}(\mathbf{x}; \theta) \end{bmatrix},$$

where $\tilde{\mathbf{H}}_{n+1}(\mathbf{x}; \theta)$ is a new hidden layer architecture. This construction provides experts of increasing nested complexity as the hierarchy is extended more deeply and exposes parallelism.

Cooperative/competitive EM update. We next demonstrate how the assumed mean-field approximation impacts the standard EM update from Jordan and Jacobs (1994). The observed data log likelihood is given by

$$\log L(\theta; \mathcal{D}) = \sum_{d=1}^{N_d} \log \left[\sum_{I_n} \tilde{\pi}_{I_n}(\mathbf{x}^d; \theta) \mathcal{N}(y^d; \mu_{I_n}(\mathbf{x}^d), \sigma_{\text{comp}}^2) \mathcal{N}(y^d; \hat{y}_n(\mathbf{x}^d), \sigma_{\text{coop}}^2) \right],$$

which, by Jensen’s inequality, is bounded from below by

$$\ell(\theta) = \sum_{d=1} \sum_{I_n} w_{I_n}(\mathbf{x}^d) \log \frac{\tilde{\pi}_{I_n}(\mathbf{x}^d; \theta) \mathcal{N}(y^d; \mu_{I_n}(\mathbf{x}^d), \sigma_{\text{comp}}^2) \mathcal{N}(y^d; \hat{y}_n(\mathbf{x}^d), \sigma_{\text{coop}}^2)}{w_{I_n}(\mathbf{x}^d)}.$$

We choose $w_{I_n}(\mathbf{x}^d)$ such that the ELBO is a tight lower bound,

$$w_{I_n}(\mathbf{x}^d) := p(Z_1 = i_1, \dots, Z_n = i_n \mid Y_n = y^d) = \frac{\tilde{\pi}_{I_n} \mathcal{N}(y^d; \mu_{I_n}(\mathbf{x}^d), \sigma_{\text{comp}}^2)}{\sum_{J_n} \tilde{\pi}_{J_n} \mathcal{N}(y^d; \mu_{J_n}(\mathbf{x}^d), \sigma_{\text{comp}}^2)}.$$

Computing $w_{I_n}(\mathbf{x}^d)$ prescribes the E-step of each iteration. In the M-step, we find the optimal parameters $\mathbf{c}_{I_n}$, $\mathbf{H}_{I_n}$ and $\tilde{\pi}_{I_n}$ to maximize the ELBO. Taking the derivative of the ELBO with respect to the expert coefficients $\mathbf{c}_{I_n}$ yields the weighted least-squares problem

$$\sum_{d, J_n, \beta} \left(\frac{w_{I_n}^d \delta_{I_n J_n}}{\sigma_{\text{comp}}^2} + \frac{\tilde{\pi}_{I_n}^d \tilde{\pi}_{J_n}^d}{\sigma_{\text{coop}}^2} \right) H_{I_n, \alpha}^d H_{J_n, \beta}^d c_{J_n, \beta} = \sum_d \left(\frac{w_{I_n}^d}{\sigma_{\text{comp}}^2} + \frac{\tilde{\pi}_{I_n}^d}{\sigma_{\text{coop}}^2} \right) y^d H_{I_n, \alpha}^d \quad (\text{LS})$$

where $\delta_{I_n J_n}$, a generalization of the Kronecker delta, equals 1 if and only if all corresponding components of I_n and J_n are the same, and equals 0 otherwise. Solving this weighted least-square problem requires a dense matrix solve; in the following section we demonstrate how this can be scalably solved.

$\mathbf{H}_{I_n}$ and $\tilde{\pi}_{I_n}$ are optimized with a gradient descent loss

$$\mathcal{L}_{\text{GD}} = -\ell(\theta)$$

By marginalizing the posterior of the n level, we can obtain another estimator for the $n - 1$ level, i.e.,

$$\hat{w}_{l_{n-1}} = \sum_{i_n} w_{l_n}.$$

Note that the summation is performed only on the last index i_n . Thus, we obtain a hierarchical sequence of M step solves traversing up the hierarchy, accessing information at each level from the solve of the previous level. See Appendix A for a worked two-level example.

3 SPLITTING SCHEME FOR PARALLELIZABLE ITERATIVE SOLVER

We can split the tensor on the left-hand side of equation LS into the competitive component tensor $\mathbf{M}$ and the cooperative component tensor $\mathbf{N}$ defined by

$$\begin{aligned} \mathbf{M}_{\mathbf{IJ}} &= \sum_d \frac{w_{l_n}^d \delta_{l_n l_n}}{\sigma_{\text{comp}}^2} H_{l_n, \alpha}^d H_{l_n, \beta}^d, \\ \mathbf{N}_{\mathbf{IJ}} &= - \sum_d \frac{\tilde{\pi}_{l_n}^d \tilde{\pi}_{l_n}^d}{\sigma_{\text{coop}}^2} H_{l_n, \alpha}^d H_{l_n, \beta}^d, \end{aligned}$$

where $\mathbf{I} = \{l_n, \alpha\}$ and $\mathbf{J} = \{l_n, \beta\}$, allowing us to matricize these tensors and to vectorize $\mathbf{c}$ by flattening each vector $\mathbf{I}$ and $\mathbf{J}$ into a single index. While $\mathbf{N}$ is a dense matrix, $\delta_{l_n l_n}$ makes $\mathbf{M}$ block-diagonal, which allows the computation of its inverse to be parallelized efficiently. Exploiting this structure, we replace the direct solution of the linear system with an iterative scheme of the form

$$\mathbf{M}\mathbf{c}^{(k+1)} = \mathbf{N}\mathbf{c}^{(k)} + \mathbf{b}. \quad (\text{I})$$

Theorem 1. Assume that the experts are linearly independent with respect to the $\mathbf{M}$ and $\mathbf{N}$ -weighted inner-products, i.e. for $l_n \neq l_n$,

$$\det \left(\frac{\tilde{\pi}_{l_n}^d \tilde{\pi}_{l_n}^d}{\sigma_{\text{coop}}^2} H_{l_n, \alpha}^d H_{l_n, \beta}^d \right) \neq 0, \quad \det \left(\sum_d \frac{w_{l_n}^d \delta_{l_n l_n}}{\sigma_{\text{comp}}^2} H_{l_n, \alpha}^d H_{l_n, \beta}^d \right) \neq 0,$$

then the spectral radius $\rho(\mathbf{M}^{-1}\mathbf{N}) < 1$ provides a sufficient condition for convergence. Under these conditions, choosing $R_\sigma = \sigma_{\text{coop}}^2 / \sigma_{\text{comp}}^2$ as

$$R_\sigma^* \geq \frac{\max_i \sum_d \tilde{\pi}_{l_n}^d}{\min_i \sum_d w_{l_n}^d}. \quad (3)$$

guarantees convergence of the splitting scheme.

In Appendix B we provide a proof of this result, which follows from a generalized Rayleigh quotient analysis and the Gerschgorin circle theorem (Gerschgorin, 1931). In practice, R_σ may be either selected adaptively following equation 3, or it may be treated as a hyperparameter to be fixed to a sufficiently large value before training.

With minor modification the Successive Over-Relaxation (SOR) method Young (1954) may be adopted to further accelerate solution. At iteration $k + 1$, the update rule is modified via

$$\mathbf{c}^{(k+1)} = (1 - \omega)\mathbf{c}^{(k)} + \omega\mathbf{M}^{-1}(\mathbf{N}\mathbf{c}^{(k)} + \mathbf{b}),$$

where $\omega \in (0, 2)$ a tunable parameter that recovers Theorem 1 for $\omega = 1$.

4 NUMERICAL EXPERIMENTS

In our experiments, we first present a pedagogical regression task to illustrate how the hierarchical basis enhances localization across scales. We then benchmark GPU scalability, demonstrating efficient implementation of embarrassingly parallel weighted least squares solves, as well as less

obvious parallelism via JAX’s `vmap`. Next, we assess the tightness and practical impact of the bound in Theorem 1, along with an ablation study showing robustness to data dimensionality. Finally, we include a scientific machine learning example in which incorporating cooperative effects yields an order-of-magnitude improvement when solving a numerically stiff partial differential equation.

Details of training data, hyperparameters, architecture, and a link to reproducible code are provided in Appendix D.

4.1 ILLUSTRATIVE 1D REGRESSION EXAMPLE

In Figure 2, we regress the function $y(x) = \exp\left[-\frac{1}{2}\left(\frac{x-0.5}{0.05}\right)^2\right]$ on the unit interval. For this problem, we observe that the hierarchical experts allow for an unsupervised concentration of resolution around the relevant feature of the problem. While several works have constructed multistage function approximators by hand (see e.g. Wang and Lai (2024)), this example highlights how hierarchy, cooperative gating, and optimal EM-updates combine to provide highly accurate function approximation.

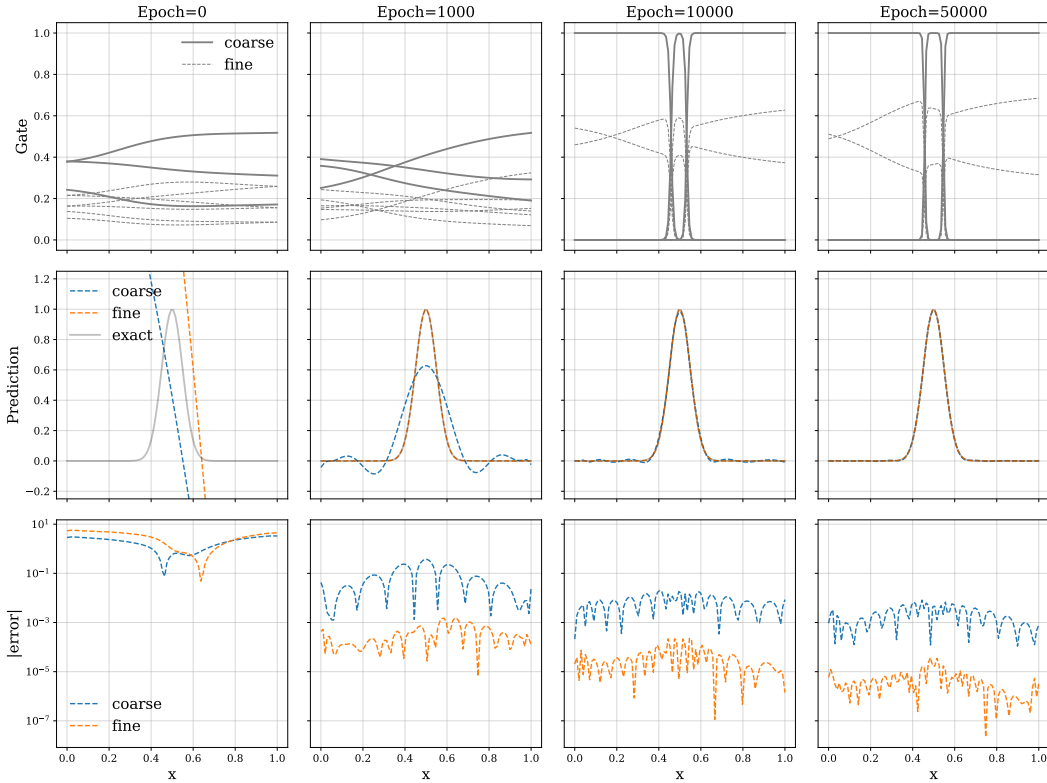


Figure 2: Regression of a Gaussian bump, highlighting unsupervised expert localization, specialization, hierarchical approximation and improved accuracy across levels. **Top:** Gating functions localize on bump, **Middle:** Upon localizing gates, experts provide refined approximation, **Bottom:** Experts deeper in hierarchy can specialize and provide orders of magnitude improved accuracy.

4.2 STRONG GPU SCALING OF BLENDED LEAST-SQUARES SOLVE

We consider a 64-dimensional regression problem using N_c coarse and N_f fine experts, benchmarking acceleration across 1, 2, and 4 GPUs relative to a single-GPU baseline. Figure 3 highlights two forms of GPU-parallelism exposed by our scheme. First, the M-step splitting decouples the global least-squares problem into many small, embarrassingly-parallel solves, yielding strong scaling. Second, the nested hierarchy exposes expert-level decoupling across levels, which JAX’s `vmap` efficiently parallelizes. Appendix A details the mathematical structure behind this decoupling. Speedup is measured against a naive LU-based single-GPU solve of the monolithic system. As expected, the

split formulation achieves near-ideal scaling: $1.97\times$ and $3.79\times$ with 2 and 4 GPUs, respectively. When `vmap` is activated, we observe up to $30\times$ speedup; this improvement is directly attributable to the combination of splitting and hierarchy which allows a complete utilization of GPUs.

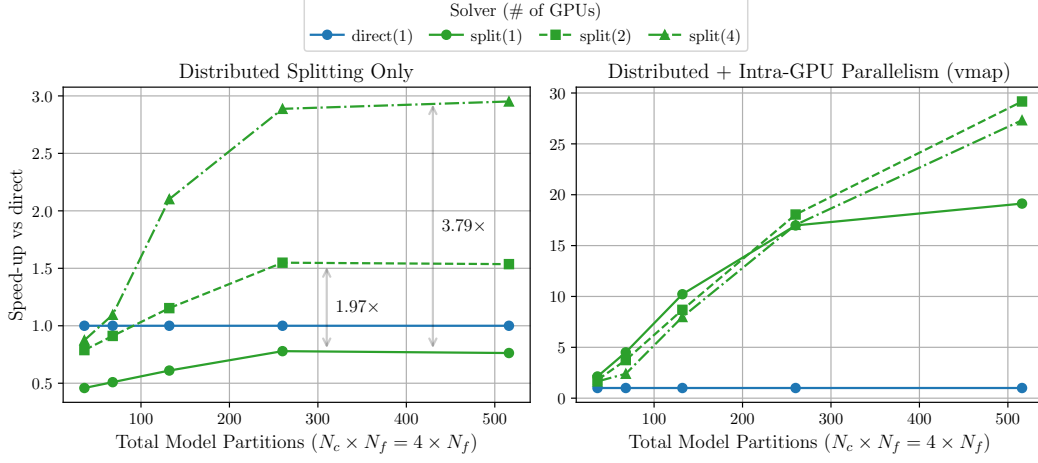


Figure 3: Strong scaling for an iteration of the split linear solver. **Left.** Distributed splitting with no intra-GPU parallelism achieves near ideal speedup. **Right.** JAX’s `vmap` exploits intra-GPU parallelism exposed by splitting and model hierarchy.

4.3 TIGHTNESS OF BOUND IN THEOREM 1 AND PRACTICAL IMPLICATIONS

The fundamental assumption of the method is that R_σ may be chosen small enough to incorporate cooperative effects but large enough to guarantee convergence of the splitting. The bound for R_σ^* in Theorem 1 makes several assumptions, raising the question of whether it is tight enough to provide practical significance. **1.** The experts are assumed linearly independent, which is particularly unlikely at initialization (see e.g. Cyr et al. (2020) for discussion of degenerate rank at initialization time), **2.** Expert posterior collapse may lead to an explosion in R_σ^* as $w_{id} \rightarrow 0$ (See discussion in Appendix B), and **3.** the derived using Gerschgorin is too pessimistic to provide a tight and predictive estimate. We design an experiment to explore two hypotheses: **1.** There exists an experimentally calibrated constant C such that choosing $R_\sigma = C R_\sigma^*$ gives an effective threshold for guaranteeing convergence; and **2.** at the conclusion of training the experts are linearly independent so that R^* is tight.

To test, we construct an artificial scenario in which we generate several different training runs corresponding to a range of R_σ values spanning 1 to 10^{10} . Define N_{split} as the number of iterations for the splitting scheme to converge to a given tolerance. At each epoch, we perform an ablation study fixing the matrices A and B and sweeping over different values of σ_{coop} and σ_{comp} to identify the dependence of N_{split} on alternate weightings of M and N . This provides a realistic range of possible weightings which can illustrate whether the bound in Theorem 1 is tight. We arbitrarily impose a maximum iteration count of 2000 as an indicator of divergence.

Figure 4 illustrates the results of this study. By plotting N_{split} as a function of R_σ/R_σ^* , we obtain a scatter plot with a clear delineation consistent with selecting $C = 1e4$. Realistically C may be problem dependent, but this suggests a practical strategy where C may be gradually increased for a given problem until it is sufficiently large. Interestingly however, when plotting a range of values at the end of training we observe that the transition to convergence occurs at the theory-predicted $C = 1$, suggesting that the bound is tight at the conclusion of training.

4.4 INSENSITIVITY OF SPLITTING SCHEME TO PROBLEM DIMENSION

We consider a regression problem on the d -dimensional unit hypercube, regressing a multivariate Gaussian problem with mean at the center of the cube and standard deviation $\frac{1}{10}$. Similar to Section 4.1, this provides a test whether the gating function can localize on a simple feature in high-dimensions,

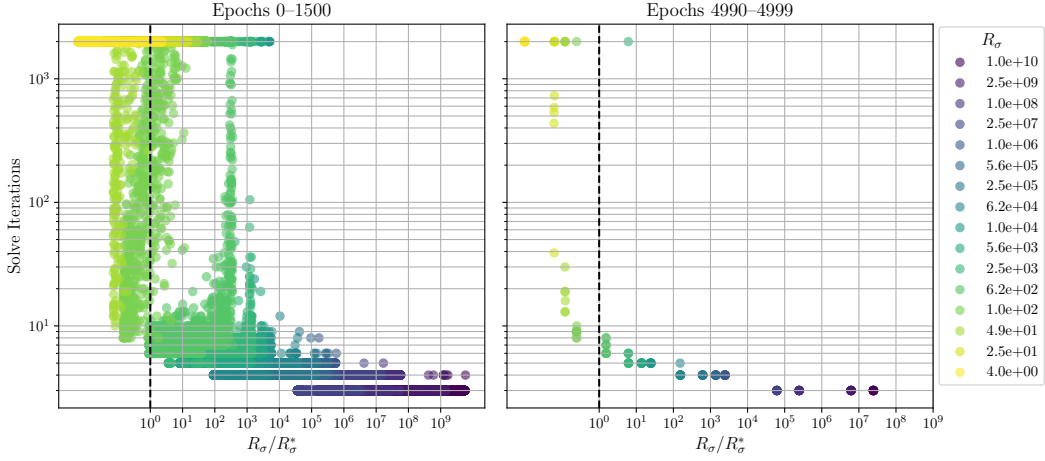


Figure 4: Ablation study exploring the tightness and practical implications of Theorem 1. **Left.** A scatter plot of the number of splitting iterations to convergence as a function of R_σ/R_σ^2 illustrates a sharp transition at $R_\sigma/R_\sigma^2 = 10^4$. **Right.** At the conclusion of training, we see that the predicted bound $R_\sigma = R_\sigma^*$ denoted by a vertical dashed line clearly indicates the threshold of stability, suggesting that the assumptions of Theorem 1 are valid.

and indicating whether the split cooperative/competitive scheme may realistically be deployed on high-dimensional problems.

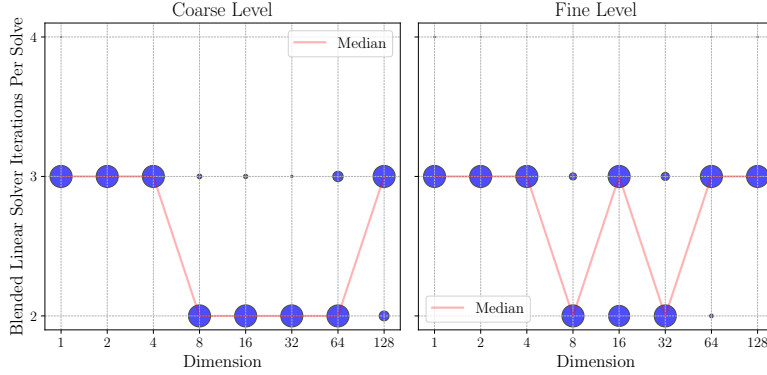


Figure 5: Ablation study illustrating insensitivity of convergence for splitting scheme to problem dimension. Radius of blue circles denotes the frequency of convergence for a given dimension for both coarse (**Left**) and fine (**Right**) levels of hierarchy. Performance is independent of dimension, illustrating suitability for high-dimensional machine learning tasks.

4.5 PERFORMANCE ON A PHYSICS-INFORMED NEURAL NETWORK

Physics-informed neural networks (PINNs) offer a simple case where neural networks provide candidate solutions that minimize a partial differential equation (PDE) residual Lagaris et al. (1998); Raissi et al. (2019). Many works have established pathologies in the training of PINNs Wang et al. (2021); Krishnapriyan et al. (2021); Fuks and Tchelepi (2020). While there are multiple challenges, one is that solutions to PDEs have strict regularity requirements on continuity, which many have shown may be avoided by using more advanced PDE discretizations Yu et al. (2018); Patel et al. (2022). We show in Figure 6 that an application of a cooperative expert to the original "vanilla" scheme is sufficient to achieve results without modification to the original scheme and only via choice of architecture/optimizer. We consider as a benchmark the singularly perturbed advection-diffusion equation, which in the limit as transport becomes advection-dominated, exhibits many sharp

gradients that are challenging even for mature PDE-solution techniques Roos (2008). This regularity is preserved by cooperative experts, but competitive experts are unable to resolve it. This problem requires a non-trivial extension of the framework to a multi-objective loss (See Appendix C) that provides a concrete example of how the framework may be extended to a broader class of problems.

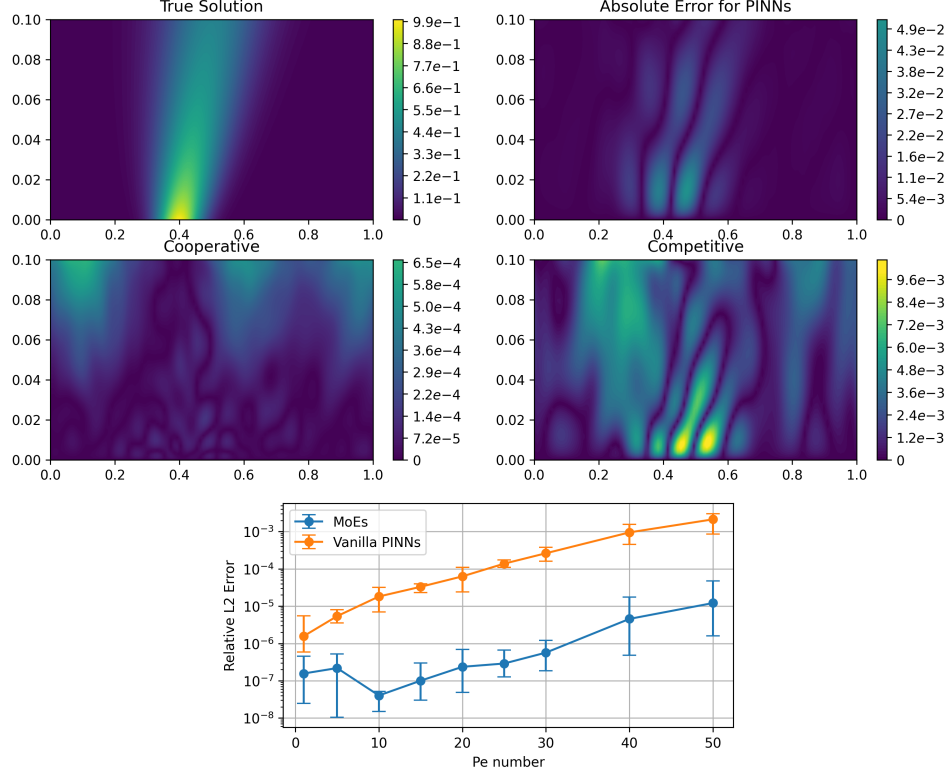


Figure 6: **Extension to multi-objective loss: Physics-informed neural networks (PINNs).** **Top.** In the advection-diffusion problem, High-Péclet cases yield steep gradients and pathological behavior while requiring continuity. Compared to the analytic solution (**Top-left**), standard PINNs exhibit large errors at steep gradients (**Top-right**), which are mitigated by the cooperative scheme (**Bottom-left**). Competitive experts improve training stability but yield solutions with $10\times$ larger error (**Bottom-right**). **Bottom.** An ablation over Péclet numbers in 1D shows the cooperative scheme consistently maintains $> 10\times$ lower error as the problem becomes more singular. Error bars indicate min/max error across five random seeds.

5 CONCLUSION AND FUTURE WORK

We introduced a novel splitting scheme for cooperative/competitive hierarchical mixture-of-experts models and provided analytical criteria guaranteeing robust performance. On simple tasks, the method exposes new forms of GPU parallelism, yielding effective results for both regression and multi-objective optimization, with performance largely insensitive to input dimension. While scaling this scheme to large transformer-based architectures remains a substantial engineering challenge, the present work establishes a clear proof of concept and a promising foundation for future development.

REFERENCES

- Jonas A Actor, Xiaozhe Hu, Andy Huang, Scott A Roberts, and Nathaniel Trask. Data-driven whitney forms for structure-preserving control volume analysis. *Journal of Computational Physics*, 496: 112520, 2024.
- Mark Adams, Marian Brezina, Jonathan Hu, and Ray Tuminaro. Parallel multigrid smoothing: polynomial versus gauss–seidel. *Journal of Computational Physics*, 188(2):593–610, 2003.
- Junhyeok Ahn and Luis Sentis. Nested mixture of experts: Cooperative and competitive learning of hybrid dynamical system. In *Learning for Dynamics and Control*, pages 779–790. PMLR, 2021.
- Mark Ainsworth and Justin Dong. Galerkin neural networks: A framework for approximating variational equations with error control. *SIAM Journal on Scientific Computing*, 43(4):A2474–A2501, 2021.
- Michael S Albergo, Gurtej Kanwar, and Phiala E Shanahan. Flow-based generative models for markov chain monte carlo in lattice field theory. *Physical Review D*, 100(3):034515, 2019.
- Achi Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of computation*, 31(138):333–390, 1977.
- Achi Brandt. Algebraic multigrid theory: The symmetric case. *Applied mathematics and computation*, 19(1-4):23–56, 1986.
- William L Briggs, Van Emden Henson, and Steve F McCormick. *A multigrid tutorial*. SIAM, 2000.
- Nithin Chalapathi, Yiheng Du, and Aditi Krishnapriyan. Scaling physics-informed hard constraints with mixture-of-experts. *arXiv preprint arXiv:2402.13412*, 2024.
- Rees Chang, Yu-Xiong Wang, and Elif Ertekin. Towards overcoming data scarcity in materials science: unifying models and datasets with a mixture of experts framework. *npj Computational Materials*, 8(1):242, 2022.
- Alexandre Joel Chorin. Numerical solution of the navier-stokes equations. *Mathematics of computation*, 22(104):745–762, 1968.
- Edmond Chow, Robert D Falgout, Jonathan J Hu, Raymond S Tuminaro, and Ulrike Meier Yang. A survey of parallelization techniques for multigrid solvers. *Parallel processing for scientific computing*, pages 179–201, 2006.
- Eric C Cyr, Mamikon A Gulian, Ravi G Patel, Mauro Perego, and Nathaniel A Trask. Robust training and initialization of deep neural networks: An adaptive basis viewpoint. In *Mathematical and Scientific Machine Learning*, pages 512–536. PMLR, 2020.
- Giang Do, Hung Le, and Truyen Tran. Sparse mixture of experts as unified competitive learning. *arXiv preprint arXiv:2503.22996*, 2025.
- Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. Glam: Efficient scaling of language models with mixture-of-experts. In *International conference on machine learning*, pages 5547–5569. PMLR, 2022.
- Robert D Falgout, Ruipeng Li, Björn Sjögreen, Lu Wang, and Ulrike Meier Yang. Porting hypre to heterogeneous computer architectures: Strategies and experiences. *Parallel Computing*, 108: 102840, 2021.
- William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.
- Robert M French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128–135, 1999.

- Olga Fuks and Hamdi A Tchelepi. Limitations of physics informed machine learning for nonlinear two-phase transport in porous media. *Journal of Machine Learning for Modeling and Computing*, 1(1), 2020.
- S. Gerschgorin. Über die abgrenzung der eigenwerte einer matrix. *Izvestija Akademii Nauk SSSR, Serija Matematika*, 7(3):749–754, 1931.
- Gene H Golub and Charles F Van Loan. *Matrix computations*. JHU press, 2013.
- Stefanie Gunther, Lars Ruthotto, Jacob B Schroder, Eric C Cyr, and Nicolas R Gauger. Layer-parallel training of deep residual neural networks. *SIAM Journal on Mathematics of Data Science*, 2(1): 1–23, 2020.
- Amanda A Howard, Sarah H Murphy, Shady E Ahmed, and Panos Stinis. Stacked networks improve physics-informed training: applications to neural networks and deep operator networks. *arXiv preprint arXiv:2311.06483*, 2023.
- Huda Ibeid, Luke Olson, and William Gropp. Fft, fmm, and multigrid on the road to exascale: Performance challenges and opportunities. *Journal of Parallel and Distributed Computing*, 136: 63–74, 2020.
- Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.
- Michael I Jordan and Robert A Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 6(2):181–214, 1994.
- Tsung-Wei Ke, Michael Maire, and Stella X Yu. Multigrid neural architectures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6665–6673, 2017.
- Aditi Krishnapriyan, Amir Gholami, Shandian Zhe, Robert Kirby, and Michael W Mahoney. Characterizing possible failure modes in physics-informed neural networks. *Advances in neural information processing systems*, 34:26548–26560, 2021.
- Isaac E Lagaris, Aristidis Likas, and Dimitrios I Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE transactions on neural networks*, 9(5):987–1000, 1998.
- Kookjin Lee, Nathaniel A Trask, Ravi G Patel, Mamikon A Gulian, and Eric C Cyr. Partition of unity networks: deep hp-approximation. *arXiv preprint arXiv:2101.11256*, 2021.
- Dmitry Lepikhin, HyounJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020.
- Mike Lewis, Shruti Bhosale, Tim Dettmers, Naman Goyal, and Luke Zettlemoyer. Base layers: Simplifying training of large, sparse models. In *International Conference on Machine Learning*, pages 6265–6274. PMLR, 2021.
- Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Daya Guo, et al. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024.
- Dominik Oswald, Andrew Trask, and Adrian Sandu. Learning multigrid solvers with graph neural networks. *SIAM Journal on Scientific Computing*, 45(2):A819–A844, 2023.
- Ravi G Patel, Indu Manickam, Nathaniel A Trask, Mitchell A Wood, Myoungkyu Lee, Ignacio Tomas, and Eric C Cyr. Thermodynamically consistent physics-informed neural networks for hyperbolic systems. *Journal of Computational Physics*, 449:110754, 2022.
- Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

- Hans-Görg Roos. *Robust numerical methods for singularly perturbed differential equations*. Springer, 2008.
- Yousef Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.
- Ramansh Sharma and Varun Shankar. Ensemble and mixture-of-experts deepnets for operator learning. *arXiv preprint arXiv:2405.11907*, 2024.
- Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarczyk, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.
- Victor Yukio Shirasuna, Eduardo Soares, Emilio Vital Brazil, Karen Fiorella Aquino Gutierrez, Renato Cerqueira, Seiji Takeda, and Akihiro Kishimoto. A multi-view mixture-of-experts based on language and graphs for molecular properties prediction. In *ICML 2024 AI for Science Workshop*, 2024.
- Gilbert Strang. On the construction and comparison of difference schemes. *SIAM Journal on Numerical Analysis*, 5(3):506–517, 1968. doi: 10.1137/0705041. URL <https://doi.org/10.1137/0705041>.
- Ali Taghibakhshi, Nicolas Nytko, Tareq Uz Zaman, Scott MacLachlan, Luke Olson, and Matthew West. Mg-gnn: Multigrid graph neural networks for learning multilevel domain decomposition methods. In *International Conference on Machine Learning*, pages 33381–33395. PMLR, 2023.
- Nathaniel Trask, Amelia Henriksen, Carianne Martinez, and Eric Cyr. Hierarchical partition of unity networks: fast multilevel training. In *Mathematical and Scientific Machine Learning*, pages 271–286. PMLR, 2022.
- Sifan Wang, Yujun Teng, and Paris Perdikaris. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing*, 43(5):A3055–A3081, 2021.
- Yongji Wang and Ching-Yao Lai. Multi-stage neural networks: Function approximator of machine precision. *Journal of Computational Physics*, 504:112865, 2024.
- David Young. Iterative methods for solving partial difference equations of elliptic type. *Transactions of the American Mathematical Society*, 76(1):92–111, 1954.
- Bing Yu et al. The deep ritz method: a deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics*, 6(1):1–12, 2018.
- Bob Zigon and Luoding Zhu. Modeling 2d unsteady flows at moderate reynolds numbers using a 3d convolutional neural network and a mixture of experts. *Computer Physics Communications*, page 109540, 2025.

A DETAILS OF ALGORITHM AND DISTINCTION FROM TRASK ET AL. 2022

For specificity, we provide here the individual steps of a two-level scheme, illustrating the operations performed in a single V-cycle in Figure 7. We define operations consisting of *OPT*, *Restrict*, *LSS* and *Prolongate* appearing in the figure. For further derivation, we direct the interested reader to Trask et al. (2022), and provide a Github for a scalable GPU implementation at https://anonymous.4open.science/r/coopcompsplit_neurips2025-3F27/readme.md.

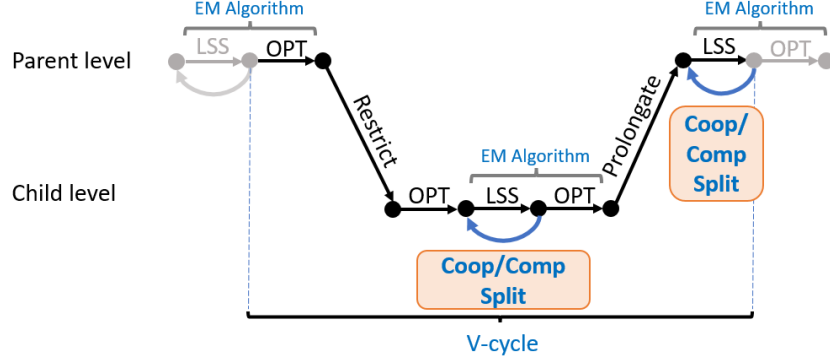


Figure 7: An illustration of the hierarchical training for a simple two level scheme. The key contribution of this work is the introduction of the cooperative/competitive splitting which allows expensive least square solves (LSS) to be distributed in an embarrassingly parallel manner and allowing gradient updates (OPT) to be performed.

Restrict/Prolongate blocks: In the EM algorithm we follow Trask et al. (2022) where the standard evidence lower bound derivation of expectation maximization yields closed form expressions for the posterior distribution of the gating function conditioned on the data. For coarse and fine scales, we obtain the expressions

$$w_{id} := p(Z_1 = i | Y_1 = y_d) = \frac{\pi_i(\mathbf{x}_d, \theta) \mathcal{N}(y_d; \mu_i(\mathbf{x}_d), \sigma_i^2 \mathbf{I})}{\sum_I \pi_I(\mathbf{x}_d) \mathcal{N}(y_d; \mu_I(\mathbf{x}_d), \sigma_I^2 \mathbf{I})}, \quad (4)$$

$$w_{ijd} := p(Z_1 = i, Z_2 = j | Y_2 = y_d) = \frac{\pi_i(\mathbf{x}_d, \theta) \pi_{ij}(\mathbf{x}, \theta_i) \mathcal{N}(y_d; \mu_{ij}(\mathbf{x}_d), \sigma_{ij}^2 \mathbf{I})}{\sum_{I,J} \pi_I(\mathbf{x}_d) \pi_{IJ}(\mathbf{x}, \theta_I) \mathcal{N}(y_d; \mu_{IJ}(\mathbf{x}_d), \sigma_{IJ}^2 \mathbf{I})}, \quad (5)$$

We note that the coarse posterior w_{id} can be evaluated independently of the fine level, and then sequentially used to evaluate the fine level posterior w_{ijd} . Following the nomenclature in multigrid methods, we refer to this evaluation of w_{ijd} from w_{id} as a *restriction*. Then, to propagate information from the child to parent the following marginalization formula is used

$$\hat{w}_{id} := p(Z_1 = i | Y_2 = y_d) = \frac{\sum_j \pi_i(\mathbf{x}_d) \pi_{ij}(\mathbf{x}, \theta_i) \mathcal{N}(y_d; \mu_{ij}(\mathbf{x}_d), \sigma_{ij}^2 \mathbf{I})}{\sum_{I,j} \pi_I(\mathbf{x}_d) \pi_{IJ}(\mathbf{x}, \theta_I) \mathcal{N}(y_d; \mu_{IJ}(\mathbf{x}_d), \sigma_{IJ}^2 \mathbf{I})}. \quad (6)$$

Again, motivated multigrid nomenclature, we refer to this marginalization as *prolongation*.

LSS block. At each scale a posterior weighted least square solve prescribes optimal values for final linear layers of expert models. In Trask et al. (2022) these least squares problems are defined as:

$$\sum_{d=1}^{N_d} w_{id} \Phi_\alpha(\mathbf{x}_d) \Phi_\beta(\mathbf{x}_d) c_{i,\beta} = \sum_{d=1}^{N_d} w_{id} \Phi_\alpha(\mathbf{x}_d) y_d, \quad (7)$$

$$\sum_{d=1}^{N_d} w_{ijd} \Phi_\alpha(\mathbf{x}_d) \Phi_\beta(\mathbf{x}_d) c_{ij,\beta} = \sum_{d=1}^{N_d} w_{ijd} \Phi_\alpha(\mathbf{x}_d) y_d. \quad (8)$$

In the current work, we replace both with the hierarchical cooperative/competitive loss defined in LS.

Coop/comp split block. In the literature, the incorporation of optimal layers requires choosing between a dense solve in the LSS block for a purely cooperative formulation Lee et al. (2021), or a parallelizable but purely competitive loss Trask et al. (2022). Our primary contribution is a splitting scheme outlined in equation I which preserves the benefits of both. Analysis in Theorem 1 provides guaranteed conditions for convergence of this scheme, and experimental results in Figure 3 show that it exposes a source of GPU parallelism

OPT block. A standard gradient descent update is finally applied at each layer of the hierarchy. As explained in Trask et al. (2022), the loss may be derived rigorously from the evidence lower bound and results in a cross-entropy loss penalizing mismatch between the posterior distribution and gating functions. In light of policy gradient methods, one may interpret the posterior as a reward and the gating network as a policy. The updates are thus localized to each level, exposing the source of parallelism that JAX’s `vmap` is able to exploit (See Figure 3).

$$\mathcal{L}_c(\theta; \mathcal{D}) = \sum_{i,d} w_{id} \log \pi_i(\mathbf{x}_d; \theta), \quad (9)$$

$$\mathcal{L}_f(\theta_i; \mathcal{D}) = \sum_{i,j,d} w_{ijd} \log (\pi_i(\mathbf{x}_d; \theta) \pi_{ij}(\mathbf{x}_d; \theta_i)), \quad (10)$$

$$\mathcal{L}_{f2c}(\theta; \mathcal{D}) = \sum_{i,d} \hat{w}_{id} \log \pi_i(\mathbf{x}_d; \theta). \quad (11)$$

A key equation from Trask et al. (2022) demonstrates that the total loss decouples across scales:

$$\mathcal{L}_f(\theta_i; \mathcal{D}) = \mathcal{L}_{f2c}(\theta; \mathcal{D}) + \sum_{i,j,d} w_{ijd} \log (\pi_{ij}(\mathbf{x}_d; \theta_i)). \quad (12)$$

This decoupling allows the scales to separate so that training at the fine level does not impede progress from the coarse. A useful interpretation of this is that the the prolonged loss at the coarse scale may be viewed as a correction to the fine scale.

B PROOF OF THEOREM 1

We first summarize why $\rho(M^{-1}N) < 1$ is a sufficient condition for convergence.

With the scheme given by

$$Mc_{k+1} = Nc_k + b, \quad (13)$$

we may rewrite to obtain

$$c_{k+1} = c_k + M^{-1}(N - M)c_k + M^{-1}b. \quad (14)$$

We now prove that $\rho(M^{-1}N) < 1$. For simplicity, we adopt the notation

$$N = n \mathbf{P}^\top \mathbf{A} \mathbf{P} \quad (15)$$

$$M = m \mathbf{P}^\top \mathbf{B} \mathbf{P} \quad (16)$$

where $m = \sigma_{comp}^{-2}$, $n = -\sigma_{coop}^{-2}$ and $m, n > 0$ denote the variance scaling, $\mathbf{P} \in \mathbb{R}^{N_{width} \times N_{experts} \times N_{data}}$ denotes the output of the i^{th} output neuron of the j^{th} expert model evaluated at the k^{th} node, and $\mathbf{A}$ and $\mathbf{B}$ denote a scaling by either the gating or posterior distribution, respectively.

$$\mathbf{A}_{ij} = \sum_d \pi_{id} \pi_{jd} \quad (17)$$

$$\mathbf{B}_{ij} = \sum_d w_{id} \delta_{ij}. \quad (18)$$

After an appropriate reshaping, it is clear that equation 15 are matrices associated with weighted least squares problems, with $\mathbf{A}$ and $\mathbf{B}$ serving as weights, and are symmetric positive definite provided the weights are non-degenerate.

Rewriting the desired spectral radius inequality in terms of equation 15 and lower bounding the spectral radius by the maximal generalized Rayleigh quotient, we obtain

$$\frac{\sigma_{coop}^2}{\sigma_{comp}^2} > \max_x \frac{x^\top \mathbf{P}^\top \mathbf{A} \mathbf{P} x}{x^\top \mathbf{P}^\top \mathbf{B} \mathbf{P} x}, \quad (19)$$

or after simplifying by defining $y = \mathbf{P}x$

$$\frac{\sigma_{coop}^2}{\sigma_{comp}^2} > \max_y \frac{y^\top \mathbf{A} y}{y^\top \mathbf{B} y}. \quad (20)$$

The numerator and denominator can be treated in worst case by bounding by the maximum and minimum eigenvalues, respectively

$$\frac{\sigma_{coop}^2}{\sigma_{comp}^2} \geq \frac{(\max \lambda_A) y^\top y}{(\min \lambda_B) y^\top y} = \frac{\max \lambda_A}{\min \lambda_B}. \quad (21)$$

As both matrices are positive definite, we can bound using standard element-wise expressions following the Gerschgorin circle theorem Gerschgorin (1931).

$$\max \lambda_A \geq \max_i \sum_j |A_{ij}| \quad (22)$$

$$\min \lambda_B \leq \min_i \left(|B_{ii}| - \sum_{j \neq i} |B_{ij}| \right) \quad (23)$$

$$\frac{\sigma_{coop}^2}{\sigma_{comp}^2} \geq \frac{\max_i \sum_j |A_{ij}|}{\min_i \left(|B_{ii}| - \sum_{j \neq i} |B_{ij}| \right)}. \quad (24)$$

By direct calculation we compute

$$\sum_j |A_{ij}| = \sum_{d,j} \pi_{id} \pi_{jd} \quad (25)$$

$$= \sum_d \pi_{id} \quad (26)$$

and

$$|B_{ii}| - \sum_{j \neq i} |B_{ij}| = \sum_d w_{id} \delta_{ij} - \sum_{d,j \neq i} w_{id} \delta_{ij} \quad (27)$$

$$= \sum_d w_{id} \quad (28)$$

Finally providing our desired bound

$$\frac{\sigma_{coop}^2}{\sigma_{comp}^2} \geq \frac{\max_i \sum_d \pi_{id}}{\min_i \sum_d w_{id}}. \quad (29)$$

Validity of assumptions, anticipated consequences, and practical use. In practice, there are several assumptions that may not be met. The expert models may not be linearly independent (particularly at initialization time); we explore this result experimentally in the Section 4.3. Secondly, in the event of expert posterior collapse (i.e. there exists an i where $w_{id} = 0$ for all d) the estimate could give a division by zero. In exact precision, w_{id} is never zero, as it is the posterior distribution of a Gaussian mixture and Gaussians have non-compact support. In practice, in the extreme tails of the Gaussian contributions the denominator may be vanishingly small, requiring a small stabilizing background white noise in the mixture model to avoid division by zero in machine precision.

C DERIVATIONS FOR PINN APPLICATION

For a general PDE of the form

$$\begin{aligned}\mathcal{L}[u](\mathbf{x}, t) &= f(\mathbf{x}, t), & \mathbf{x} \in \Omega \\ \mathcal{B}[u](\mathbf{x}, t) &= g(\mathbf{x}, t), & \mathbf{x} \in \partial\Omega \\ u(\mathbf{x}, 0) &= u_0(\mathbf{x}), & t = 0\end{aligned}$$

where $\mathcal{L}$ is any derivative operator, we can modify equation 1 with the residue in place of the true prediction and add the terms for boundary/initial conditions to the mean-field approximation. For simplicity, we only show the boundary condition terms, as the initial condition terms are analogous

$$p(\mathcal{E}_{i_1} = u | Z_1 = i_1) = \frac{1}{Q_{i_1}} \mathcal{N}(f; \mathcal{L}[\mu_{i_1}], \sigma_{r_1}^2) \mathcal{N}(f; \mathcal{L}[\hat{y}_1], \sigma_{r_2}^2) \mathcal{N}(\gamma g; \gamma \mu_{i_1}, \sigma_{b_1}^2) \mathcal{N}(\gamma g; \gamma \hat{y}_1, \sigma_{b_2}^2)$$

where γ is an indicator function, i.e.

$$\gamma(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in \partial\Omega \\ 0 & \text{otherwise} \end{cases}$$

The observed data log-likelihood becomes

$$\log L = \sum_{d=1}^{N_d} \log \left[\sum_{i_n} \tilde{\pi}_{i_n}^d \mathcal{N}(f^d; \mathcal{L}[\mu_{i_n}]^d, \sigma_{r_1}^2) \mathcal{N}(f^d; \mathcal{L}[\hat{y}_n]^d, \sigma_{r_2}^2) \mathcal{N}(\gamma^d g^d; \gamma^d \mu_{i_n}^d, \sigma_{b_1}^2) \mathcal{N}(\gamma^d g^d; \gamma^d \hat{y}_n^d, \sigma_{b_2}^2) \right]$$

from which a new ELBO is given in a similar manner to the regression problem. In the E-step, we compute the posterior distribution, i.e.,

$$w_{i_n}(\mathbf{x}^d) = \frac{\tilde{\pi}_{i_n} \mathcal{N}(f^d; \mathcal{L}[\mu_{i_n}](\mathbf{x}^d), \sigma_{r_1}^2) \mathcal{N}(\gamma g^d; \gamma \mu_{i_n}^d, \sigma_{b_1}^2)}{\sum_{j_n} \tilde{\pi}_{j_n} \mathcal{N}(f^d; \mathcal{L}[\mu_{j_n}](\mathbf{x}^d), \sigma_{r_1}^2) \mathcal{N}(\gamma g^d; \gamma \mu_{j_n}^d, \sigma_{b_1}^2)}.$$

If $\mathcal{L}$ is a linear operator, taking the derivative of the ELBO with respect to the expert coefficients yields a similar weighted least-squares problem to equation LS, where the matrix on the left-hand side

$$\sum_d \left[\frac{w_{i_n}^d \delta_{i_n, j_n}}{\sigma_{r_1}^2} \mathcal{L}[H]_{i_n, \alpha}^d \mathcal{L}[H]_{j_n, \beta}^d + \frac{\mathcal{L}[\tilde{\pi}_{i_n} H_{i_n, \alpha}]^d \mathcal{L}[\tilde{\pi}_{j_n} H_{j_n, \beta}]^d}{\sigma_{r_2}^2} + \gamma^d \left(\frac{w_{i_n}^d \delta_{i_n, j_n}}{\sigma_{b_1}^2} + \frac{\tilde{\pi}_{i_n}^d \tilde{\pi}_{j_n}^d}{\sigma_{b_2}^2} \right) H_{i_n, \alpha}^d H_{j_n, \beta}^d \right]$$

and the right-hand side vector

$$\sum_d \left[\left(\frac{w_{i_n}^d \mathcal{L}[H]_{i_n, \alpha}^d}{\sigma_{r_1}^2} + \frac{\mathcal{L}[\tilde{\pi}_{i_n} H_{i_n, \alpha}]^d}{\sigma_{r_2}^2} \right) f^d + \gamma^d \left(\frac{w_{i_n}^d}{\sigma_{b_1}^2} + \frac{\tilde{\pi}_{i_n}^d}{\sigma_{b_2}^2} \right) g^d H_{i_n, \alpha}^d \right]$$

If $\mathcal{L}$ is nonlinear, we will need to set up a Newton solver. $\mathbf{H}_{i_n}$ and $\tilde{\pi}_{i_n}$ are optimized with a gradient descent loss

$$\mathcal{L}_{\text{GD}} = -\ell(\theta)$$

similar to the regression problem.

D EXPERIMENTAL DETAILS AND HYPERPARAMETERS

Simulations were conducted on an Nvidia A100 cluster, an Nvidia H200 cluster, as well initial experiments prototyped on a Macbook Air M2 16GB/256GB. Scripts for experiments will be provided on the anonymized github with associated seeds.

D.1 HYPERPARAMETERS USED FOR DATA COLLECTION

Remark. For the PINN case the additional terms present in the multiobjective loss require special care. To accelerate hyperparameter tuning for the ablation study of Peclet number the splitting scheme was turned off and the SOR scheme turned on with small ω for a few cases to ensure convergence.

Table 1: Key hyperparameters used in section 4.1.

Component	Parameter	Value
Problem	Dataset	1D Gaussian peak, $\mu = 0.5$, $\sigma = 0.05$
	Input dimension	1
Gating Network	Hidden units / Depth (excluding input layer)	30 / 0
	Activation	$\tanh$
Hierarchical MoE	Partitions (N_c , N_f)	3, 2
	Basis size / Hidden units / Depth	30 / 30 / 1
Training	Outer iterations	50,000 (staged training)
	Learning rates (coarse/fine)	1e-3 / 1e-3
	$\sigma_{\text{comp}} / \sigma_{\text{coop}}$	1.0 / 1e6
Iterative Solver	Type	Iterative splitting scheme
	Tolerance / Regularization	1e-12 / 1e-4
	Max iterations	10,000
Precision	Data and parameters	float64

Table 2: Key hyperparameters used for strong scaling study (section 4.2).

Component	Parameter	Value
Problem	Dataset	Gaussian peak, $\mu = 0.5$, $\sigma = 0.1$
	Input dimension	64
Gating Network	Hidden units / Depth (excluding input layer)	30 / 0
	Activation	$\tanh$
Hierarchical MoE	Partitions sweep (N_c , N_f)	(4, 2), (4, 4), (4, 8), (4, 16), (4, 32)
	Basis size / Hidden units / Depth	30 / 20 / 1
Training	Benchmark iterations	5000 (solver timing only)
	Learning rates (coarse/fine)	1e-3 / 1e-3
	$\sigma_{\text{comp}} / \sigma_{\text{coop}}$	1.0 / 1e5
Iterative Solver	Type	Direct / Iterative splitting scheme (with and without vmap)
	Tolerance / Regularization	1e-12 / 1e-4
	Max iterations	5000
Precision	Data and parameters	float32

Table 3: Key hyperparameters used in section 4.3.

Component	Parameter	Value
Problem	Dataset	1D Gaussian peak, $\mu = 0.5$, $\sigma = 0.05$
	Input dimension	1
Gating Network	Hidden units / Depth (excluding input layer)	50 / 0
	Activation	$\tanh$
Hierarchical MoE	Partitions (N_c , N_f)	4, 2
	Basis size / Hidden units / Depth	20 / 20 / 1
Training	Outer iterations	5,000
	Learning rates (coarse/fine)	1e-3 / 1e-3
	$\sigma_{\text{comp}} / \sigma_{\text{coop}}$	1.0 / 1e5
Iterative Solver	Type	Iterative splitting scheme
	Tolerance / Regularization	1e-12 / 1e-4
	Max iterations	2,000
Precision	Recorded iterations counts	performed for σ_{coop} given in legend
	Data and parameters	float64

Table 4: Key hyperparameters used for linear solve iteration scaling with problem dimension (section 4.4).

Component	Parameter	Value
Problem	Dataset	1D Gaussian peak, $\mu = 0.5$, $\sigma = 0.1$
	Input dimension	1, 2, 4, 8, 16, 32, 64, 128
Gating Network	Hidden units / Depth (excluding input layer)	100 / 0
	Activation	tanh
Hierarchical MoE	Coarse / Fine partitions (N_c , N_f)	4 / 2
	Basis size / Hidden units / Depth	10 / 10 / 1
Training	Outer iterations	100,000
	Accuracy threshold	1% relative error
	Learning rates (coarse/fine)	1e-3 / 1e-3
Iterative Solver	$\sigma_{\text{comp}} / \sigma_{\text{coop}}$	1.0 / 1e5
	Type	Iterative splitting scheme
	Tolerance / Regularization	1e-12 / 1e-4
	Max iterations	100,000
Precision	Data and parameters	float64

Table 5: Key hyperparameters used for convection-diffusion problem (section 4.5).

Component	Parameter	Value
Problem	Dataset	100x100 grid points
Gating Network	Hidden units / Depth (excluding input layer)	40/0 or 20/0
	Activation	tanh
Hierarchical MoE	Coarse / Fine partitions (N_c , N_f)	4 / 2
	Basis size / Hidden units / Depth	MoEs: 10 / 10 / 2
		PINNs: 40/40/3
Training	Outer iterations	MoEs: 50,000
	Learning rate	PINNs: 100,000
	$\sigma_{\text{comp}} / \sigma_{\text{coop}}$	1e-4
Precision		1e-3 / 1e-3
	Data and parameters	float64

Table 6: Key hyperparameters used for Peclet number ablation study (section 4.5).

Component	Parameter	Value
Problem	Dataset	500 points on [0,1]
Gating Network	Hidden units / Depth (excluding input layer)	40/1
	Activation	tanh
Hierarchical MoE	N_c (1 level)	4
	Basis size / Hidden units / Depth	MoEs: 10 / 10 / 0
		PINNs: 40/30/4
Training	Outer iterations	MoEs: 30,000
	Learning rate	PINNs: 100,000
	$\sigma_{\text{comp}} / \sigma_{\text{coop}}$	1e-4
Precision		1e5 / 1e-3
	Data and parameters	float64