Efficient Knowledge Transfer in Federated Bayesian Optimization through Neural Network Surrogates

Alexander Gräfe* **Max Van Gemmeren*** **Paul Brunzema** **Sebastian Trimpe**
Institute for Data Science in Mechanical Engineering, RWTH Aachen University
Aachen, Germany
{alexander.graefe, paul.brunzema, trimpe}@dsme.rwth-aachen.de

Abstract

Federated Bayesian Optimization (FedBO) has emerged as a powerful paradigm for enhancing Bayesian Optimization (BO) in distributed, privacy-sensitive settings, enabling agents to accelerate the optimization of their local objectives by sharing knowledge about their similarities without exposing raw data. However, existing methods rely on restrictive assumptions, such as near-identical objective functions or shared Gaussian process hyperparameters that are often violated in heterogeneous real-world applications. We present GLOBAFED, a novel framework that enables efficient knowledge transfer across different but related objectives without compromising privacy. At its core is a probabilistic neural network surrogate with a shared feature-extracting backbone and agent-specific heads equipped with variational Bayesian last layers to provide uncertainty estimates. During optimization, agents use federated learning on the shared backbone to collaboratively build a common latent feature representation, while agent-specific heads adapt to their individual objectives. Our empirical results demonstrate that GLOBAFED significantly accelerates convergence and reaches better final performance compared to state-of-the-art FedBO baselines, and enables agents to warm-start their optimization from previously learned global features.

1 Introduction

Bayesian Optimization (BO) has become the de facto standard for the sample-efficient tuning of black-box systems where evaluations are expensive [12]. In real-world deployments, ranging from healthcare devices and wearable exoskeletons to autonomous vehicles, BO enables populations of deployed units to adapt to dynamic environments and heterogeneous user-defined objectives [5, 18, 19, 22, 24, 26, 28]. Crucially, while each individual unit (e.g., a specific exoskeleton) optimizes its own unique objective function, these functions remain inherently related across the fleet because the underlying systems share the same physical characteristics or structural dynamics. However, traditional BO methodologies are predominantly designed for single-instance optimization, treating each system as an isolated agent. This "isolated agent" paradigm forces every new agent to undergo an individual tuning process from scratch, which can be regarded as a cold-start penalty in situations where other agents (and thus data) are available *in principle*. In domains where every sample represents a physical experiment or a human-in-the-loop interaction, failing to exploit these cross-agent relationships introduces inefficiencies.

A fundamental challenge in leveraging these relationships is ensuring privacy protection. Because many applications involve sensitive personal data, like individual health metrics or driving patterns, knowledge sharing must occur without compromising local information. Federated Bayesian Op-

*Equal contribution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

timization (FedBO) has emerged to address these privacy concerns and facilitate rapid distributed optimization through inter-agent knowledge transfer [8, 9, 23, 34, 37].

However, current FedBO approaches often rely on overly strong assumptions about cross-agent similarity. In particular, they presume near-identical objective functions [8, 9] or closely matched hyperparameters of the underlying Gaussian Process (GP) models [37]. Such assumptions substantially limit their applicability in real-world settings, where heterogeneity across agents is the norm rather than the exception. For instance, differing user preferences (e.g., "sporty" versus "comfort" driving styles) or variations in physical scale (e.g., exoskeletons fitted to users of different body sizes) can induce objective functions that differ markedly in both magnitude and lengthscale. As a result, existing FedBO methods struggle to accommodate agents that are distinct yet share underlying structural characteristics. These limitations motivate this paper's guiding research question.

Guiding Research Question (RQ): How can we design a FedBO method that delivers *(i)* fast convergence with strong final performance, *(ii)* without necessitating strong assumptions, *(iii)* while ensuring privacy?

1.1 Approach Overview

To address the aforementioned research question, this work introduces Globally informed Federated Bayesian Optimization (GLOBAFED) (Figure 1). At the core of GLOBAFED is a probabilistic Neural Network (NN) surrogate model, specifically a Variational Bayesian Last Layer (VBLL) model [3, 14], consisting of a *shared backbone* and *agent-exclusive heads* that overcome the limitations of rigid GP priors (**RQ** *(ii)*). Knowledge transfer is facilitated through personalized Federated Learning (FL) [1, 6, 29]. The backbone learns a common feature representation of the underlying system behavior across all agents, while the local heads adapt to individual objective functions. To achieve this, agents train the full model on local data and exchange backbone weights for global aggregation, ensuring that sensitive raw data is never shared (**RQ** *(iii)*). Within each BO iteration, agents utilize the aggregated backbone and their local heads to construct individual acquisition functions that are informed by collective insights. Through the knowledge transfer via a shared latent space, GLOBAFED enables agents to achieve significantly faster convergence and strong final performance (**RQ** *(i)*).

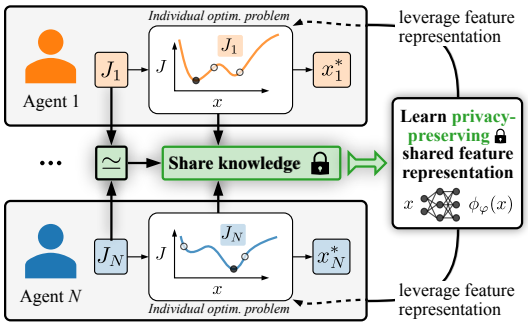


Figure 1: GLOBAFED. *K agents have different but related objective functions.* GLOBAFED *leverages federated learning combined with NN surrogate models for BO to learn a shared feature representation for efficient knowledge transfer while preserving privacy, resulting in faster convergence on the individual tasks of each agent.*

Contributions In summary, our contributions are:

- **C1** A novel FedBO framework, termed GLOBAFED, that accelerates optimization via privacy-preserving knowledge sharing while relying on weaker task similarity requirements than existing methods.
- **C2** A federated surrogate-training scheme that collaboratively learns a shared neural feature representation through backbone aggregation, while keeping each agent's observations and VBLL heads local for task-specific Bayesian uncertainty quantification.
- **C3** Comprehensive experiments showing that GLOBAFED exhibits considerable performance improvements both in convergence speed and final performance compared to existing state-of-the-art baselines when objectives are related but structurally different, performs on par on a heterogeneous Many-Task benchmark, and accelerates convergence for late-joining agents through warm-started shared features.

2 Problem Formulation

This section formalizes our **RQ**. We formalize our problem following the standard Many-task Optimization (MTO) setup [8, 9, 23, 34, 37]. We consider a set $\mathcal{I}$ consisting of N agents. Each agent

$i \in \mathcal{I}$ aims to identify a set of parameters $x_i^* \in \mathcal{X} \subseteq \mathbb{R}^d$ that maximizes its individual objective function $J_i : \mathbb{R}^d \to \mathbb{R}$:

$$\forall i \in \mathcal{I} : \; x_i^* = \arg\max_{x \in \mathcal{X}} J_i(x). \tag{1}$$

Observation Model We treat the objective functions J_i as black-box models, where agents lack knowledge of the analytical expressions. At each round $t \in \{1, \ldots, T\}$, where T is the query budget, every agent i independently selects a query point $x_t^{(i)}$ and receives noisy zeroth-order feedback

$$y_t^{(i)} = J_i\left(x_t^{(i)}\right) + \epsilon_t^{(i)}, \quad \epsilon_t^{(i)} \sim \mathcal{N}(0, \sigma_i^2), \tag{2}$$

where evaluations are corrupted by independent and identically distributed (i.i.d.) Gaussian noise with agent-specific variance σ_i^2. The functions have high sampling cost stating that such queries are expensive (e.g., obtained through a physical experiments).

Privacy Setting We assume that raw samples $(x_t^{(i)}, y_t^{(i)})$ contain privacy-sensitive information and must remain *agent-local*, i.e., they are never shared between agents. We focus exclusively on this aspect of data privacy to demonstrate that our approach to privacy-preserving knowledge transfer in FedBO via modern probabilistic NN surrogates works in principle. Consequently, we view defenses against other equally relevant privacy concerns, such as gradient inversion attacks [13, 15, 17, 36, 38], as largely orthogonal to our core contribution. Indeed, our framework can readily incorporate standard defenses against such attacks, including gradient perturbation [38], input mixing [16], or representation perturbation [32].

Task Similarity The objectives are related, meaning that samples from one function yield information about the others. Rather than assuming a fixed relational structure, we aim to develop a unified framework capable of capturing diverse scenarios, such as:

- *Near-Identical Objectives.* The objectives for any pair of agents i, j satisfy $J_i(x) \approx J_j(x)$, exhibiting only minor deviations arising from, e.g., manufacturing tolerances.
- *Linear Combinations of Basis Functions.* Objectives are formulated as $J_i(x) = \sum_{b=1}^{B} c_b^{(i)} f_b(x)$. The coefficients $c_{i,\mathrm{b}}$ vary among agents, modeling user preferences (e.g., balancing comfort vs. energy efficiency).
- *Parametric System Variations.* Objectives arise from identical system types operating under different physical parameters (e.g., exoskeletons for different body sizes).

Problem Statement The goal of this paper is to solve (1) for all agents by maximizing sample efficiency, leveraging relations between objective functions without assuming their specific type, and strictly adhering to privacy constraints by keeping raw samples agent-local to address **RQ**.

3 Related Work

This section reviews FedBO methods for the MTO problem described above. All methods follow the standard BO loop [12]: they train a surrogate model, select the next query point, and evaluate the objective. The main difference is how knowledge is transferred between agents, which determines the supported type of relationships between objective functions.

Federated Thompson Sampling (FTS) and its differentially private extension Differentially Private Federated Thompson Sampling with Distributed Exploration (DP-FTS-DE) [8, 9] let each agent fit a GP and approximate the posterior via Random Fourier Features (RFF), which allows them to share their GP without revealing private data. Agents randomly choose between using their own GP and another agent's RFF-weights when sampling. Consequently, an agent's GP must be directly informative for others, and FTS effectively requires *closely aligned objective functions*.

Federated Many-Task Bayesian Optimization (FMTBO) [37] instead shares GP hyperparameters, such as the length scale, rather than the GP itself. These hyperparameters are averaged using similarity-based weights. Each agent keeps a local GP (using only local data) and a global GP (using aggregated hyperparameters) to choose the next query. Thus, the relationship assumed by FMTBO is that *objective functions are generated by GPs with similar hyperparameters*.

Instead of GPs, Federated Data-driven Optimization with Ensemble Model Distillation (FD-EMD) [34] employs Radial Basis Function Networks (RBFNs) as surrogate models. Agents

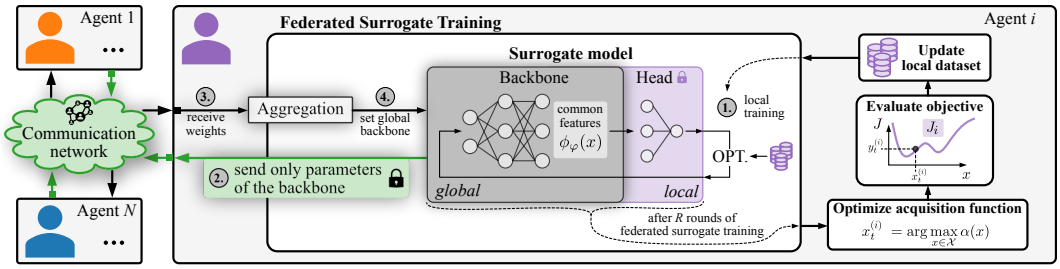


Figure 2: Overview of GLOBAFED. *In each iteration, agents* 1. *train their surrogate models on local data,* 2. *broadcast their learned backbone parameters,* 3. *receive weights from other agents, and* 4. *aggregate these weights and use them as initialization of the backbone for the next round. Through this, privacy-sensitive samples of the objective function remain agent-local.*

share weights with a server, which constructs an ensemble RBFN and distills a student network. Agents then combine this network with their local RBFN to select query points. FD-EMD's knowledge transfer therefore *requires that objectives are closely aligned*.

In Federated Bayesian Optimization via Optimization of an Implicit Acquisition Function (IAF-FBO) [23], agents maintain a local GP and distill it into a classifier that predicts which of two points has a higher acquisition value. These classifiers are shared and clustered based on weight similarity. Each cluster's weights are averaged, and the resulting classifiers guide the next queries. This mechanism hence implicitly assumes that *objectives are closely aligned within clusters*.

To summarize, existing FedBO methods rely on restrictive similarity assumptions. They either require closely aligned objectives [8, 9, 23, 34], which is already violated by simple linear combinations of basis functions (cf. Section 2), or they assume similar underlying GP hyperparameters [37], which basic input scaling, e.g., when one agent optimizes a fast system and another a slow one, breaks. In contrast, GLOBAFED removes these constraints and does not rely on restrictive assumptions. We achieve this via probabilistic NN surrogate models with a shared backbone and exclusive heads, which has not been proposed for FedBO before. This architecture facilitates a more flexible representation of objective function relations.

Methods related to MTO While several frameworks address multi-task optimization, they differ significantly from our setting in their assumptions regarding data locality and task relationships. Standard multi-task GP and BO approaches [33] typically rely on an explicit task-relation kernel to model dependencies between tasks. However, performing inference in these models requires centralized access to the data from all tasks, which is infeasible in the federated setting we consider. Similarly, contextual GP and BO methods [20] assume that tasks can be characterized by additional task-specific context or features. Beyond the challenge of defining such context in many practical applications, these methods still require a central entity to aggregate data from all agents to train the global surrogate, thereby violating the privacy constraints inherent in **RQ**. In contrast, GLOBAFED facilitates knowledge transfer without requiring centralized data access or predefined task descriptors.

Perrone et al. [27] propose a multi-task transfer learning approach for BO, combining task-specific Bayesian linear regression surrogates with a shared neural feature map. Both are learned jointly by optimizing a sum of task-specific marginal likelihoods, and the model can be retrained as new BO observations become available. The approach demonstrates transfer by using evaluations from previous hyperparameter optimization tasks to warm-start a new target task and by exploiting related auxiliary signals, such as training loss or evaluation cost. Similarly, GLOBAFED combines a shared representation with task-specific Bayesian heads. However, unlike [27], agents optimize concurrently using local observations and retain private VBLL heads, exchanging only backbone parameters for federated aggregation without pooling task data.

4 Globally Informed Federated BO

We present our method addressing the stated problem. Section 4.1 first discusses our proposal to leverage probabilistic NN surrogates and specifically VBLL models, as opposed to GPs, which a majority of FedBO approaches rely on. Section 4.2 then details the training of these surrogates to

achieve efficient knowledge transfer between agents while ensuring privacy. Lastly, in Section 4.3, we discuss the overall operation of GLOBAFED.

4.1 Decomposed NN Surrogates for Flexible Modeling

GLOBAFED uses a decomposed surrogate architecture that balances global knowledge transfer with agent-specific specialization. For each agent, the surrogate model is

$$\hat{J}_i(x) = \hat{\psi}^{(i)}_{\theta_i}(\hat{\phi}_\varphi(x)), \tag{3}$$

where $\hat{\phi}_\varphi : \mathbb{R}^d \to \mathbb{R}^m$ is a shared Multi-Layer Perceptron (MLP) backbone with parameters φ, and $\hat{\psi}^{(i)}_{\theta_i} : \mathbb{R}^m \to \mathbb{R}$ is an agent-specific head with parameters θ_i. This structure allows GLOBAFED to model various degrees of relationships between objectives, as illustrated by the following examples:

Example 4.1 (Equal Objectives)**.** *Consider tasks where $J_i = J_j$ for all i, j. In this setting, the surrogate admits an optimal solution in which all heads coincide, i.e., $\theta_i = \theta_*$ for all i, meaning all surrogate models are equal and approximate the same function.*

Example 4.2 (Linear Combination of Basis Functions)**.** *Suppose that the objectives are linear combinations of shared basis functions (cf. Section 2). In this case, the backbone would learn the shared basis functions and the heads the agent-specific linear transformation.*

Example 4.3 (Weakly Related or Unrelated Functions)**.** *For unrelated objectives, (3) avoids negative transfer by allowing the shared backbone to learn a task-agnostic mapping. By setting $m = d$, the backbone may reduce to an identity mapping, $\hat{\phi}_\varphi(x) \approx x$. If the local head $\hat{\psi}^{(i)}_{\theta_i}$ has sufficient expressive capacity, the surrogate becomes $\hat{J}_i(x) = \hat{\psi}^{(i)}_{\theta_i}(\hat{\phi}_\varphi(x)) \approx \hat{\psi}^{(i)}_{\theta_i}(x) \approx J_i(x)$.*

To explicitly model predictive uncertainty, we adopt VBLL models [3, 14] as the final layer of the surrogate's head. VBLL models are structurally similar to standard MLPs, but replace the deterministic last linear layer with a Bayesian Linear Regression (BLR) layer. Unlike other Bayesian last layer approaches that rely on explicit conditioning to infer the posterior over the final-layer weights [11, 31, 35], VBLLs directly learn the parameters of a variational posterior $q_i(w_i) = \mathcal{N}(\bar{w}_i, S_i)$. Given this variational posterior and the observation model in (2), conjugacy yields a tractable posterior predictive distribution for each agent's surrogate model of the individual objective:

$$\hat{J}_i(x) \sim \mathcal{N}(\bar{w}_i^\top f^{(i)}_{\tilde{\theta}_i,\varphi}(x), f^{(i)}_{\tilde{\theta}_i,\varphi}(x)^\top S_i f^{(i)}_{\tilde{\theta}_i,\varphi}(x) + \sigma_i^2), \tag{4}$$

where $f^{(i)}_{\tilde{\theta}_i,\varphi}(x) = \hat{g}^{(i)}_{\tilde{\theta}_i}(\hat{\phi}_\varphi(x))$ and $\hat{g}^{(i)}_{\tilde{\theta}_i}(x)$ comprises all head layers except the last one with parameters $\tilde{\theta}_i$. Hence, the learnable head parameters are $\theta_i = \{\tilde{\theta}_i, \bar{w}_i, S_i, \sigma_i^2\}$.

4.2 Surrogate Model Training Ensuring Privacy and Knowledge Transfer

The effectiveness of GLOBAFED relies on the ability of the surrogate models to capitalize on shared structural similarities across agents without compromising the privacy of local observations. Having defined the decomposed surrogate architecture in Section 4.1, we now describe a tailored FL procedure that enables efficient global knowledge transfer while ensuring privacy.

Local Surrogate Training Each agent trains a surrogate model on its local dataset $\mathcal{D}_i = \{(x_t^{(i)}, y_t^{(i)})\}_{t=1}^T$, which consists of noisy evaluations (cf. (2)). Training is performed by maximizing the variational lower bound on the marginal likelihood [3, 14], given by

$$\mathcal{L}(\theta_i, \varphi) \coloneqq \frac{1}{T}\sum_{t=1}^{T}\left[\log \mathcal{N}\Big(y_t \mid \bar{w}_i^\top f^{(i)}_{\tilde{\theta}_i,\varphi}(x), \sigma_i^2\Big) - \frac{1}{2} f^{(i)}_{\tilde{\theta}_i,\varphi}(x)^\top S_i f^{(i)}_{\tilde{\theta}_i,\varphi}(x)\sigma_i^{-2}\right] - \mathcal{R}(\theta_i, \varphi), \tag{5}$$

$$\text{where} \quad \mathcal{R}(\theta_i, \varphi) \coloneqq \frac{1}{T}\Big[\mathrm{KL}(q_i(w_i) \,\|\, p(w_i)) - \log p\left(\{\tilde{\theta}_i, \varphi\}\right) - \log p\big(\sigma_i^2\big)\Big].$$

where $\mathrm{KL}(q_i(w_i)||p(w_i))$ is the Kullback-Leibler divergence between the variational posterior and the prior, and negative log-priors imposed on $\{\tilde{\theta}_i, \varphi\}$ and σ_i^2. In practice, additional regularization terms are applied to the backbone and the noise.

We refer to Harrison et al. [14, Algorithm 1] for the full formulation as well as the details on the parameterization of q_i through Cholesky factors of S_i. The objective in (5) jointly optimizes the variational posterior together with the remaining head parameters $\tilde{\theta}_i$ and the backbone parameters φ. All optimization is performed locally, and no raw data are shared between agents.

Federated Knowledge Transfer To enable knowledge transfer across agents, we embed the training procedure in a personalized FL loop [1, 6, 29] shown in Algorithm 1. After randomly initializing the head and backbone parameters, agents perform R communication rounds. In each round, agents update their local models by optimizing (5) on their private datasets. Subsequently, only the backbone parameters are shared and aggregated using an FL method. In our implementation, we employ FedOpt [30], which treats local weight updates as pseudo-gradients and aggregates them using an ADAM optimizer to account for heterogenous data distributions caused by different objectives. The resulting aggregated backbone defines a shared feature representation for the next round. The end of the loop is always a fine-tuning of the head parameters given the aggregated backbone.

Algorithm 1 Federated Surrogate Training on Agent i

Input: Local dataset $\mathcal{D}_i$, Rounds R
Initialize: $\theta_i[0]$, $\varphi[0]$ % Random initialization
function FedSurrogateTraining[$\mathcal{D}_i$]
  for $r = 0$ **to** $R-1$ **do**
    % Local surrogate training maximizing (5)
    $(\theta_i[r+\frac{1}{2}], \varphi_i[r+\frac{1}{2}]) \leftarrow$ Train($[\theta_i[r], \varphi[r]], \mathcal{D}_i$)
    % Communication & aggregation
    Send $\varphi_i[r+\frac{1}{2}]$ to communication network
    $\Omega \leftarrow \{\varphi_j[r+\frac{1}{2}] \mid \forall j \in \mathcal{I}\} \leftarrow$ Receive()
    $\varphi[r+1] \leftarrow$ AggregateBackbones($\Omega, \varphi[r]$)
    % Surrogate head finetuning
    $\theta_i[r+1] \leftarrow$ TrainHead($[\theta_i[r+\frac{1}{2}], \varphi[r+1]], \mathcal{D}_i$)
  end for
  return $\theta_i[R], \varphi[R]$
end function

4.3 Overall GLOBAFED Operation Through Agent-local Thompson Sampling

The overall agent operation follows the standard BO framework for VBLL models as surrogates [3] and is shown in Figure 1 and Algorithm 2. At every iteration t, the agents use their obtained surrogate models from the previous section to build an acquisition function $\alpha(\hat{J}_i(x), \mathcal{D}_i)$ which is globally maximized for the next query location. While VBLL models allow for a variety of acquisition functions, we adopt Thompson sampling as a convenient acquisition function for such NN surrogate models. For this, we first sample a realization of the last layer weight distribution $w_t^{(i)} \sim q_i(w_i)$. The Thompson sample is then constructed as a linear model using the sample from the last layer as $\alpha_{TS}(x) = (w_t^{(i)})^\top \hat{g}_{\hat{\theta}_i}^{(i)}(\hat{\phi}_\varphi(x))$. It is then maximized using ten multi-starts with L-BFGS-B [4] leading to the next query point $x_t^{(i)}$. Each agent i then queries the objective function at their respective new query locations to get a noisy measurement and updates its dataset $\mathcal{D}_i$. The agents then continue with the next iteration and train the surrogate models on the new datasets.

Algorithm 2 BO Loop of Agent i

$\mathcal{D}_i \leftarrow$ Initial set of observations
for $t = |\mathcal{D}_i|$ **to** T **do**
  $(\theta_i[t], \varphi[t]) \leftarrow$ FedSurrogateTraining($\mathcal{D}_i$)
  Build local surrogate $\hat{J}_i(x)$ from $\theta_i[t]$ and $\varphi[t]$
  $x_t^{(i)} \leftarrow \arg\max_{x \in \mathcal{X}} \alpha(\hat{J}_i(x), \mathcal{D}_i)$
  $y_t^{(i)} \leftarrow$ QueryObjective($x_t^{(i)}$)
  $\mathcal{D}_i \leftarrow \mathcal{D}_i \cup \{(x_t^{(i)}, y_t^{(i)})\}$
end for
return $\arg\max_{(x,y) \in \mathcal{D}_i} y$

5 Experiments

In this section, we demonstrate that our approach accelerates the optimization process in MTO problems through efficient knowledge transfer[2]. Our main results are:

- GLOBAFED substantially outperforms existing FedBO methods and agent-local BO, which does not leverage knowledge transfer, under the same optimization budget.
- GLOBAFED allows agents that join later to immediately exploit shared features encoding information from earlier agents for effective warm-starts.
- GLOBAFED is robust to negative knowledge transfer achieving comparable performance to existing FedBO methods on an MTO benchmark with multiple task groups.

[2]Code is available at https://github.com/Data-Science-in-Mechanical-Engineering/GlobaFed.

5.1 Experimental Setup

Benchmark Problems We consider two sets of benchmark problems, which we summarize here and detail in Appendix B. The first benchmark, referred to as the related-task benchmark (Section 5.2), comprises nine tasks derived from objective functions that are systematically transformed to yield distinct yet related objectives for each agent. The first five tasks are based on standard benchmark functions in which both the input and the output value (i.e., the location and height of the optimum) are shifted. These include the Ackley function (in 1D, 5D, and 10D), the Branin function (2D) and the Hartmann function (6D). The *combination* (Comb.) task is a linear combination of five base functions (cf. Section 4.1) with 1, 5 and 10 input dimensions. The last task is a controller tuning problem (5D) as a real-world example, where the agents tune controllers for cart-pole systems that differ in their physical parameters. Figures with selected benchmarks report Ackley (1D), cart-pole and Comb. (10D) as representative examples; full results for the remaining functions are included either in the main figures or in Section D.

The second benchmark is the many-task benchmark [7] (Section 5.3). It comprises 18 distinct 10-dimensional objective functions that, unlike the related-task benchmark, may not share common structure (cf. Section B). While this benchmark does not reflect the primary use case of GLOBAFED, which targets settings with shared task structure, it provides a valuable testbed for evaluating robustness to potential negative knowledge transfer in more heterogeneous scenarios.

Baselines We compare our approach against baselines with different levels of information sharing. First, as alternative FedBO baselines, we compare our method against DP-FTS-DE [9], FMTBO [37], FD-EMD [34] and IAF-FBO [23]. Second, we include *agent-local BO* with either GP surrogate model (L-GP) or a VBLL surrogate (L-VBLL) to compare GLOBAFED with methods without any knowledge sharing. Third, to assess the influence of privacy constraints on performance, we include a *centralized version* of our approach, where we train the surrogate model on the entire dataset, i.e., with data sharing. The full details on its training and results can be found in Appendix E.

Hyperparameters If not specified otherwise, we perform all experiments with 10 agents and d initial points. To evaluate the generalization capabilities, we tuned the hyperparameters, which can be found in Appendix F, for all NN-based approaches once on the shifted 10-dimensional Ackley problem from the related-task benchmark. These hyperparameters were then kept fixed for all other benchmark problems to test whether a single configuration performs robustly across tasks.

Beyond our main results, we conduct a comprehensive ablation study on key hyperparameters to quantify their impact on performance. In this section, we highlight the effect of varying the number of agents, while the remaining ablation results are deferred to Appendix F.

Metrics We measure each agent's simple regret at iteration t to evaluate how quickly each approach identifies high-performing parameters. Results for the related-task benchmark report the average simple regret across agents, since the magnitudes of objectives are comparable. For the many-task benchmark, where objectives' magnitudes differ substantially and regret scales vary, we report the simple regret for each agent individually.

5.2 Related-Task Benchmark

This section evaluates GLOBAFED on the related-task benchmark, comparing it with related FedBO methods and agent-local baselines. We first consider a standard setting where all agents start simultaneously, and then a warm-start setting where one agent joins the optimization later.

Figure 3 compares GLOBAFED with related FedBO approaches. In low dimensions, all methods except IAF-FBO and FD-EMD quickly reach low regret, whereas in higher dimensions GLOBAFED clearly dominates, converging faster and achieving the lowest final regret of all methods across all benchmarks, including rapid convergence on the controller optimization task. On Combination, FMTBO attains lower regret initially, yet GLOBAFED overtakes it after a few iterations. These results show that the NN surrogate learns expressive shared features that the head can adapt to guide the search. As a result, GLOBAFED enables effective knowledge transfer, particularly in high dimensions. In contrast, the restrictive assumptions of DP-FTS-DE, FMTBO, IAF-FBO and FD-EMD (cf. Section 3) limit information sharing about task relations, which slows down optimization.

Comparison with agent-local baselines We compare GLOBAFED with agent-local approaches in Figure 4. For low-dimensional problems, our method matches agent-local L-

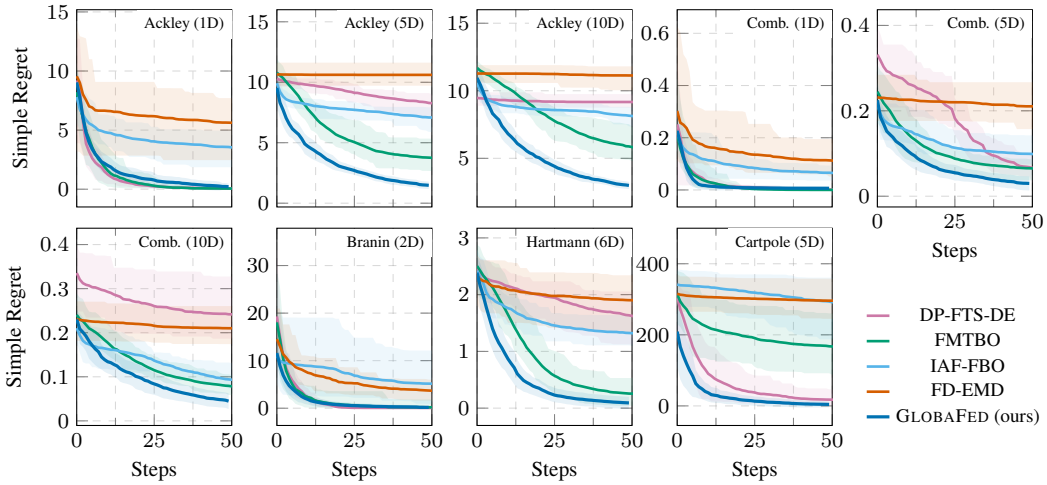


Figure 3: Performance of FedBO baselines. *All approaches perform similarly in low dimensions, except IAF-FBO and FD-EMD, which do not converge to the optimum.* GlobaFed *converges much faster and to lower final regrets for high-dimensional problems than all FedBO baselines.*

GP and outperforms agent-local L-VBLL. As dimensionality increases, GlobaFed substantially outperforms all agent-local variants, showing faster convergence and lower final regret.

Comparison with FedBO baselines These results suggest that simple low-dimensional problems are well handled by traditional BO with GPs and require no knowledge transfer. In high dimensions, however, knowledge transfer significantly speeds up convergence. Since agents sample different initial points, they explore different regions of the search space and collectively gather more information than purely agent-local methods.

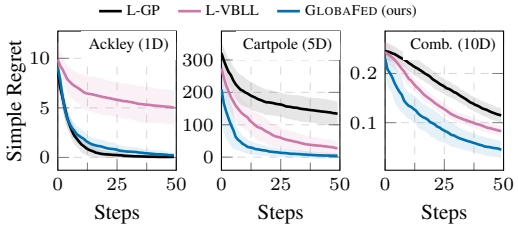


Figure 4: GlobaFed versus other agent-local approaches. *Remaining benchmark objective functions are shown in Section D, Figure 8.* GlobaFed *demonstrates faster convergence and better final performance than agent-local approaches through effective knowledge transfer.*

Warmstarting Finally, we study a scenario where an agent joins the optimization at a later iteration and leverages the collective knowledge of existing agents to accelerate its search. We refer to late-joining agents as warm-started and to those present from the beginning as cold-started. Our experiments compare the warm-starting performance of GlobaFed, FMTBO, DP-FTS-DE, IAF-FBO and FD-EMD.

Figure 5 shows our results. Warm- and cold-started agents exhibit similar initial regret. In low-dimensional problems, all methods reduce regret at comparable rates compared to the cold-started agents, yielding no notable warm-start advantage. In high-dimensional problems, however, GlobaFed enables warm-started agents to match the regret of cold-started agents within only a few iterations, whereas in FMTBO, DP-FTS-DE, IAF-FBO and FD-EMD, warm-started agents improve at roughly the same pace as the cold-start agents.

These experiments demonstrate that the shared features learned by GlobaFed encode global knowledge that generalizes to unseen objectives. This efficient knowledge transfer allows warm-started agents to train accurate local heads with just a few observations and accelerate optimization.

5.3 Many-Task Benchmark

This section presents the results of our approach on the many-task benchmark shown in Table 1. We use this benchmark to assess how well our algorithm performs in a setting where only a subset of tasks is related. Our approach achieves results comparable to recent FedBO methods. Generally, the regrets that all approaches achieve are of similar magnitude. While FD-EMD wins most tasks, ours achieves the best mean rank through more consistent performance. Our approach consistently outperforms DP-FTS-DE and FMTBO, except on a few tasks such as that of agent 6.

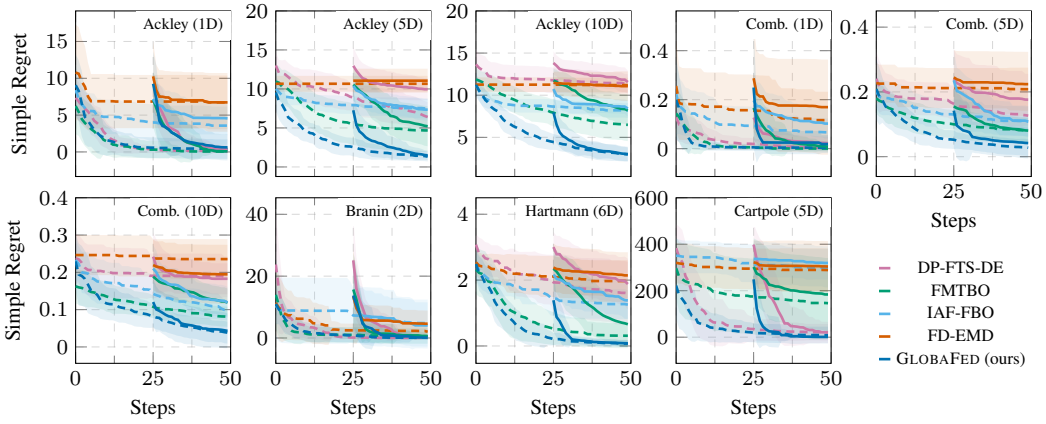


Figure 5: Performance for warmstarting of agents. *Dashed lines represent initial agents. Solid lines represent warm-started agents; shaded regions show one standard deviation. Via* GlobaFed*, warm-started agents converge much faster.*

Table 1: 10-dimensional many-tasks results (lower is better). Cell shading is log-scaled relative to the best method per task (darker = better; white = at least 3× worse than the best). The best result per task is in bold.

Task	GlobaFed (ours)	L-VBLL	L-GP	FMTBO	DP-FTS-DE	IAF-FBO	FD-EMD
1	9.05×10^{-1}	1.01	1.04	1.00	1.36	9.44×10^{-1}	**7.06×10^{-1}**
2	7.33×10^{1}	1.16×10^{2}	1.66×10^{2}	1.08×10^{2}	3.71×10^{2}	8.89×10^{1}	**6.68×10^{1}**
3	4.03	7.58	1.16×10^{1}	6.57	1.54×10^{1}	6.53	**3.23**
4	7.32×10^{1}	1.36×10^{2}	1.76×10^{2}	1.23×10^{2}	3.33×10^{2}	9.18×10^{1}	**6.63×10^{1}**
5	9.28	**7.61**	1.24×10^{1}	7.70	1.89×10^{1}	1.68×10^{1}	1.84×10^{1}
6	2.60×10^{3}	2.67×10^{3}	2.53×10^{3}	**2.22×10^{3}**	2.72×10^{3}	2.77×10^{3}	2.77×10^{3}
7	7.75×10^{1}	1.18×10^{2}	1.73×10^{2}	1.06×10^{2}	4.32×10^{2}	8.80×10^{1}	**6.34×10^{1}**
8	1.76×10^{2}	1.50×10^{2}	2.46×10^{2}	1.78×10^{2}	1.02×10^{4}	5.81×10^{2}	**5.89×10^{1}**
9	4.29	8.06	1.19×10^{1}	7.20	1.53×10^{1}	6.41	**3.51**
10	**1.54×10^{1}**	4.02×10^{1}	1.89×10^{2}	9.64×10^{5}	1.25×10^{8}	3.45×10^{5}	5.82×10^{7}
11	4.19	7.82	1.12×10^{1}	7.29	1.59×10^{1}	6.37	**3.36**
12	2.59	4.85	8.75	5.78	1.17×10^{1}	3.91	**1.98**
13	**1.50×10^{1}**	4.31×10^{1}	2.16×10^{2}	1.39×10^{6}	1.41×10^{8}	4.06×10^{5}	5.75×10^{7}
14	7.65×10^{1}	1.36×10^{2}	1.95×10^{2}	1.33×10^{2}	4.38×10^{2}	1.10×10^{2}	**6.76×10^{1}**
15	9.55×10^{-1}	9.86×10^{-1}	1.06	1.02	1.41	9.89×10^{-1}	**9.10×10^{-1}**
16	2.05	4.28	6.60	4.42	9.96	3.01	**1.62**
17	7.23×10^{1}	1.28×10^{2}	1.86×10^{2}	1.14×10^{2}	4.31×10^{2}	1.01×10^{2}	**6.35×10^{1}**
18	2.67×10^{3}	2.68×10^{3}	2.52×10^{3}	**2.10×10^{3}**	2.87×10^{3}	2.69×10^{3}	2.69×10^{3}
Avg. Rank	**2.11**	3.94	5.06	3.83	6.89	3.78	2.39

To determine whether learning a shared representation becomes detrimental under such heterogeneous conditions, we compared GlobaFed with the agent-local baselines L-VBLL and L-GP, neither of which transfers information across agents. If the shared representation induced substantial negative transfer under weakly or non related functions, we would expect GlobaFed to consistently underperform these non-sharing baselines, particularly on weakly related tasks. Instead, GlobaFed outperforms the local baselines on 14 of the 18 tasks. L-VBLL performs only better on Tasks 5 and 8 and L-GP on tasks 6 and 18, which may indicate task-specific interference, but not systematic negative transfer. Overall, these results suggest that GlobaFed can exploit shared structure when present while remaining robust to substantial task heterogeneity

These results show that despite lacking explicit mechanisms to identify similar client tasks, GlobaFed's federated surrogate training appears reasonably robust to negative knowledge transfer. This is noteworthy as most FedBO baselines employ dedicated strategies to identify similar tasks and are hence tailored to this benchmark, while our approach is not and relies on learning them directly.

5.4 Influence of the number of agents on performance

To evaluate the influence of the number of agents participating in the federated surrogate training, we varied the number of agents with results shown in Figure 6. We observe that increasing the number of

participating agents improves overall performance, speeding up convergence and reducing variance. The gains become more pronounced as the problem dimensionality grows.

These results highlight the effectiveness of leveraging many agents for richer, more diverse exploration of the search space. With more agents, more data can be used to learn the shared features, directly translating into more accurate shared features and improved optimization performance. However, this benefit comes at the cost of increased communication overhead, as a larger number of agents must exchange information. In practice, the optimal number of agents is therefore problem-dependent and should balance the advantages of enhanced collaboration against the additional communication cost.

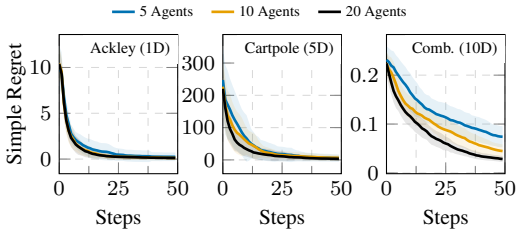


Figure 6: GLOBAFED versus number of agents. *Remaining benchmark objective functions are shown in Section F, Figure 10. More agents increase convergence speed and decrease variance of the outcome.*

6 Limitations and Future Work

GLOBAFED has been shown to significantly accelerate BO in MTO settings by efficiently exchanging knowledge between agents. Nevertheless, it also has limitations that motivate future research.

Privacy As outlined in Section 2, GLOBAFED preserves privacy in the sense that raw query point-observation pairs are never shared across agents and does not defend against other mechanisms such as gradient inversion attacks [13, 15, 17, 36, 38]. However, as our framework is independent of the specific FL method, it can in principle support defenses such as gradient perturbation [38], input mixing [16], or representation perturbation [32]. In such scenarios, we expect a trade-off between the strength of the privacy guarantees and optimization performance.

Communication Our ablations in Appendix F show that performance degrades when the number of communication rounds per BO iteration is reduced, particularly in higher dimensions. This indicates that GLOBAFED currently relies on frequent synchronization to learn effective shared backbones. In realistic deployments with bandwidth constraints, this requirement may limit scalability and increase optimization time. Future work could integrate FL techniques that reduce communication demands, such as device selection [21, Ch. 6.2.2], into GLOBAFED.

Asynchronous Optimization Agents perform their experiments and surrogate training in a fully synchronized manner. However, many practical scenarios feature heterogeneous experiment durations and computational resources, so stragglers can delay global updates. This motivates an asynchronous extension of GLOBAFED, which is a promising direction for future research.

7 Conclusion

This work presented GLOBAFED, a novel FedBO method that learns a global feature representation to share knowledge between agents during optimization while keeping all raw observations local. GLOBAFED combines a shared MLP backbone with agent-specific VBLL heads and trains the backbone via standard FL, enabling knowledge transfer without sharing data. Across a suite of benchmarks including shifted functions, linear combinations of base functions and a cart-pole controller tuning task, GLOBAFED consistently achieves faster convergence and better final performance than existing FedBO methods such as DP-FTS-DE and FMTBO, as well as agent-local BO with GP or NN surrogates. It furthermore matches the performance of a centralized variant that trains on pooled data (Appendix E), indicating that aggregating backbone parameters via FL is sufficient to capture the benefits of data sharing. On a many-task benchmark with partially related and unrelated tasks, GLOBAFED attains performance comparable to recent clustering-based FedBO methods while using a single shared backbone. Furthermore, late-arriving agents can warm-start their optimization using the shared backbone learned from earlier agents. Together, these results demonstrate that shared NN representations provide an effective and flexible mechanism for knowledge transfer in FedBO, enabling fast convergence without creating overly restrictive assumptions on the setting all while ensuring privacy.

Acknowledgments and Disclosure of Funding

This work was supported in part by the German Research Foundation (DFG) through Priority Programme 1914 (project number 315036061), Research Training Group RTG 2236/2 (UnRAVeL), and project number 556142469 (ParDyBO). Additional support was provided by the German Federal Ministry of Research, Technology and Space (BMFTR) through the Robotics Institute Germany (RIG).

The authors gratefully acknowledge the computing time provided to them at the NHR Center NHR4CES at RWTH Aachen University (project numbers p0021919 and p0027430). This is funded by the Federal Ministry of Research, Technology and Space, and the state governments participating on the basis of the resolutions of the GWK for national high performance computing at universities (www.nhr-verein.de/unsere-partner).

References

[1] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

[2] Andrew G. Barto, Richard S. Sutton, and Charles W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13(5), 1983.

[3] Paul Brunzema, Mikkel Jordahn, John Willes, Sebastian Trimpe, Jasper Snoek, and James Harrison. Bayesian optimization via continual variational last layer training. In *International Conference on Learning Representations (ICLR)*, 2025.

[4] Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5), 1995.

[5] Ankush Chakrabarty, Emilio Maddalena, Hongtao Qiao, and Christopher Laughman. Scalable Bayesian optimization for model calibration: Case study on coupled building and HVAC dynamics. *Energy and Buildings*, 253, 2021.

[6] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *International conference on machine learning*, 2021.

[7] Bingshui Da, Yew-Soon Ong, Liang Feng, A Kai Qin, Abhishek Gupta, Zexuan Zhu, Chuan-Kang Ting, Ke Tang, and Xin Yao. Evolutionary multitasking for single-objective continuous optimization: Benchmark problems, performance metric, and baseline results. *arXiv preprint arXiv:1706.03470*, 2017.

[8] Zhongxiang Dai, Bryan Kian Hsiang Low, and Patrick Jaillet. Federated Bayesian optimization via Thompson sampling. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 2020.

[9] Zhongxiang Dai, Bryan Kian Hsiang Low, and Patrick Jaillet. Differentially private federated Bayesian optimization with distributed exploration. *Advances in Neural Information Processing Systems (NeurIPS)*, 34, 2021.

[10] David Eriksson, Michael Pearce, Jacob Gardner, Ryan D Turner, and Matthias Poloczek. Scalable global optimization via local Bayesian optimization. *Advances in neural information processing systems*, 32, 2019.

[11] Felix Fiedler and Sergio Lucia. Improved uncertainty quantification for neural networks with Bayesian last layer. *IEEE Access*, 2023.

[12] Roman Garnett. *Bayesian Optimization*. Cambridge University Press, 2023.

[13] Jiahui Geng, Yongli Mou, Qing Li, Feifei Li, Oya Beyan, Stefan Decker, and Chunming Rong. Improved gradient inversion attacks and defenses in federated learning. *IEEE Transactions on Big Data*, 10(6):839–850, 2023.

[14] James Harrison, John Willes, and Jasper Snoek. Variational Bayesian last layers. In *International Conference on Learning Representations (ICLR)*, 2024.

[15] Ali Hatamizadeh, Hongxu Yin, Pavlo Molchanov, Andriy Myronenko, Wenqi Li, Prerna Dogra, Andrew Feng, Mona G Flores, Jan Kautz, Daguang Xu, et al. Do gradient inversion attacks make federated learning unsafe? *IEEE Transactions on Medical Imaging*, 42(7):2044–2056, 2023.

[16] Yangsibo Huang, Zhao Song, Kai Li, and Sanjeev Arora. Instahide: Instance-hiding schemes for private distributed learning. In *International conference on machine learning*, pages 4507–4518. PMLR, 2020.

[17] Yangsibo Huang, Samyak Gupta, Zhao Song, Kai Li, and Sanjeev Arora. Evaluating gradient inversion attacks and defenses in federated learning. *Advances in neural information processing systems*, 34:7232–7241, 2021.

[18] GilHwan Kim, Haider Chishty, and Fabrizio Sergi. Pilot testing of dynamic Bayesian optimization for exoskeleton-assisted training of propulsion mechanics. In *International Conference On Rehabilitation Robotics (ICORR)*. IEEE, 2025.

[19] Myunghee Kim, Charles Liu, Jinsoo Kim, Sangjun Lee, Adham Meguid, Conor J Walsh, and Scott Kuindersma. Bayesian optimization of soft exosuits using a metabolic estimator stopping process. In *International Conference on Robotics and Automation (ICRA)*. IEEE, 2019.

[20] Andreas Krause and Cheng Ong. Contextual Gaussian process bandit optimization. *Advances in Neural Information Processing Systems (NeurIPS)*, 24, 2011.

[21] Le Liang, Shi Jin, Hao Ye, and Geoffrey Ye Li. *Wireless Communications and Machine Learning*. Cambridge University Press, 2025.

[22] Xingbin Lin, Qi Guo, Deyu Yuan, and Min Gao. Bayesian optimization framework for HVAC system control. *Buildings*, 13(2), 2023.

[23] Qiqi Liu, Yaochu Jin, and Guodong Chen. Optimization of an implicit acquisition function for federated Bayesian many-task optimization. *IEEE Transactions on Evolutionary Computation*, 2025.

[24] Qiugang Lu, Leonardo D González, Ranjeet Kumar, and Victor M Zavala. Bayesian optimization with reference models: A case study in MPC for HVAC central plants. *Computers & Chemical Engineering*, 154, 2021.

[25] Alonso Marco, Philipp Hennig, Jeannette Bohg, Stefan Schaal, and Sebastian Trimpe. Automatic LQR tuning based on Gaussian process global optimization. In *International Conference on Robotics and Automation (ICRA)*. IEEE, 2016.

[26] Johanna Menn, Pietro Pelizzari, Michael Fleps-Dezasse, and Sebastian Trimpe. Lipschitz safe Bayesian optimization for automotive control. In *IEEE Conference on Decision and Control (CDC)*, 2024.

[27] Valerio Perrone, Rodolphe Jenatton, Matthias W Seeger, and Cédric Archambeau. Scalable hyperparameter transfer learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 31, 2018.

[28] Utkarsh Phatale and Suresh Limkar. Enhancing medical imaging diagnosis with deep learning and Bayesian optimization. In *International Conference on Information System Design and Intelligent Applications*. Springer, 2024.

[29] Krishna Pillutla, Kshitiz Malik, Abdel-Rahman Mohamed, Mike Rabbat, Maziar Sanjabi, and Lin Xiao. Federated learning with partial model personalization. In *International Conference on Machine Learning*, 2022.

[30] Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and Hugh Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations (ICLR)*, 2021.

[31] Jasper Snoek, Oren Rippel, Kevin Swersky, Ryan Kiros, Nadathur Satish, Narayanan Sundaram, Mostofa Patwary, Mr Prabhat, and Ryan Adams. Scalable Bayesian optimization using deep neural networks. In *International Conference on Machine Learning (ICML)*, 2015.

[32] Jingwei Sun, Ang Li, Binghui Wang, Huanrui Yang, Hai Li, and Yiran Chen. Soteria: Provable defense against privacy leakage in federated learning from representation perspective. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9311–9319, 2021.

[33] Kevin Swersky, Jasper Snoek, and Ryan P Adams. Multi-task Bayesian optimization. *Advances in Neural Information Processing Systems (NeurIPS)*, 26, 2013.

[34] Xilu Wang and Yaochu Jin. Distilling ensemble surrogates for federated data-driven many-task optimization. *IEEE Transactions on Evolutionary Computation*, 2024.

[35] Joe Watson, Jihao Andreas Lin, Pascal Klink, Joni Pajarinen, and Jan Peters. Latent derivative Bayesian last layer networks. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021.

[36] Hui Zhou, Zheng Qin, Yipeng Zou, Xin Deng, Jiaxin Jiang, Ge Xiao, and Hao Chen. Do adversarial perturbations truly mitigate gradient inversion in federated learning? In *2025 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2025.

[37] Hangyu Zhu, Xilu Wang, and Yaochu Jin. Federated many-task Bayesian optimization. *IEEE Transactions on Evolutionary Computation*, 28(4), 2023.

[38] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. *Advances in neural information processing systems*, 32, 2019.

A Surrogate Training Subroutines

Algorithms 3 to 5 detail the subroutines used in Algorithm 1. We denote the local and head-only epoch counts by E and E_{head}, respectively. For a minibatch $\mathcal{B}$, $\ell_i(\theta_i, \varphi; \mathcal{B})$ denotes the negative variational lower bound in (5), with the data term evaluated on $\mathcal{B}$ and regularization scaled by the full local dataset size $|\mathcal{D}_i|$. The local optimizer uses Adam with gradient clipping. Its state is initialized anew for each call to TRAIN or TRAINHEAD. The aggregation optimizer instead retains its state across communication rounds within one call to FEDSURROGATETRAINING.

Algorithm 3 Local Surrogate Training on Agent i

Input: Head and backbone parameters $[\theta_i, \varphi]$, local dataset $\mathcal{D}_i$, epochs E
function Train$[[\theta_i, \varphi], \mathcal{D}_i]$
  Initialize local optimizer state s
  for $e = 1$ **to** E **do**
    Form shuffled minibatches from $\mathcal{D}_i$
    for each minibatch $\mathcal{B}$ **do**
      $g \leftarrow \nabla_{(\theta_i, \varphi)} \ell_i(\theta_i, \varphi; \mathcal{B})$
      % Jointly update head and backbone
      $([\theta_i, \varphi], s) \leftarrow$ LocalOptimizerStep$([\theta_i, \varphi], g, s)$
    end for
  end for
  return θ_i, φ
end function

Algorithm 4 Backbone Aggregation Using FedOpt [30]

Input: Received backbones $\Omega = \{\varphi_j \mid j \in \mathcal{I}\}$, previous backbone φ, dataset sizes $n_j = |\mathcal{D}_j|$
Persistent state: Aggregation optimizer state s_{agg}, initialized before the first round
function AggregateBackbones$[\Omega, \varphi]$
  $\Delta \leftarrow 0$
  for each agent $j \in \mathcal{I}$ **do**
    $w_j \leftarrow n_j / \sum_{k \in \mathcal{I}} n_k$
    $\Delta \leftarrow \Delta + w_j(\varphi_j - \varphi)$
  end for
  $g_{\text{agg}} \leftarrow -\Delta$ % Treat the averaged update as a negative gradient
  $(\varphi, s_{\text{agg}}) \leftarrow$ AdamStep$(\varphi, g_{\text{agg}}, s_{\text{agg}})$
  return φ
end function

Algorithm 5 Local Head Finetuning on Agent i

Input: Head and aggregated backbone parameters $[\theta_i, \varphi]$, local dataset $\mathcal{D}_i$, epochs E_{head}
function TrainHead$[[\theta_i, \varphi], \mathcal{D}_i]$
  Freeze φ and initialize local optimizer state s
  for $e = 1$ **to** E_{head} **do**
    Form shuffled minibatches from $\mathcal{D}_i$
    for each minibatch $\mathcal{B}$ **do**
      $g \leftarrow \nabla_{\theta_i} \ell_i(\theta_i, \varphi; \mathcal{B})$
      % Adapt the local head to the aggregated features
      $(\theta_i, s) \leftarrow$ LocalOptimizerStep(θ_i, g, s)
    end for
  end for
  return θ_i
end function

B Benchmark Problems

Shift The first type of transformation we use to simulate correlation is a simple shift applied to both the input and output of the shared objective, ensuring that each client has a distinct minimum location. For the shared objectives, we employ standard benchmark functions commonly used in BO, namely the Ackley, Branin, and Hartmann functions [10].

Table 2: Homogeneous Task Benchmark Specification

Shared Objective	Local Transformation	Input Dimension
Ackley Function	Shift	1D, 5D, 10D
Branin Function	Shift	2D
Hartmann Function	Shift	6D
Combination	Linear Combination	1D, 5D, 10D
Controller Optimization	Environment Factors	5D

Linear Combination As motivated in Section 2, some MTO problems can be modeled as a linear combination of shared components, weighted by user-specific preferences. Therefore, we design a benchmark problem to reflect this relationship by defining each agent's objective as a unique combination of five base benchmark functions. The objective function for agent k is defined as:

$$\text{Combination}^{(k)}(\mathbf{x}) = \mathbf{a}^{(k)} \cdot \begin{bmatrix} \text{Ackley}(x) \\ \text{Cosine}(x+1) \\ \text{Rastrigin}(x-8) \\ \text{Griewank}(x+4) \\ \text{Sphere}(x-3) \end{bmatrix} \tag{6}$$

where $\mathbf{a}^{(k)}$ is the agent-specific row vector of coefficients. The coefficients in $\mathbf{a}^{(k)}$ are randomized for each agent. This ensures that each agent has a differently shaped objective function, modeling unique preferences. Additionally, we apply fixed shifts to the inputs of the base functions. This prevents trivial alignment of the functions and allows for different optimal locations to emerge depending on the agent's coefficient vector. The output of each base benchmark function is scaled to the range $[0, 1]$. This normalization ensures that all components contribute equitably to the final combination, regardless of their original output scales.

Controller Optimization The final type of MTO problem we include is an application scenario, where we tune controllers operating under different environmental conditions. Specifically, each client tunes a Linear Quadratic Regulator (LQR) controller for the classical cart-pole problem [2]. The cart-pole system consists of a pole attached to a cart that moves along a track. The objective is to stabilize the system by applying horizontal forces to the cart while keeping the pole upright. The system's dynamics are nonlinear and are typically described by a set of coupled differential equations involving the cart's position and velocity, as well as the pole's angle and angular velocity.

To construct the controller, a linearized model around the upright equilibrium is used, from which an LQR feedback law is obtained that minimizes

$$J = \int_0^\infty (\mathbf{x}^T(t) Q \mathbf{x}(t) + \mathbf{u}^T(t) R \mathbf{u}(t))\, dt \tag{7}$$

where $\mathbf{x}(t)$ denotes the system state, $\mathbf{u}(t)$ the control input, and Q and R are positive semi-definite weighting matrices that balance the trade-off between state error and control effort. Rather than directly optimizing the quadratic LQR cost, each agent treats the weighting matrices Q and R as tunable parameters that define a feedback controller through the solution of an algebraic Riccati equation as in Marco et al. [25]. In our setup, Q is restricted to be diagonal and R is scalar, so that each agent's parameter vector consists of the four diagonal entries of Q and the single entry of R.

We randomize the environmental factors for each agent, including parameters such as the mass of the cart or the length of the pole. This variation causes the behavior of the systems to differ, while their underlying cart-pole dynamics remain the same.

The objective function $f^{(k)}$ for agent k is defined as the negative average number of time steps for which the pole remains upright and within a certain range of the track. For a given set of Q and R parameters, we simulate the agent's specific cart-pole system for ten episodes, each running for a maximum of 500 time steps. The function value is the negative average number of steps the pole remained stable across these episodes. This formulation turns the stabilization task into a minimization problem.

Table 3: Heterogeneous Task Benchmark Specification based on [7]

Category	Task	Degree of intersection	R_s
CI+HS	Griewank (T_1) Rastrigin (T_2)	Complete intersection	1.000
CI+MS	Ackley (T_3) Rastrigin (T_4)	Complete intersection	0.2261
CI+LS	Ackley (T_5) Schwefel (T_6)	Complete intersection	0.0002
PI+HS	Rastrigin (T_7) Sphere (T_8)	Partial intersection	0.8670
PI+MS	Ackley (T_9) Rosenbrock (T_{10})	Partial intersection	0.2154
PI+LS	Ackley (T_{11}) Weierstrass (T_{12})	Partial intersection	0.0725
NI+HS	Rosenbrock (T_{13}) Rastrigin (T_{14})	No intersection	0.9434
NI+MS	Griewank (T_{15}) Weierstrass (T_{16})	No intersection	0.3669
NI+LS	Rastrigin (T_{17}) Schwefel (T_{18})	No intersection	0.0016

C Hyperparameter

Unless otherwise specified, we use a fully connected backbone with 4 hidden layers and 64 neurons per layer (Table 4). For the model head, we employ a single output layer that depends on the chosen type of neural network, with ReLU as the activation function for all neurons. The VBLL models use the variational lower bound for regression defined in Equation (5). For the GP baseline, we employ a Matérn kernel as well as TS as an acquisition function using a candidate set as described in Eriksson et al. [10, Appendix E].

Moreover, similar to Brunzema et al. [3, Appendix C.1] this appendix also evaluates a second variant of our approach that employs MLP networks as surrogates instead of a VBLL, i.e., it learns the Thompson sample. We refer to this variant as GLOBAFED (MLP) for the FedBO implementation, L-MLP for the agent-local implementation and C-MLP for the central variant (cf. Section E). These MLP-based surrogates are trained using a Mean Squared Error loss.

Table 4: Hyperparameter of neural network surrogate models.

Approach	**Local LR**	**Agg. LR**	**Features**	**Backbone Layers**	**Head Layers**
FL-MLP	1e-4	8e-3	64	4	1
FL-VBLL	1e-3	8e-3	64	4	1
L-MLP	1e-4	/	64	/	5
L-VBLL	1e-4	/	64	/	5

D Extended Results

Figure 7 shows the comparison of GLOBAFED with other FedBO baselines for all objective functions. Figure 8 shows the comparison with agent-local baselines for all objective functions. The effects discussed in Section 5.2, namely similar behavior of all approaches except IAF-FBO and FD-EMD for low dimensions and significantly faster optimization via GLOBAFED than baselines for higher dimension, remain consistent among all objectives.

Table 5: VBLL Hyperparameters.

Approach	Prior Scale	Wishart Scale	Degrees of Freedom
FL-VBLL	10.0	1.0	25.0
L-VBLL	10.0	0.01	1.0
C-VBLL	10.0	1.0	25.0
FL-MLH-VBLL	10.0	1.0	25.0

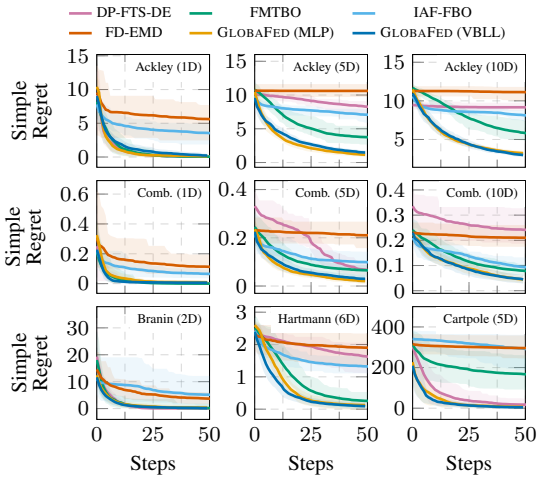


Figure 7: Performance of FedBO baselines in comparison to the VBLL and MLP variants of GLOBAFED. *All approaches perform similar in low dimensions, except IAF-FBO and FD-EMD, which do not converge to the optimum.* GLOBAFED *converges much faster and to lower final regrets for high-dimensional problems than all FedBO baselines.*

Both the MLP and VBLL variants of GLOBAFED perform similarly. We believe this parity is mainly due to two aspects. First, we utilize Thompson sampling in this pipeline, which we employed specifically to maintain consistency with the original VBLL paper's method [3]. The authors of [3] noted a slight advantage for VBLLs over standard MLPs. And second, we hypothesize that training for more epochs with early stopping could further improve VBLL performance as demonstrated in prior work on VBLL surrogate models [3].

Only the NN-based agent-local approaches perform poorly for low-dimensional problems. We attribute the poor performance to suboptimal initialization that can trap the MLP surrogate, causing repeated sampling at the search-space boundaries.[3] Our federated approach mitigates this issue: by sharing the model structure across agents, information from other clients can correct poorly initialized or stuck surrogates, resulting in more robust and stable optimization behavior.

E Training of Central Baselines

To examine whether our privacy-preserving knowledge transfer incurs any performance loss compared to centralized training with full data access, we designed a centralized version of our approach. In this setting, a shared backbone is optimized using a global loss defined over the data of all agents. Let (X_i, Y_i) be the batch of input and target data for agent i, and $\mathcal{L}$ be the loss function. We define the global loss as:

$$\mathcal{L}_{\text{global}} = \frac{1}{K} \sum_{k=1}^{K} \mathcal{L}(X_i, Y_i, (\varphi, \theta_i)). \tag{8}$$

Training follows the same procedure as the federated surrogate training illustrated in Section 4.1, except that the backbone is trained jointly using the global loss instead of via FL. Note that especially

[3] Note that the architecture obtained from the 10D Ackley task differs from the one used in Brunzema et al. [3].

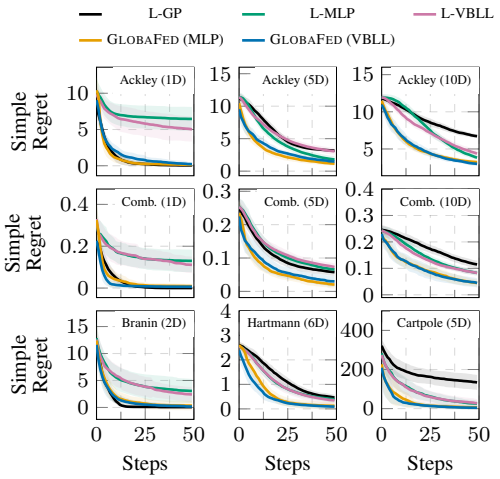


Figure 8: GLOBAFED versus other agent-local approaches (extended results). *Agent-local approaches are labeled with an "L-". The rest names the surrogate model (GP, VBLL or MLP)*

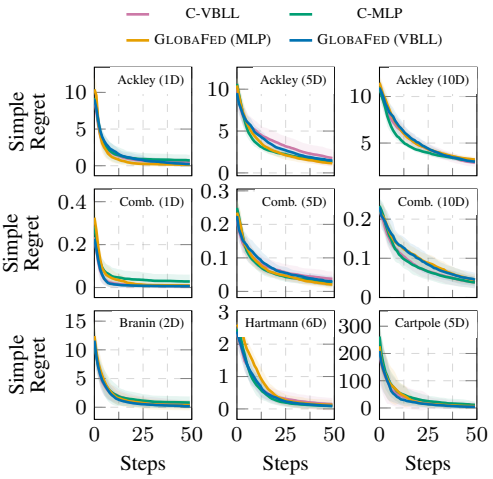


Figure 9: GLOBAFED versus its central variant. GLOBAFED *shows similar performance to its centrally trained variant, demonstrating that the privacy-preserving federated training enables efficient knowledge transfer.*

for the VBLL baseline, formulating a joint optimization objective over multiple heads is non-trivial as they may converge at different rates and it is an interesting direction for future work to prevent overfitting of individual heads. However, the results in Table 1 show that a simple aggregation as in (8) already yields good performance.

Figure 9 compares our approach with its centralized variants to assess whether knowledge transfer via model aggregation is as effective as training a centralized backbone on the raw data of all agents. Our federated method matches the performance of the centralized approach, with small problem-dependent differences where either variant may perform slightly better. We therefore conclude that the privacy-preserving nature of our method does not degrade performance. These results suggest that shared feature representations, whether learned centrally or federated, provide a powerful and robust mechanism for knowledge transfer across agents.

F Ablations

F.1 Number of Agents

Figure 10 presents the results of our study regarding the number of agents for all objectives functions showing consistent behavior across all functions, i.e., more agents improve convergence and reduce variance especially for higher-dimensional problems.

F.2 Number of Communication Rounds

For our experiments, we use a high number of communication rounds R per optimization step to ensure optimal performance. However, for real-world applications, this introduces high communication costs and synchronization overhead. Therefore, we tested how reducing the number of communication rounds while keeping the overall number of training epochs the same affects performance. Specifically, we evaluated our approach for $R = 25$, $R = 50$, and $R = 100$ communication rounds.

As Figure 11 illustrates, the highest number of rounds achieves the best overall performance. A possible explanation for this behavior is that more frequent communication enables more direct and consistent knowledge transfer, thereby improving the quality of the surrogate model approximation. Furthermore, each communication round includes an aggregation step, which serves as an additional update step during training. Configurations with fewer rounds inherently perform this update step less often. The performance impact of reducing communication rounds is less prominent for simpler problems. While fewer rounds still achieve comparable performance on the linear combination and

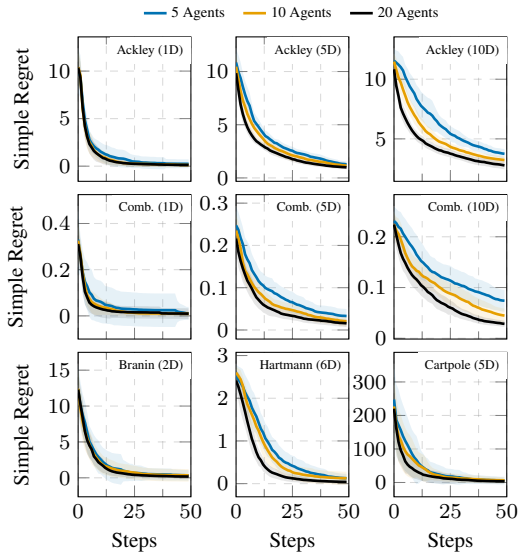


Figure 10: GLOBAFED versus number of agents (extended results).

controller optimization tasks, they result in significantly worse performance and higher variance on the shifted benchmark problems. This suggests that the shifted problems are inherently more complex to model with our surrogate, and therefore require more frequent communication to learn a sufficiently expressive shared feature representation.

We want to note that our method does incur higher communication costs than the considered baselines. Existing approaches typically require only one or a small number of model exchanges per newly evaluated objective point, whereas our decentralized knowledge-sharing mechanism may involve hundreds of communication rounds.

This additional overhead is a deliberate trade-off: more frequent communication enables agents to continuously refine the shared representation and thereby improve sample efficiency without requiring additional, potentially expensive objective evaluations. We therefore expect our method to be most attractive in settings where objective evaluations are substantially more costly than communication.

F.3 Feature Dimensionality

The number of features per layer determines the total number of parameters each agent must transmit, thereby directly affecting the size of the messages exchanged during communication. Consequently, it has a direct impact on the communication cost of the entire knowledge transfer process.

As shown in Figure 12, the optimal number of features depends on the complexity of the underlying problem. While a small number of features is sufficient to model low-dimensional problems, more complex problems require larger feature representations to capture intricate relationships between tasks. Reducing the number of features to 16 per layer already incurs a loss in performance for the 2-dimensional shifted Branin benchmark. As the dimension of the problem increases, so does the loss of performance for using a low number of features.

A higher number of features per layer can directly improve the performance of our approach, but also increases its communication costs. However, the performance improvement might only be minor if the complexity of the problem does not increase in the same way. Consequently, it is essential to strike a balance between model expressiveness and communication efficiency. As a general guideline, we suggest tuning the surrogate model on a fast-to-evaluate test problem of similar dimensionality to the real application to determine optimal parameters.

F.4 Backbone Depth

In combination with the number of hidden features, the backbone depth determines the message size of the federated surrogate training. Contrary to previous results, where increasing a hyperparameter, such as the number of rounds or features, usually improved performance, Figure 13 shows that this does not necessarily hold for the number of backbone layers. In our experiments, a backbone with only two layers performed the best overall.

This observation may be closely related to the amount of local data available. Since agents optimize objective functions that are expensive to evaluate, their local datasets are small. While deeper backbones can theoretically capture more complex relationships, they also require more data to train effectively. In this low-data context, the smallest backbone appears to produce better shared features, suggesting more stable training. Smaller models are also less prone to overfitting and can learn more robust features from limited data. While a network that is too shallow may lack the expressive power to model complex relationships, our findings indicate that a small backbone depth was sufficient for the specific relations we simulated, even for high-dimensional problems.

F.5 Head Depth

The number of head layers does not influence the communication cost of our approach, since the head parameters are not transmitted. However, it remains an important hyperparameter to investigate, as it determines how efficiently each agent can transform and adapt the shared backbone features to its local problem.

A single output layer includes no activation functions and can therefore only combine the shared features as a linear combination. Our previous results on the high-dimensional heterogeneous task benchmark already indicated that more complex adaptation is especially important for heterogeneous tasks that may have only limited similarities. The effect of the number of head layers is presented in Figure 14.

For low-dimensional problems, a single output layer is typically sufficient and can even be advantageous. While adding one additional head layer performs comparably in these cases, introducing too many layers can degrade performance and increase variance, as observed for configurations with three head layers. A possible explanation for this behavior could be that introducing additional layers reduces the direct influence of the shared features and can result in overfitting on the local task. As we have illustrated in Figure 4, shared knowledge is vital for neural network surrogates in low dimensions to prevent the agent from becoming stuck in sampling the boundaries of the search space. In contrast, for high-dimensional problems, adding more head layers proves beneficial, as it allows for a more flexible adaptation of the shared features to the increased problem complexity.

Interestingly, across all benchmark problems, models with a single output layer often achieve better performance during the early iterations. This may be because only linearly combining the shared features allows them to exert a stronger influence on the predictions. This increased reliance on shared knowledge might, in turn, encourage exploration using the shared features, which proves to be advantageous in early iterations. As a consequence, it might be beneficial to employ an adaptive partitioning of the surrogate model for optimal performance. Early iterations could strongly leverage shared knowledge by employing a single output layer, while later iterations could switch to a complex head to adapt to the local problem as more local observations become available.

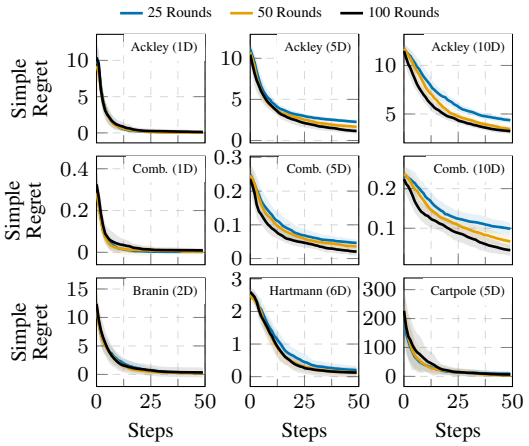


Figure 11: Influence of Number of Communication Rounds. *Optimization performance increases with the number of communication rounds per iteration.*

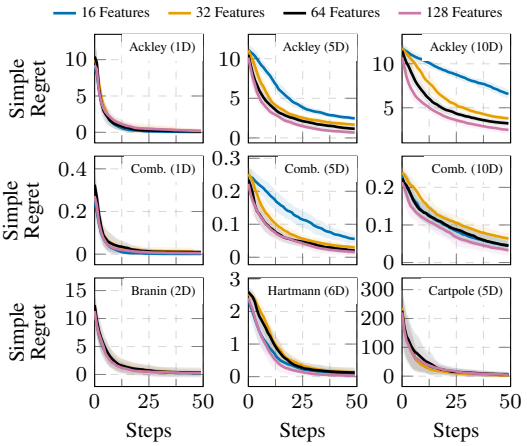


Figure 12: Influence of Feature Dimension. *The optimal number of features coincides with the complexity of the problem.*

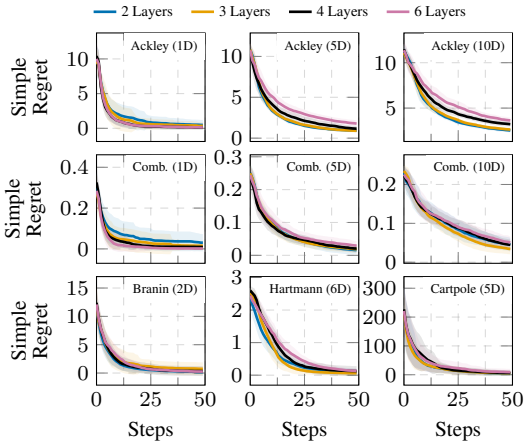


Figure 13: Influence of Backbone Depth. *For the objective functions shown, the optimal number of backbone layers is two. The reason for this is that more layers tend to overfit quickly in the low-data regime of* GLOBAFED.

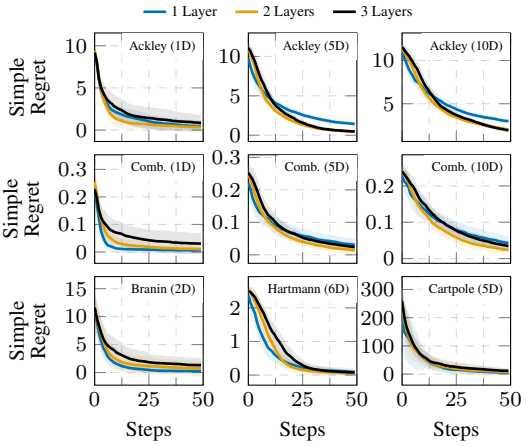


Figure 14: Influence of Head Depth. *The optimal performance is reached at around two layers. More layers degrade performance as the surrogate starts overfitting on the local objective and does not learn a common feature representation via the backbone.*

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it

(after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our abstract's and introduction's main claim, which states that GLOBAFED's knowledge exchange via NN surrogates provides improved optimization performance, is empirically demonstrated in Section 5.2.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of GLOBAFED in Section 6.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be

used reliably to provide closed captions for online lectures because it fails to handle technical jargon.

- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification:

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All experimental details needed to reproduce our results can be found in Sections B, C and 5.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example

(a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Please refer to the supplementary material.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All hyperparameter can be found in Section C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: All experiments were repeated ten times with different random seeds. Please refer to the uncertainty intervals provided in our plots like, e.g., Figure 3.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: Our method is lightweight enough to run locally on a desktop computer.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: This work advances the field of black-box optimization, a general-purpose methodology with broad applications across numerous domains. While black-box optimization enables beneficial uses such as optimizing medical devices, drug discovery and clean energy systems, it could also potentially be applied to harmful purposes. Given the fundamental and general nature of black-box optimization techniques, we do not believe this work raises specific ethical concerns beyond those inherent to the broader field of optimization research.

   Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: Please refer to our answer regarding the code of ethics.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We use the many-task benchmark, which we cited in Section 5.1.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification:

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification:

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification:

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification:

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.