HYCO: A FORMALISM FOR HYBRID-COOPERATIVE PDE MODELLING

LORENZO LIVERANI[†] AND ENRIQUE ZUAZUA[†*§]

Abstract. We present Hybrid-Cooperative Learning (HYCO), a hybrid modeling framework that integrates physics-based and data-driven models through mutual regularization. Unlike traditional approaches that impose physical constraints directly on synthetic models, HYCO treats both components as co-trained agents nudged toward agreement. This cooperative scheme is naturally parallelizable and demonstrates robustness to sparse and noisy data. Numerical experiments on static and time-dependent benchmark problems show that HYCO can recover accurate solutions and model parameters under ill-posed conditions. The framework admits a game-theoretic interpretation as a Nash equilibrium problem, enabling alternating optimization. This paper is based on the extended preprint [6].

1. Introduction

Mathematical modeling has long relied on two dominant paradigms. Physics-based models are grounded in physical laws and typically involve a small number of interpretable parameters, offering theoretical guarantees and good extrapolation capacity. Data-driven models, in contrast, employ flexible function approximators such as neural networks, excelling in empirical performance but requiring large datasets and often lacking interpretability. Each approach has its limitations: physics-based models struggle when governing equations are unknown or difficult to specify, while data-driven models perform poorly in low-data regimes and fail to extrapolate beyond training distributions.

Recent hybrid approaches attempt to bridge this gap. Physics-Informed Neural Networks (PINNs) [10] impose physical constraints as soft penalties in the loss function, while methods like SINDy [2] discover governing equations from data. Neural operators [7, 5] learn mappings between function spaces. These approaches, however, remain fundamentally constrained: they attempt to satisfy both physical laws and data fidelity within a single model, leading to competing optimization objectives that can be difficult to balance effectively.

We introduce Hybrid-Cooperative Learning (HYCO), a fundamentally different approach that trains two models in parallel: a physical model grounded in differential equations (with unknown parameters Λ) and a synthetic model guided by data (with parameters Θ). Rather than encoding physics as constraints within the synthetic model, HYCO couples the models through an interaction loss that measures the discrepancy between their predictions. The key mechanism can be understood through three loss components. First, the physical model has its own loss $\mathsf{L}_{phy}(\Lambda)$, measuring how well it fits observational data (if any is provided). Second, the synthetic model has a loss $\mathsf{L}_{syn}(\Theta)$ quantifying its fit to the data. Third, and most importantly, the interaction

[†]Chair for Dynamics, Control, Machine Learning, and Numerics (Alexander von Humboldt Professorship), Department of Mathematics, Friedrich-Alexander-Universität Erlangen-Nürnberg, 91058 Erlangen, Germany.

[*]Departamento de Matemáticas, Universidad Autónoma de Madrid, 28049 Madrid, Spain.

[§]Chair of Computational Mathematics, Universidad de Deusto. Av. de las Universidades, 24, 48007 Bilbao, Basque Country, Spain.

E-mail addresses: lorenzo.liverani@fau.de, enrique.zuazua@fau.de.

1991 *Mathematics Subject Classification.* 00A71, 35R30, 41A27, 68T07, 91A12.

Key words and phrases. Hybrid modeling, Machine Learning, Physics-Based, Nash equilibrium.

loss $\mathsf{L}_{int}(\Theta, \Lambda)$ measures the distance between the two models' predictions across the domain. The overall objective combines these terms as

$$\min_{\Theta,\Lambda} \alpha \mathsf{L}_{syn}(\Theta) + \beta \mathsf{L}_{phy}(\Lambda) + \mathsf{L}_{int}(\Theta, \Lambda),$$

where $\alpha, \beta \geq 0$ control the relative importance of each component. Crucially, either weight may be zero, allowing scenarios where one model trains exclusively through the interaction term without direct access to observational data.

As a simple diagnostic baseline, one may set the interaction term to zero, which decouples the two components and reduces HYCO to two independently trained models.

This cooperative formulation admits a natural game-theoretic interpretation. Instead of jointly optimizing both models, we can view them as two autonomous agents or players. The physical model seeks to minimize its own cost

$$\mathsf{L}_1 := \beta \mathsf{L}_{phy}(\Lambda) + \mathsf{L}_{int}(\Theta, \Lambda), \tag{1.1}$$

while the synthetic model aims to minimize

$$\mathsf{L}_2 := \alpha \mathsf{L}_{syn}(\Theta) + \mathsf{L}_{int}(\Theta, \Lambda). \tag{1.2}$$

Each player strives to reduce its own loss while being constrained by the other's choices through the shared interaction term. The solution concept is a Nash equilibrium: a configuration where neither agent can unilaterally reduce its loss by changing parameters. This perspective naturally suggests an alternating minimization scheme: at each iteration, we update the physical parameters Λ while holding Θ fixed, then update Θ while holding Λ fixed. This back-and-forth process nudges both models toward mutual agreement, enabling them to cooperate despite having potentially different objectives.

Contributions. This short paper has a primarily expository and methodological scope. Our main contributions are:

- ◇ *Symmetric hybridization.* We formulate HYCO as a genuinely *peer-to-peer* coupling of a physics-based model and a data-driven model, avoiding the usual paradigm where physics is only enforced as a constraint on the synthetic component.
- ◇ *Game-theoretic viewpoint and practical algorithm.* We reinterpret the coupled training as a two-player game with alternating updates, leading to a simple and implementable training procedure.
- ◇ *Stochastic interaction via ghost points.* We introduce a lightweight interaction-evaluation strategy based on randomly sampled "ghost points", which decouples the interaction loss from the physical mesh and is convenient in sparse/irregular-data settings.

1.1. **Advantages of HYCO.**

The scope of HYCO is broad. At its simplest, it can be used to embed standard machine learning models with physical biases, much like in the spirit of PINNs. However, the symmetric treatment of both models in HYCO provides notable flexibility: neither component is subordinate to the other, and the framework imposes minimal restrictions on model choice.

Besides, *parallelization* emerges naturally from HYCO's design. During each training iteration, both models can compute their predictions and gradients simultaneously, with synchronization required only when evaluating the interaction loss. This parallel structure enables distributed training across multiple devices, and in principle could offer computational advantages, though the need to train two models simultaneously means efficiency gains depend significantly on implementation details and the specific problem structure.

Privacy awareness is another practical advantage. The two components need not exchange raw data or model parameters; in our implementations they can interact through predicted states (and/or residual-type summaries). This is particularly attractive in federated or multi-institution settings where data cannot leave its source due to regulatory or proprietary constraints. A full

privacy analysis (e.g., potential information leakage through shared predictions) is beyond the scope of this short note.

Finally, HYCO is designed to be robust to data sparsity. The interaction loss provides implicit regularization: even when observational data is limited, the physical model benefits from the synthetic model's learned patterns, while the synthetic model gains stability from physical consistency. This mutual regularization can be particularly valuable for extrapolation beyond the training domain, a scenario where purely data-driven approaches often struggle.

2. Methodology

2.1. **Problem Formulation.** We consider a general setting where observational data is available from some underlying ground truth model or physical system. The goal is to simultaneously recover the solution field and identify unknown parameters characterizing the system. To this end, we employ two models operating in parallel.

The *physical model* is based on a structural ansatz with unknown parameters Λ. This could be a discretized differential equation solver, a system of ordinary differential equations, or any model grounded in domain knowledge where certain coefficients, forcing terms, or boundary conditions are unknown. Given parameters Λ, the physical model produces a prediction $\boldsymbol{u}_{phy}$. If observational data is available to the physical model, it seeks to minimize

$$\mathsf{L}_{phy}(\Lambda) := \frac{1}{M}\sum_{i=1}^{M} \ell(\boldsymbol{u}_{phy}(x_i), \boldsymbol{u}^D(x_i)) + \mathscr{P}(\Lambda), \tag{2.1}$$

where $\boldsymbol{u}^D$ represents observational data at locations $\{x_i\}_{i=1}^M$, ℓ is a loss function (typically mean squared error), and $\mathscr{P}(\Lambda)$ is an optional regularization term on the parameters.

The *synthetic model* is a data-driven component, typically a neural network with parameters Θ. It learns to approximate the underlying system directly from data by minimizing

$$\mathsf{L}_{syn}(\Theta) := \frac{1}{M}\sum_{i=1}^{M} \ell(\boldsymbol{u}_{syn}(x_i), \boldsymbol{u}^D(x_i)) + \mathscr{P}(\Theta). \tag{2.2}$$

The key innovation lies in the *interaction loss*, which couples the two models by measuring the discrepancy between their predictions over the entire domain Ω:

$$\mathsf{L}_{int}(\Theta, \Lambda) := \int_\Omega \|\boldsymbol{u}_{syn}(x) - \boldsymbol{u}_{phy}(x)\|^2 dx. \tag{2.3}$$

This term enforces consistency between the models across the full spatial domain, not merely at the observed data points. The overall optimization problem becomes

$$\min_{\Theta,\Lambda} \alpha \mathsf{L}_{syn}(\Theta) + \beta \mathsf{L}_{phy}(\Lambda) + \mathsf{L}_{int}(\Theta, \Lambda), \tag{2.4}$$

where $\alpha, \beta \geq 0$ are weights that balance the objectives. Importantly, either weight may be zero, allowing one model to train exclusively through the interaction term without direct access to observational data.

The framework is agnostic to whether the problem is static or dynamic; the essential structure remains unchanged. In particular, this formulation extends naturally to time-dependent problems, where the domain becomes $\Omega \times [0, T]$ and all integrals are taken over both space and time.

2.2. **The HYCO Algorithm.** Rather than jointly minimizing (2.4), we adopt an alternating optimization scheme motivated by the game-theoretic perspective. Recalling $\mathsf{L}_1, \mathsf{L}_2$ defined at (1.1)-(1.2), at each iteration we update one set of parameters while holding the other fixed, taking a gradient descent step to minimize either the physical or the synthetic loss:

Algorithm 1: HYCO Alternating Optimization

Input: Initial parameters $\Lambda^{(0)}, \Theta^{(0)}$; weights α, β; learning rates $\eta_\Lambda, \eta_\Theta$; maximum iterations K

Output: $(\Lambda^{(K)}, \Theta^{(K)})$

for $k = 0, 1, 2, \ldots, K-1$ **do**

  Physical update: $\Lambda^{(k+1)} \leftarrow \Lambda^{(k)} - \eta_\Lambda \nabla_\Lambda \mathsf{L}_1$;

  Synthetic update: $\Theta^{(k+1)} \leftarrow \Theta^{(k)} - \eta_\Theta \nabla_\Theta \mathsf{L}_2$;

  if *stopping criterion satisfied* **then**

    break;

  end

end

2.3. **Ghost Points and Efficient Computation.** A critical implementation detail concerns the computation of the interaction loss L_{int}. Evaluating the integral in (2.3) exactly at every training iteration would require discretizing the entire domain Ω on a fine grid, which can be prohibitively expensive, especially for high-dimensional problems or when the physical model itself already operates on a dense mesh.

To address this, we introduce the concept of *ghost points*. The term 'ghost" reflects that these points are not part of the dataset and do not directly contribute to the data-fitting losses L_{phy} or L_{syn}. Rather, they are auxiliary evaluation points sampled randomly from Ω at each training iteration, serving solely to approximate the interaction loss. At iteration k, we sample H points $\{x_h\}_{h=1}^H$ uniformly (or according to some other sampling strategy) from Ω and approximate

$$\mathsf{L}_{int}(\Theta, \Lambda) \approx \frac{|\Omega|}{H} \sum_{h=1}^{H} \|\boldsymbol{u}_{syn}(x_h) - \boldsymbol{u}_{phy}(x_h)\|^2.$$

This stochastic approximation offers several advantages. First, it decouples the computational cost of evaluating L_{int} from the discretization resolution of the physical model. Second, the random sampling introduces a form of stochastic regularization that can help prevent overfitting to specific regions of the domain. Third, different ghost points are sampled at each iteration, ensuring that over the course of training, the interaction loss is enforced approximately uniformly across Ω. This approach draws inspiration from Random Batch Methods and stochastic optimization techniques commonly used in large-scale machine learning.

The number of ghost points H represents a trade-off: too few points may yield noisy gradient estimates, while too many increase computational cost without commensurate benefit. In our experiments, we found that moderate values of H (on the order of hundreds) suffice to obtain stable training.

2.4. **Stopping Criterion.** A natural question is when to terminate the alternating optimization. One possible approach monitors the change in the interaction loss between successive iterations. If the models have reached approximate agreement, the interaction loss should stabilize. We can thus define a stopping criterion as

$$\frac{|\mathsf{L}_{int}^{(k+1)} - \mathsf{L}_{int}^{(k)}|}{\mathsf{L}_{int}^{(k)}} < \epsilon, \tag{2.5}$$

where $\epsilon > 0$ is a tolerance parameter. When this condition is satisfied, further alternating updates are unlikely to significantly improve the solution, and training can be halted.

We emphasize that this is only one possible implementation choice. Other stopping criteria could be based on the change in parameter values, validation loss, or a fixed computational budget. The appropriate choice depends on the specific application and available computational resources.

3. Related Work

Hybrid modeling approaches have emerged as a promising direction to combine physics and data. Physics-Informed Neural Networks (PINNs) [10] encode PDE residuals directly in the loss function, enabling the network to learn solutions that approximately satisfy governing equations. Variants include variational formulations [4] and Bayesian extensions [13]. Neural operators [7, 5] learn mappings between function spaces, offering a different paradigm for data-driven PDE solving. SINDy [2] discovers governing equations from data through sparse regression.

These methods share a common limitation: they attempt to satisfy both physical constraints and data fidelity within a single model, leading to challenging multi-objective optimization. In contrast, HYCO decouples these objectives by employing two independent models that coordinate through consistency constraints. This design is conceptually closer to ensemble learning [14], though aggregation occurs during training rather than at inference. The alternating optimization bears similarity to ADMM-based PINN methods [11], but HYCO does not impose a hierarchical relationship between models.

HYCO's philosophy also connects to federated learning [8], where distributed agents train collaboratively while preserving privacy. In HYCO, the physical and synthetic models exchange only predictions, not parameters or raw data, making the framework naturally privacy-aware. Recent work on minimal variance model aggregation [1] and bilevel optimization for neural operators [15] explores related ideas, though from different perspectives.

A distinguishing feature of HYCO is the symmetric treatment of the two models. Neither is subordinate to the other; instead, they cooperate as peers to reach consensus. This symmetry provides flexibility in model selection and allows for scenarios in which one model is trained solely through interaction (e.g., by setting $\beta = 0$ or $\alpha = 0$ in the minimization problem), a configuration that is not naturally accommodated by most hybrid frameworks. For comparison, we mention the works [9, 12], where hybrid models for inverse problems are developed, but with a clear hierarchy between the physical and the synthetic model.

4. Numerical Experiments

We validate HYCO on two benchmark problems that are well-established in the inverse problems literature: the Gray-Scott reaction-diffusion system [3] and the heterogeneous Helmholtz equation. The former tests parameter identification in a dynamic setting with complex pattern formation, whereas the latter tests spatial parameter reconstruction with limited observational data.

4.1. Gray-Scott Reaction-Diffusion System.

The Gray-Scott model [3] describes autocatalytic chemical reactions between two species $u = u(t, x, y)$ and $v = v(t, x, y)$ on the plane, and it is well known to exhibit rich spatio-temporal dynamics and pattern formation. On the unit square $\Omega = [0, 1]^2 \subset \mathbb{R}^2$ with periodic boundary conditions, the system of PDEs reads

$$\begin{cases} u_t - D_u \Delta u - uv^2 + F(1 - u) = 0, \\ v_t - D_v \Delta v + uv^2 - (F + k)v = 0, \\ u(0, x, y) = u_0(x, y), \quad v(0, x, y) = v_0(x, y). \end{cases} \tag{4.1}$$

Here, D_u and D_v are the diffusion coefficients for u and v, respectively; F is the feed rate of u; and k denotes the decay rate of v. The ground-truth parameters used to generate the data in this experiment are:

$$D_u = 2 \times 10^{-6}, \quad D_v = 0.8 \times 10^{-6}, \quad F = 0.018, \quad k = 0.051.$$

The initial values are

$$u_0(x, y) = \chi_{B_{0.1}(1/2)}, \quad v_0(x, y) = \chi_{\Omega \setminus B_{0.1}(1/2)},$$

where χ_A denotes the characteristic function of the set A and $B_{0.1}(1/2)$ is the ball of radius 0.1 centered at $(x, y) = (1/2, 1/2)$. In what follows, we assume the initial datum is known. This information will be supplied to all compared models.

The dataset is constructed by numerically solving the Gray-Scott equations on a 64×64 grid, using a finite difference scheme with small temporal steps (5000 time steps from 0 to $T = 2000$) to generate a reference true solution $\boldsymbol{u}^D$. From this solution, we sample the values of both species at $M = 5000$ random points, obtaining the dataset

$$\mathcal{D} := \{\boldsymbol{u}^D(x_i, y_i, t_i)\}_{i=1}^M = \{u^D(x_i, y_i, t_i), v^D(x_i, y_i, t_i)\}_{i=1}^M,$$

where the superscript D denotes "Data". Note, in particular, that our data is not equispaced in space nor in time.

The goal of this experiment is to recover the time-evolution of the solution over the entire domain Ω, generalizing to points not included in the training set, and to recover the diffusivity parameters D_u, D_v, henceforth assumed unknown.

Model configurations. We compare three methods: HYCO, a baseline deep neural network, and a physics-informed neural network (PINN). The HYCO framework consists of two cooperatively trained components:

- ◇ *Physical model* ($\boldsymbol{u}_{phy}$): This consists of a finite difference discretization of (4.1) on the same 64×64 spatial grid. The unknown diffusion coefficients $\Lambda = \{D_u, D_v\}$ are initialized at $D_u = 1 \times 10^{-6}$ and $D_v = 0.5 \times 10^{-6}$, deliberately chosen away from the true values to test parameter recovery.
- ◇ *Synthetic model* ($\boldsymbol{u}_{syn}$): This is a feedforward neural network with four hidden layers (128 neurons each, ReLU activation), mapping $(x, y, t) \in \mathbb{R}^3$ to $(u_{syn}, v_{syn}) \in \mathbb{R}^2$. This network learns to approximate the solution directly from data while being regularized by the physical model through the interaction loss.

The HYCO interaction loss is computed using $H = 1000$ ghost points at each iteration. The baseline neural network uses the same architecture as HYCO's synthetic component but is trained purely on data without physical constraints. The PINN also uses the same architecture but with tanh activation (required for computing PDE residuals via automatic differentiation). It is trained with 50,000 randomly sampled collocation points to enforce the PDE residuals in the loss function, and also attempts to identify D_u and D_v by treating them as additional learnable parameters. HYCO and the baseline network were trained for 600 epochs. The PINN required 3500 epochs to achieve convergence.

Results. Figure 4.1 shows the evolution of the u-component at five time snapshots. The physical and synthetic components of HYCO both capture the pattern formation dynamics throughout the time horizon. The baseline neural network and PINN initially approximate the solution but become unstable as pattern complexity increases, failing to extrapolate beyond the temporal range of the training data.

Table 4.1 reports the final normalized L^2 errors

$$\mathsf{e}_s^m := \frac{\|\boldsymbol{u} - \boldsymbol{u}_m\|_{L^2(\Omega\times[0,T])}}{\|\boldsymbol{u}\|_{L^2(\Omega\times[0,T])}},$$

computed on the full discretization grid. Here $\boldsymbol{u}$ is the ground truth solution and $\boldsymbol{u}_m$ the prediction of the model m (HYCO physical or synthetic, PINN or NN). The HYCO physical model achieves error 0.020, followed by HYCO synthetic (0.063). The PINN and baseline network exhibit higher errors (0.123 and 0.153 respectively) due to temporal instability.

HYCO Physical	**HYCO Synthetic**	**PINN**	**NN**
0.020	0.063	0.123	0.153

TABLE 4.1. Normalized L^2 errors for Gray-Scott approximations.

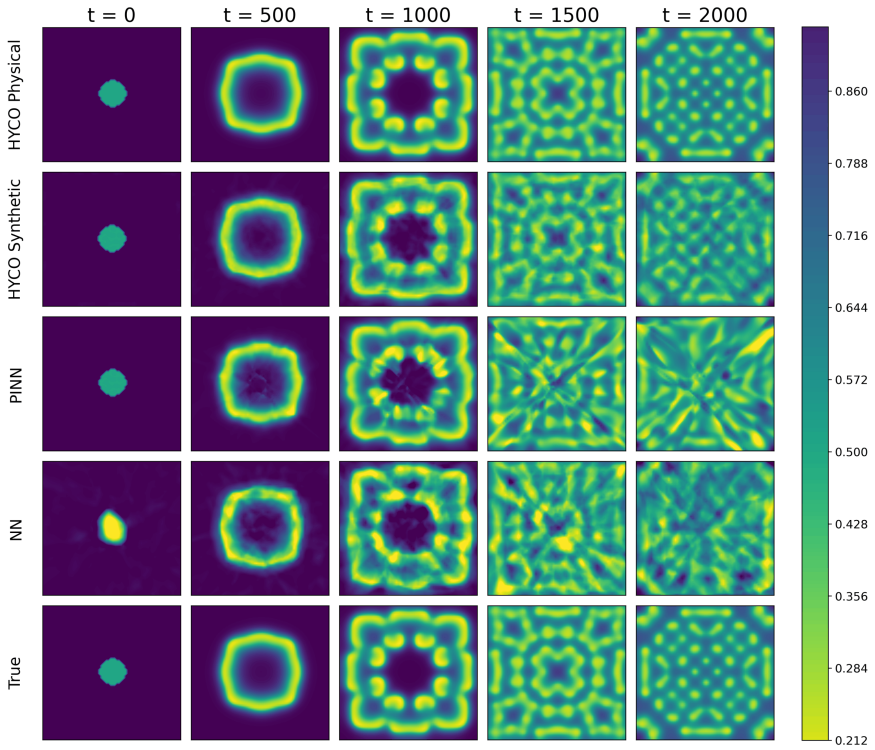

FIGURE 4.1. Gray-Scott experiment: evolution of the u-component. Rows show HYCO Physical, HYCO Synthetic, pure NN, PINN, and ground truth.

For parameter identification, Table 4.2 compares the recovered diffusion coefficients. HYCO accurately identifies the ground-truth values, while the PINN converges to incorrect parameters despite fitting the training data reasonably well. This occurs because the PINN's flexible network can compensate for incorrect parameters by learning a solution that approximately satisfies the PDE with biased coefficients. The HYCO framework mitigates this issue through the consistency constraint between the physical and synthetic models.

	D_u	D_v
Ground truth	2×10^{-6}	0.8×10^{-6}
HYCO Physical	$\mathbf{1.989 \times 10^{-6}}$	$\mathbf{0.799 \times 10^{-6}}$
PINN	0.0999×10^{-6}	0.0381×10^{-6}

TABLE 4.2. Identified diffusivity parameters for Gray-Scott.

4.2. **Helmholtz Equation: Static Inverse Problem.** The heterogeneous Helmholtz equation is a classical benchmark for inverse problems in the machine learning and PDE literature, widely used for testing both forward and inverse solvers. Our setup closely follows Example 3.3.2 from the work by Yang, Meng, and Karniadakis on Bayesian PINNs [13].

The governing equation is the 2D heterogeneous Helmholtz equation with null Dirichlet boundary conditions, namely

$$\begin{cases} -\nabla \cdot (\kappa(x,y)\nabla \boldsymbol{u}) + \eta(x,y)^2 \boldsymbol{u} = f(x,y), & \text{for } (x,y) \in \Omega, \\ \boldsymbol{u}(x,y) = 0, & \text{for } (x,y) \in \partial\Omega, \end{cases} \tag{4.2}$$

where $\Omega = [-\pi, \pi]^2$. Here $\boldsymbol{u} = \boldsymbol{u}(x,y)$ is the unknown solution field, $\kappa(x,y)$ represents the diffusion coefficient, $\eta(x,y)$ is the wave number, and $f(x,y)$ is a known forcing term. The ground-truth

coefficients are defined as

$$\kappa(x,y) = \varphi(x,y;\alpha_1,c_1) + 1, \quad \eta(x,y) = \varphi(x,y;\alpha_2,c_2) + 1,$$

with parameters

$$\alpha_1 = 4, \quad \alpha_2 = 1, \quad c_1 = (-1,-1), \quad c_2 = (2,1),$$

and $\varphi(\cdot,\cdot;\alpha,c)$ being the Gaussian function with amplitude α and center $c = (c_x, c_y)$:

$$\varphi(x,y;\alpha,c) = \alpha e^{-(x-c_x)^2-(y-c_y)^2}.$$

Following [13], the reference solution is set to

$$\boldsymbol{u}(x,y) = \sin(x)\sin(y),$$

with the corresponding forcing term $f(x,y)$ computed by substituting this solution into (4.2).

The dataset is constructed by randomly placing $M = 25$ sensors within specified regions of the domain and evaluating the reference solution at these locations:

$$\mathcal{D} := \{\boldsymbol{u}^D(x_i,y_i)\}_{i=1}^M.$$

The forcing term f is assumed known and supplied to all models.

The goal of this experiment is to solve the inverse problem of recovering the parameters $\Lambda := \{\alpha_1, c_1, \alpha_2, c_2\} \in \mathbb{R}^6$ that define the coefficients κ and η. Since c_1 and c_2 are 2D points, the total number of parameters to identify is six.

Data generation. We test three scenarios that progressively reduce the observational coverage:

- ◇ *Full domain (Ω)*: sensors randomly placed across the entire domain $[-\pi,\pi]^2$.
- ◇ *Partial domain (Q_1)*: sensors restricted to $[-\pi/2,\pi]^2$ (approximately three-quarters of the domain).
- ◇ *Quarter domain (Q_2)*: sensors restricted to $[0,\pi]^2$ (one-quarter of the domain, most challenging).

In each case, $M = 25$ sensors are randomly positioned within the specified region.

Model configurations. We compare three methods: HYCO, a classical finite element inverse solver (FEM), and a physics-informed neural network (PINN). The HYCO framework consists of two cooperatively trained components:

- ◇ *Physical model* ($\boldsymbol{u}_{phy}$): This consists of a finite element discretization of (4.2) with unknown parameters $\Lambda = \{\alpha_1, c_1, \alpha_2, c_2\}$ initialized randomly. In this experiment the physical model is also trained on the dataset $\mathcal{D}$, minimizing the data mismatch loss along with the interaction term.
- ◇ *Synthetic model* ($\boldsymbol{u}_{syn}$): This is a feedforward neural network with ResNet-like architecture featuring two hidden layers (256 neurons each, ReLU activation) and skip connections, mapping $(x,y) \in \mathbb{R}^2$ to $\boldsymbol{u}_{syn}(x,y) \in \mathbb{R}$.

The HYCO interaction loss is computed using $H = 200$ ghost points at each iteration.

The finite element method (FEM) baseline solves the classical inverse problem by optimizing the six parameters via gradient-based methods on the discretized PDE, minimizing only the data mismatch. In the partial-coverage settings, this inverse problem becomes markedly ill-posed: many parameter configurations can explain the measurements inside the observed region while producing very different fields elsewhere, so the optimization is sensitive to initialization and regularization. The PINN uses a similar neural architecture (with tanh activation required for automatic differentiation), enforcing the PDE residual through 400 randomly sampled collocation points (boundary and interior), and treating the six parameters as learnable alongside the network weights. All models were trained for a maximum of 2000 epochs using the Adam optimizer.

Results. Table 4.3 summarizes the parameter error e_p^m and the solution error e_s^m across the three test cases, where we report $\mathsf{e}_p^m := \|\Lambda^{(m)} - \Lambda^\dagger\|_2 / \|\Lambda^\dagger\|_2$ and $\mathsf{e}_s^m := \|u^{(m)} - u^\dagger\|_{L^2(\Omega)} / \|u^\dagger\|_{L^2(\Omega)}$ (approximated on the reference grid). When data covers the full domain (Ω), all methods achieve accurate

Domain	Model	e_p^m	e_s^m
Ω	FEM	0.024	0.012
	HYCO Physical	**0.016**	**0.012**
	PINN	0.158	0.061
Q_1	FEM	**0.024**	**0.012**
	HYCO Physical	0.031	**0.012**
	PINN	0.178	0.078
Q_2	FEM	1.109	0.366
	HYCO Physical	**0.041**	**0.072**
	PINN	0.750	0.361

TABLE 4.3. Helmholtz results: parameter error e_p and solution error e_s across observational domains.

parameter recovery with small errors. As observational coverage decreases, however, performance differences become pronounced. For the intermediate case (Q_1), HYCO and FEM perform comparably, while the PINN exhibits larger errors. For the most challenging case (Q_2), HYCO achieves parameter error $\mathsf{e}_p^m = 0.041$ and solution error $\mathsf{e}_s^m = 0.072$, substantially outperforming both FEM ($\mathsf{e}_p^m = 1.109$) and PINN ($\mathsf{e}_p = 0.750$).

Figure 4.2 visualizes the recovered coefficients for the most challenging case (Q_2). HYCO accurately reconstructs both κ and η across the entire domain, including regions without observations (marked by red dots). In contrast, FEM and PINN converge to alternative parameter configurations that fit the observed data well but fail to generalize outside the observation region. This demonstrates that all three models find mathematically valid solutions to the inverse problem in the sense that they reproduce the observed data. However, HYCO's cooperative framework, which implicitly encourages consistency between the physical and synthetic models beyond the data region, leads to parameter estimates closer to the ground truth.

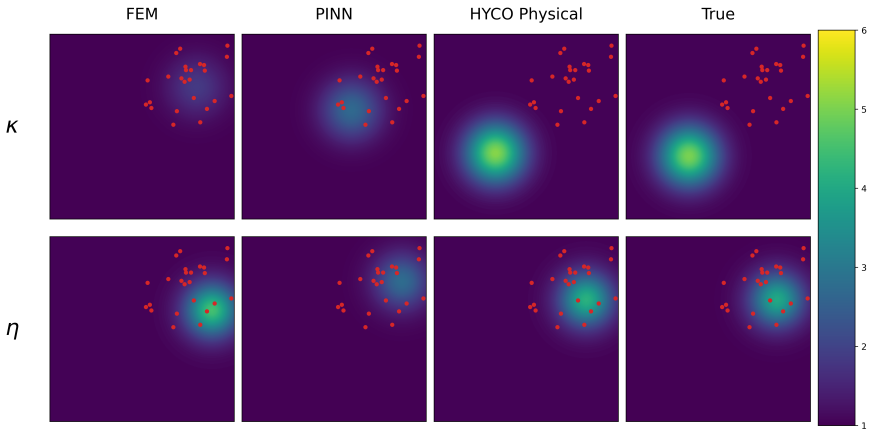

FIGURE 4.2. Recovered coefficients for Helmholtz equation with observations in Q_2: κ (top row), η (middle row). Columns show FEM, PINN, HYCO Physical, and ground truth. Red dots indicate sensor positions.

Figure 4.3 shows the evolution of errors during training for the Q_2 case. All models effectively reduce the data mismatch (top panel), which is expected since this term is explicitly minimized. However, only HYCO maintains low solution and parameter errors throughout training (middle and bottom panels), while FEM and PINN converge to solutions with accurate data fit but incorrect parameters.

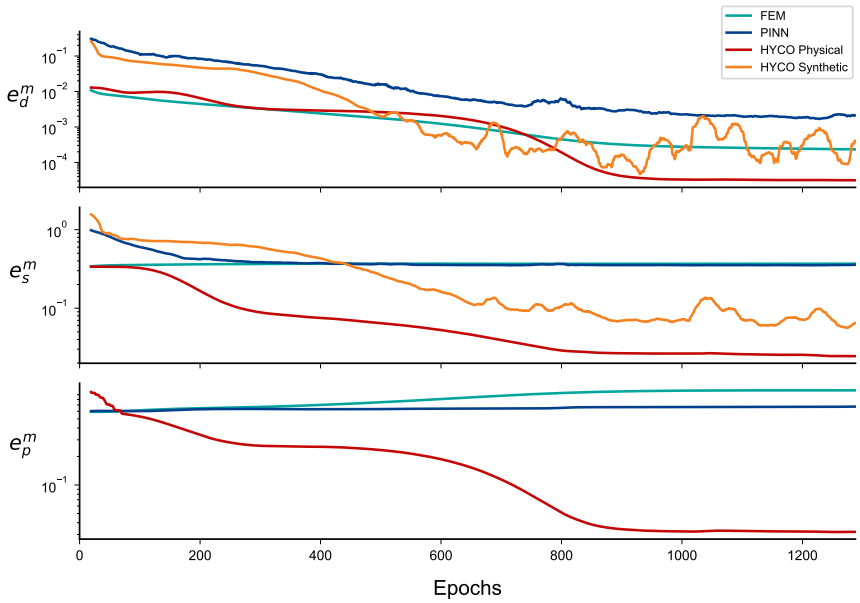

FIGURE 4.3. Error evolution for Helmholtz experiment with observations in Q_2. Top: data mismatch. Middle: solution error. Bottom: parameter error.

5. Conclusions and Future Work

This paper presents HYCO (Hybrid-Cooperative Learning), a modeling strategy that integrates physical and synthetic models through joint optimization while encouraging alignment between their predictions. The work presented here is based on the extended preprint [6], where a comprehensive theoretical and numerical analysis is developed.

In HYCO, both models are treated as active participants in the learning process, with mutual influence serving as a form of regularization. Through numerical experiments we demonstrated that HYCO can perform well on tasks ranging from solution recovery to parameter identification. These advantages are particularly evident when observations are sparse or spatially localized, where the interplay between models helps reconstruct missing information.

However, important limitations and open questions remain. The computational cost of training two models simultaneously means that efficiency gains from parallelization depend significantly on implementation details and problem structure. While our experiments show competitive performance, the overhead of maintaining two models is non-negligible. Furthermore, the theoretical analysis remains incomplete: convergence guarantees for the game-theoretical formulation are established in [6] only for simplified convex settings, and the behavior in general non-convex cases (which dominate practical applications) requires further investigation.

Looking ahead, several research directions emerge. HYCO's general architecture invites application to more complex scenarios: multiphysics problems where each model represents a different physical process, domain decomposition where models govern different spatial regions, and real-world applications in climate modeling, robotics, or medical imaging. Testing HYCO with advanced architectures such as transformers or neural operators as synthetic models remains an important validation step. On the theoretical front, extending convergence analysis to non-convex settings and investigating potential privacy vulnerabilities (as highlighted in federated learning literature) are critical challenges. Finally, while the game-theoretic perspective offers interpretability, further exploration of this connection may yield improved training algorithms or relaxation gap results analogous to recent work in neural network optimization.

In summary, HYCO represents an ongoing research effort that treats hybrid modeling through the lens of cooperative agents rather than constrained optimization. While promising results have

been demonstrated on benchmark problems, substantial work remains to fully understand the method's capabilities, limitations, and optimal deployment strategies.

Acknowledgements

This work was supported by the Alexander von Humboldt- Professorship program, the European Research Council (ERC) under the European Unions Horizon 2030 research and innovation programme (grant agreement NO: 101096251-CoDeFeL), the ModConFlex Marie Curie Action, the Transregio 154 Project Mathematical Modelling, Simulation and Optimization Using the Example of Gas Networks of the DFG, AFOSR Proposal 24IOE027, and grants PID2020-112617GB-C22 and TED2021131390B-I00 of the AEI (Spain), the Madrid Government- UAM Agreement for the Excellence of the University Research Staff in the context of the V PRICIT (Regional Programme of Research and Technological Innovation) and the project SURE-AI: The Norwegian Centre for Sustainable, Risk-Averse, and Ethical AI, grant 357482, Research Council of Norway.

References

[1] T. Bourdais and H. Owhadi. Minimal Variance Model Aggregation: A principled, non-intrusive, and versatile integration of black box models. *arXiv: 2409.17267*, 2025.

[2] S. L. Brunton, J. L. Proctor, and J. N. Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15), 2016.

[3] P. Gray and S. K. Scott. Autocatalytic reactions in the isothermal, continuous stirred tank reactor: Oscillations and instabilities in the system A + 2B → 3B; B → C. *Chemical Engineering Science*, 39(6), 1984.

[4] E. Kharazmi, Z. Zhang, and G. E. Karniadakis. Variational Physics-Informed Neural Networks For Solving Partial Differential Equations. *arXiv: 1912.00873*, 2019.

[5] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Fourier Neural Operator for Parametric Partial Differential Equations. In *International Conference on Learning Representations*, 2021.

[6] L. Liverani, M. Steynberg, and E. Zuazua. HYCO: Hybrid-Cooperative Learning for Data-Driven PDE Modeling. *arXiv: 2509.14123*, 2025.

[7] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3), 2021.

[8] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 54, 2017.

[9] S. Pakravan, P. A. Mistani, M. A. Aragon-Calvo, and F. Gibou. Solving inverse-PDE problems with physics-aware neural networks. *Journal of Computational Physics*, 440:110414, 2021.

[10] M. Raissi, P. Perdikaris, and G. 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, 2019.

[11] Y. Song, X. Yuan, and H. Yue. The ADMM-PINNs Algorithmic Framework for Nonsmooth PDE-Constrained Optimization: A Deep Learning Approach. *SIAM Journal on Scientific Computing*, 46(6), 2024.

[12] K. Xu and E. Darve. Physics constrained learning for data-driven inverse modeling from sparse observations. *Journal of Computational Physics*, 453:110938, 2022.

[13] L. Yang, X. Meng, and G. E. Karniadakis. B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data. *Journal of Computational Physics*, 425, 2021.

[14] C. Zhang and Y. Ma. *Ensemble Machine Learning: Methods and Applications*. Springer, 2012.

[15] R. Z. Zhang, C. E. Miles, X. Xie, and J. S. Lowengrub. BiLO: Bilevel Local Operator Learning for PDE inverse problems. *arXiv: 2404.17789*, 2025.