Supplementary Material
Model Fusion for Personalized Learning

A. Additional Experiment Results

This section provides extra experiment results and visualization plots that could not be included in the main text due to limited space. Additional information regarding our experiment setup is also included for better clarity. Our experimental code is also released at the GitHub repository below:

https://github.com/zevergreenz/model_fusion_for_personalized_learning

A.1. Sine Prediction Experiments

As mentioned briefly in the main text, the personalized performance of MAML (Finn et al., 2017) in problem domains with highly polarized task distributions deteriorates and varies substantially across different sub-domains of tasks. This can be observed in Table 5 below which tabulates the reported performance of all baseline methods across different sub-domains and few-shot settings. In Table 4 and Table 5, we have also extended the benchmark to include reported performance for the latest personalized federated learning work (PerFedAvg) of Fallah et al. (2020). The extended benchmark shown that on average over the entire (polarized) sine domain, PerFedAvg achieves significantly better performance than MAML (Finn et al., 2017) but still incurs higher normalized RMSE than the variant of our model (with embedding alignment). The performance difference is again most pronouncing in the 0-shot setting when there is no data for further fine-tuning. On a closer look, it appears that PerFedAvg slightly performs better than ours in sub-domain 1 under the richer-data settings of 20-shot and 40-shot but in the remaining scenarios across the remaining 4 sub-domains, ours is substantially better than PerFedAvg. This agrees with and supports our above observation that on average, PerFedAvg performs better than MAML but still noticeably under-perform when compared to our model. For more visualized performance plot showcasing the fitting of various m-shot personalized neural net on some (unseen) sine functions, see Appendix F.

	0-SHOT	10-SHOT	20-SHOT	30-SHOT	40-SHOT
OURS [1-100-1]	$\mathbf{0.135 \pm 0.09}$	$\mathbf{0.114 \pm 0.03}$	$\mathbf{0.029 \pm 0.01}$	$\mathbf{0.021 \pm 0.01}$	$\mathbf{0.017 \pm 0.01}$
OURS [1-100-1] (NO ALIGN)	0.357 ± 0.01	0.125 ± 0.01	0.043 ± 0.01	0.036 ± 0.01	0.028 ± 0.01
COLD-START [1-100-1]	0.445 ± 0.26	0.149 ± 0.03	0.085 ± 0.05	0.076 ± 0.05	0.052 ± 0.04
PERFEDAVG [1-100-1]	0.356 ± 0.02	0.146 ± 0.03	0.046 ± 0.03	0.045 ± 0.04	0.032 ± 0.03
MAML [1-100-1]	0.446 ± 0.12	0.281 ± 0.18	0.161 ± 0.07	0.149 ± 0.06	0.142 ± 0.06
MAML [1-40-40-1]	$\mathbf{0.258 \pm 0.11}$	$\mathbf{0.022 \pm 0.02}$	$\mathbf{0.014 \pm 0.01}$	$\mathbf{0.014 \pm 0.01}$	$\mathbf{0.012 \pm 0.01}$

Table 4. Reported performance of our method (with and without embedding alignment), cold-start, PerFedAvg (Fallah et al., 2020) and MAML (Finn et al., 2017) on the sine synthetic dataset. The results is averaged over all domains and independent runs.

To further demonstrate the necessity of incorporating an embedding alignment component while learning to embed the existing pre-trained neural nets, we provide below in Fig. 3 another set of visual excerpts demonstrating the defection of the no-alignment variant of our method in zero-shot scenario. These plots in particular demonstrate a few scenarios where the no-alignment version fails to capture the generic trend of the target sine function. In contrast, all plots show that our version with embedding alignment is able to capture the trend and fit the target sine function much better, which highlights the importance of having the embedding alignment in Section 3.3.

A.2. Meta-Model Parameterization Specification

Due to limited space, we only describe the neural parameterization of many components of our meta model representation in high-level ideas. Here, we provide a more comprehensive specification of those neural parameterizations to improve the clarity of our work.

Neural Parameterization α for the Base Module. This module comprises two network segments which are responsible for computing the mean vector and diagonal covariance matrix of the outer multivariate Gaussian that distributes $\mathbf{w}_\varnothing$. In our experiment, $\mathbf{w}_\varnothing$ is a 100-dim vector. This is a deep generative prior which probabilistically generates $\mathbf{w}_\varnothing$ from a noise vector. In particular, a noise vector of 100-dim is passed through a dense layer comprising 100 hidden neurons with no

	SUB-DOMAIN 1: $a \sim \mathrm{U}[4,5]$, $b = 2.94$			SUB-DOMAIN 2: $a \sim \mathrm{U}[3,4]$, $b = 2.44$		
	10-SHOT	20-SHOT	40-SHOT	10-SHOT	20-SHOT	40-SHOT
OURS [1-100-1]	$\mathbf{0.16 \pm 0.01}$	0.06 ± 0.01	0.04 ± 0.01	$\mathbf{0.12 \pm 0.01}$	$\mathbf{0.02 \pm 0.01}$	$\mathbf{0.01 \pm 0.01}$
COLD-START [1-100-1]	0.31 ± 0.01	0.15 ± 0.01	0.08 ± 0.01	0.26 ± 0.01	0.19 ± 0.01	0.12 ± 0.01
PERFEDAVG [1-100-1]	$\mathbf{0.16 \pm 0.05}$	$\mathbf{0.02 \pm 0.01}$	$\mathbf{0.02 \pm 0.01}$	0.14 ± 0.01	0.03 ± 0.01	$\mathbf{0.01 \pm 0.01}$
MAML [1-100-1]	0.28 ± 0.25	0.12 ± 0.02	0.11 ± 0.03	0.25 ± 0.18	0.16 ± 0.05	0.13 ± 0.02
	SUB-DOMAIN 3: $a \sim \mathrm{U}[2,3]$, $b = 1.50$			SUB-DOMAIN 4: $a \sim \mathrm{U}[1,2]$, $b = 0.91$		
	10-SHOT	20-SHOT	40-SHOT	10-SHOT	20-SHOT	40-SHOT
OURS [1-100-1]	$\mathbf{0.06 \pm 0.01}$	$\mathbf{0.02 \pm 0.01}$	$\mathbf{0.01 \pm 0.01}$	$\mathbf{0.11 \pm 0.01}$	$\mathbf{0.02 \pm 0.01}$	$\mathbf{0.01 \pm 0.01}$
COLD-START [1-100-1]	0.27 ± 0.01	0.21 ± 0.01	0.11 ± 0.01	0.23 ± 0.01	0.13 ± 0.01	0.11 ± 0.01
PERFEDAVG [1-100-1]	0.12 ± 0.01	0.03 ± 0.01	$\mathbf{0.01 \pm 0.01}$	0.14 ± 0.01	0.08 ± 0.04	0.05 ± 0.03
MAML [1-100-1]	0.31 ± 0.14	0.16 ± 0.08	0.15 ± 0.07	0.24 ± 0.10	0.16 ± 0.04	0.16 ± 0.04
	SUB-DOMAIN 5: $a \sim \mathrm{U}[0,1]$, $b = 0.91$					
	10-SHOT	20-SHOT	40-SHOT			
OURS [1-100-1]	$\mathbf{0.13 \pm 0.22}$	0.03 ± 0.01	$\mathbf{0.01 \pm 0.01}$			
COLD-START [1-100-1]	0.16 ± 0.03	$\mathbf{0.02 \pm 0.01}$	0.02 ± 0.01			
PERFEDAVG [1-100-1]	0.18 ± 0.01	0.09 ± 0.01	0.08 ± 0.01			
MAML [1-100-1]	0.23 ± 0.11	0.09 ± 0.05	0.07 ± 0.01			

Table 5. Reported (average) normalized RMSE of personalized neural nets with architecture [1-100-1] generated by our method (with alignment), cold-start, PerFedAvg (Fallah et al., 2020) and MAML (Finn et al., 2017) on each sub-domain. The (unseen) sine functions in each sub-domain share the same phase b but have different magnitudes a, which was sampled uniformly from polarized value ranges.

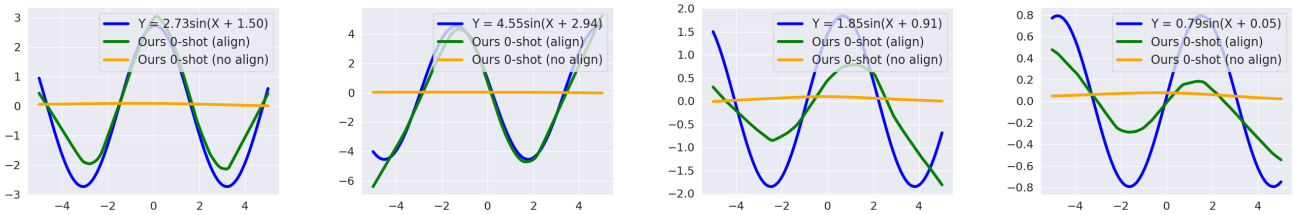


Figure 3. Visual excerpts demonstrating a few particular 0-shot scenarios where the no-alignment variant of our method generates poor/uninformed personalized neural nets. In contrast, the alignment version of our method is able to capture well the trends of the target sine functions in the same 0-shot scenarios. This highlights the importance of having an embedding alignment component when there is no learning example of the target task (i.e., 0-shot personalization) for the meta model to recognize the embedding mis-alignment via data.

activation, which produces the output of the first segment. For the second segment, the noise vector is passed through another dense layer comprising of 100 hidden neurons with tanh activation. The output is further normalized with a batch-norm layer, which finally produces the output of the second segment.

GP-modulated Neural Parameterization ϕ of the Task-Specific Module. This module comprises a collection of $e = 10$ independent sparse localized Gaussian processes (GPs) (Snelson, 2007) which represents the e (independent) priors over e random functions mapping from the task's meta-data vector to a scalar value. The collective output of this e GPs is therefore an e-dim vector of independently distributed components, which is then warped into a 100-dim space of the task-specific component $\mathbf{w}$. The warper is designed to be a 2-layer feed-forward neural net. Its first layer comprises 256 neurons with tanh activation while its second layer comprises 100 neurons with no activation. Intuitively, this means the role of the GPs is to produce independent prediction signals which are weaved together via the 2-layer feed-forward net to generate a 100-dim task specific vector with highly correlated components. For each GP prior, the parameters comprises its kernel parameters as well as its inducing input vector. In our experiment, we use 50 inducing input vectors per GP where each inducing input is a $\dim(\boldsymbol{\tau})$-dim vector whose coordinates need to be learned.

Neural Parameterization ν for the Crossing Module $p(\boldsymbol{\gamma}|\mathbf{w}_\varnothing, \mathbf{w})$. This module comprises two network segments which are responsible for producing the mean vector and diagonal covariance matrix of the outer multivariate Gaussian that distributes $\boldsymbol{\gamma}$. The two segments share a common sub-segment that maps the concatenated vector pf $\mathbf{w}_\varnothing$ and $\mathbf{w}$ to a latent space. In our experiment setting, both $\mathbf{w}_\varnothing$ and $\mathbf{w}$ are 100-dim vector. Their concatenation vector is first fed into a dense

layer of 64 hidden neurons with ReLU activation. The output of this layer is passed through another layer of $d = 32$ hidden neurons with ReLU activation, which produces the 32-dim latent embedding for the concatenation of $\mathbf{w}_\varnothing$ and $\mathbf{w}$. Next, to produce the mean vector of $\boldsymbol{\gamma}$, the latent embedding is passed through a dense layer comprising $|\boldsymbol{\gamma}|$ hidden neurons with no activation, which constitutes the first network segment. As for the other, the same latent embedding (i.e., the output of the common sub-segment) is passed through a separate dense layer comprising $\dim(\boldsymbol{\gamma}) = 100$ hidden neurons with tanh activation, which generates the output of the second network segment.

Note the the above parameterization of ν describes the processing of generative process of $\boldsymbol{\gamma}$ from $\mathbf{w}$ and $\mathbf{w}_\varnothing$ for a single task instance $\boldsymbol{\tau}$. However, as we pointed out in the main text, pre-trained models $\{\boldsymbol{\gamma}_i\}_{i=1}^n$ were generated in isolated and hence, their induced model vector $\{\boldsymbol{\gamma}_i\}_{i=1}^n$ do not necessarily align component by component (e.g., component 1 of $\boldsymbol{\gamma}_i$ might correspond to component 3 of $\boldsymbol{\gamma}_j$) and we need to learn a matching permutation to align their components. To this end, we aim to learn for each model $\boldsymbol{\gamma}_i$ a separate permutation that maps it to a universal canonical order of components. These permutations $\{\mathbf{A}_i\}_{i=1}^n$ are applied on the above 32-dim latent space of the embedding vector of the concatenation of $\mathbf{w}$ and $\mathbf{w}_\varnothing$ (i.e., the output of the common sub-segment described above). In our meta-model, the permutation matrices $\{\mathbf{A}_i\}_{i=1}^n$ are treated as the discrete part of ν, and are optimized alternatingly with the rest as detailed previously in Section 3.2.

A.3. Computational Complexity

In addition, we provide below a concise analysis of the computational complexity of our meta model training. Plots showing its training losses against the no. of training iterations are also provided to demonstrate its convergence empirically.

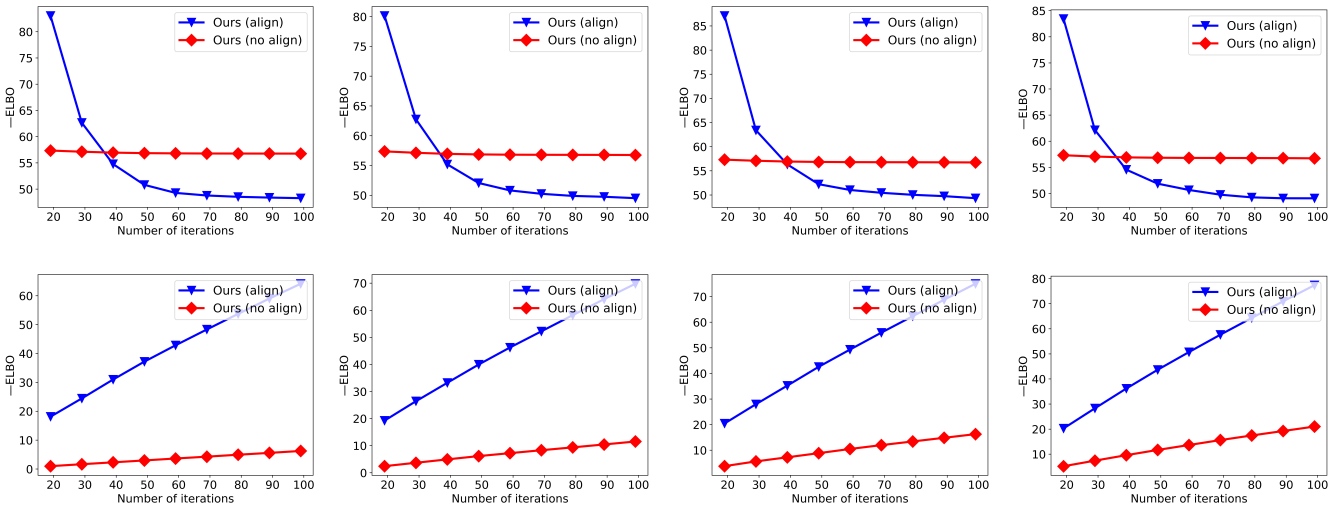
First, to understand the computation complexity of our meta-model, we first note that it comprises the 3 deep generative modules as detailed above. Two of which (the base and crossing modules) are parametric networks in nature so the cost of their feed-forward and back-propagation computation is a linear function of their parameters, input dimension and batch sizes. As both are deep generative models, their parameters are optimized numerically via re-parameterized sampling (Kingma & Welling, 2013) which incurs another linear factor of the sampling size h on their feed-forward and back-propagation complexities (since we need to repeat the same computation routine for each sample).

Second, the other component (the task-specific module) on the other hand is non-parametric due to its modulation with a sparse GP prior layer whose feed-forward (prediction) scale linearly in the total number n of observed tasks (those with pre-trained models provided) and the dimension $|\boldsymbol{\tau}|$ of their corresponding meta-data vectors. Each sparse GP also scales quadratically in the number m of inducing inputs. For back-propagation however, it scales quadratically in both the number m of inducing inputs and the meta-data dimension $\boldsymbol{\tau}$ while remaining linear in n.

Furthermore, this module is also generative in nature which means both its forward and backward computation also incur a routine of forward sampling from the posterior GP. This includes first sampling the h samples of the inducing set comprising m vectors, which scales cubically in m since (per property of GPs) these vectors are marginally distributed by a m-dimensional Gaussian with full-rank covariance matrix. Thus, summarily, the total cost of feed-forward and back-propagation via a GP component incurs cubic cost in m, quadratic cost in $|\boldsymbol{\tau}|$ and linear in n and h. Also, as there are e separate GPs and there is another 2-layer feed-forward net that warps them into the space of $\boldsymbol{\gamma}$ (pre-trained model). The total processing cost of this component has to be further scaled by linear factors of e and $\boldsymbol{\gamma}$.

Thus, putting the above together, we arrive at a complexity cost of $\mathbf{O}(h \cdot |\alpha| \cdot n \cdot \dim(\mathbf{w}_\varnothing) + h \cdot |\nu| \cdot n \cdot (\dim(\mathbf{w}_\varnothing) + \dim(\mathbf{w})) + h \cdot e \cdot \dim(\boldsymbol{\gamma}) \cdot n \cdot \dim(\boldsymbol{\tau})^2 \cdot m^3)$ for each learning epoch. In our experiment, we choose $h = 100$ samples and use 100 learning epoches to optimize our meta-model. Additionally, per learning epoch, we also need to alternate between optimizing the continuous parameters and discrete permutation matrices for embedding alignment across different pre-trained models. The cost for the former has been given above while the cost for the latter boils down to the cost of solving a linear sum assignment task concerning $d \times d$ bipartite matrix (see Section 3.2). Since the weight matrix for candidate assignment can be computed and cached while optimizing the continuous parameters, the cost of solving this linear sum assignment is cubic in d, which is the latent embedding dimension of the concatenation of $\mathbf{w}_\varnothing$ and $\mathbf{w}$ in the specification of the crossing module above. It also involves a linear factor of n since we need to solve one linear sum assignment per pre-trained model so in total, the cost is $\mathbf{O}(n \cdot d^3)$ which is likely subsumed by $\mathbf{O}(|\nu| \cdot n)$ as d only accounts for a small part of the entire crossing model's parameterization $|\nu|$. As such, we can ignore it.

To further provide a empirical grasp of the above complexity analysis, the running time plots (along with corresponding plots of converging training losses) with respect to settings with $e = 5, 10, 15$ and 20 sparse GP priors are given in Fig. 4 above. All our experiments were run on an Intel(R) Xeon(R) with 32 E5-2686 v4 CPUs (2.30GHz) and a V100 GPU.



$e = 5$ (GP priors) $e = 10$ (GP priors) $e = 15$ (GP priors) $e = 20$ (GP priors)

Figure 4. Plots show (top) the negative meta-model evidence lower-bound (ELBO) being reduced as we increase the no. of training iterations. This is the negation of the objective function in Eq. (16) that we sought to maximize (hence, its negative version would decrease as expected in the above plots); and the accumulated processing time (in minute) incurred by the alignment and no-alignment versions of our method. It can be seen that the negative ELBO decreases significantly when we allow for alignment during training as compared to without alignment. As a trade-off, the version with alignment incurs more processing time (up to eight-fold on average) as compared the version without alignment. The plots are provided for different parameter configurations with $e = 5, 10, 15$ and 20 GP priors.

B. Derivation of the Meta-Model Likelihood in Eq. (8)

The full distribution of our meta-model is given by

$$p\Big(\mathbf{w}_{\varnothing}, \mathbf{w}, \boldsymbol{\gamma}, \mathbf{s}|\boldsymbol{\tau}\Big) \;\; = \;\; p\Big(\mathbf{w}_{\varnothing}\Big)p\Big(\mathbf{s}\Big)\prod_{i=1}^{n}\Big[p\Big(\boldsymbol{\gamma}_i|\mathbf{w}_{\varnothing}, \mathbf{w}_i\Big)p\Big(\mathbf{w}_i|\mathbf{s}\Big)\Big]\,. \tag{20}$$

Marginalizing out $\mathbf{w}_{\varnothing}$ and $\mathbf{w}$, we obtain the meta-model evidence in Eq. (8), Eq. (9) and Eq. (10)

$$\begin{aligned} p(\boldsymbol{\gamma}|\boldsymbol{\tau}) \;\; &= \;\; \int_{\mathbf{w}_{\varnothing},\mathbf{w},\mathbf{s}} p\Big(\mathbf{w}_{\varnothing}, \mathbf{w}, \boldsymbol{\gamma}, \mathbf{s} \mid \boldsymbol{\tau}\Big)\mathrm{d}\mathbf{s}\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{w}_{\varnothing} \\ &= \;\; \int_{\mathbf{w}_{\varnothing},\mathbf{w},\mathbf{s}} p\Big(\mathbf{w}_{\varnothing}\Big)p\Big(\mathbf{s}\Big)\prod_{i=1}^{n}\Big[p(\boldsymbol{\gamma}_i|\mathbf{w}_{\varnothing}, \mathbf{w}_i)p(\mathbf{w}_i|\mathbf{s})\Big]\mathrm{d}\mathbf{s}\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{w}_{\varnothing} \\ &= \;\; \int_{\mathbf{w}_{\varnothing}} p(\mathbf{w}_{\varnothing})\left[\int_{\mathbf{w},\mathbf{s}} p\Big(\mathbf{s}\Big)\prod_{i=1}^{n}\Big[p(\boldsymbol{\gamma}_i|\mathbf{w}_{\varnothing}, \mathbf{w}_i)p(\mathbf{w}_i|\mathbf{s})\Big]\,\mathrm{d}\mathbf{s}\mathrm{d}\mathbf{w}\right]\mathrm{d}\mathbf{w}_{\varnothing} \;\; = \;\; \mathbb{E}_{\mathbf{w}_{\varnothing}\sim p(\mathbf{w}_{\varnothing})}\Big[h\Big(\mathbf{w}_{\varnothing}\Big)\Big]\,, \end{aligned} \tag{21}$$

where we define the auxiliary function $h(\mathbf{w}_{\varnothing})$ as

$$h\Big(\mathbf{w}_{\varnothing}\Big) \;\; = \;\; \int_{\mathbf{w},\mathbf{s}} p\Big(\mathbf{s}\Big)\prod_{i=1}^{n}\Big[p\Big(\boldsymbol{\gamma}_i|\mathbf{w}_{\varnothing}, \mathbf{w}_i\Big)p\Big(\mathbf{w}_i|\mathbf{s}\Big)\Big]\mathrm{d}\mathbf{s}\mathrm{d}\mathbf{w}\,. \tag{22}$$

Now, substituting $\mathbf{w} = \mathbf{g}(\boldsymbol{\tau})$ or $\mathbf{g}$ for short

$$\begin{aligned} h\Big(\mathbf{w}_{\varnothing}\Big) &= \int_{\mathbf{g}(\boldsymbol{\tau}),\mathbf{s}} p\Big(\mathbf{s}\Big) \prod_{i=1}^{n} p\left[\Big(\mathbf{g}(\boldsymbol{\tau}_i)|\mathbf{s}\Big)\right] \prod_{i=1}^{n} \left[p\Big(\boldsymbol{\gamma}_i|\mathbf{w}_{\varnothing}, \mathbf{g}(\boldsymbol{\tau}_i)\Big)\right] \mathrm{d}\mathbf{s}\mathrm{d}\mathbf{w} \\ &= \int_{\mathbf{g}(\boldsymbol{\tau})} \underbrace{\left[\int_{\mathbf{s}} p\Big(\mathbf{s}\Big) \prod_{i=1}^{n} \left[p\Big(\mathbf{g}(\boldsymbol{\tau}_i)|\mathbf{s}\Big)\right] \mathrm{d}\mathbf{s}\right]}_{p\Big(\mathbf{g}(\boldsymbol{\tau})\Big)} \prod_{i=1}^{n} \left[p\Big(\boldsymbol{\gamma}_i|\mathbf{w}_{\varnothing}, \mathbf{g}(\boldsymbol{\tau}_i)\Big)\right] \mathrm{d}\mathbf{w} \\ &= \mathbb{E}_{\mathbf{g}\sim p(\mathbf{g})}\left[\prod_{i=1}^{n} p\Big(\boldsymbol{\gamma}_i|\mathbf{w}_{\varnothing}, \mathbf{g}(\boldsymbol{\tau}_i)\Big)\right] \text{ where } p(\mathbf{g}) = \mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})}\left[\prod_{i=1}^{n} p\Big(\mathbf{g}(\boldsymbol{\tau}_i)|\mathbf{s}\Big)\right]. \end{aligned} \tag{23}$$

C. Derivation of the Variational Inequality in Eq. (11)

From the chain rule (also called the product rule) of probability, suppose that $p(\mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}|\boldsymbol{\gamma}, \boldsymbol{\tau}) \neq 0$, we have

$$p(\boldsymbol{\gamma}|\boldsymbol{\tau}) = \frac{p(\boldsymbol{\gamma}, \mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}|\boldsymbol{\tau})}{p(\mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}|\boldsymbol{\gamma}, \boldsymbol{\tau})}. \tag{24}$$

Assuming that $p(\boldsymbol{\gamma}|\boldsymbol{\tau}) \neq 0$ (consequently, $p(\boldsymbol{\gamma}, \mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}|\boldsymbol{\tau}) \neq 0$), taking $\log$ on both sides of the above equality

$$\log p(\boldsymbol{\gamma}|\boldsymbol{\tau}) = \log\left(\frac{p(\boldsymbol{\gamma}, \mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}|\boldsymbol{\tau})}{p(\mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}|\boldsymbol{\gamma}, \boldsymbol{\tau})}\right). \tag{25}$$

For any distribution $q(\mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}; \omega)$ parameterized by ω such that $q(\mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}; \omega) \ll p(\mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}|\boldsymbol{\gamma}, \boldsymbol{\tau})$, i.e., $q(\mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}; \omega)$ is absolutely continuous with respect to $p(\mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}|\boldsymbol{\gamma}, \boldsymbol{\tau})$, we have

$$\mathbb{E}_{q(\mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}; \omega)}\Big[\log p(\boldsymbol{\gamma}|\boldsymbol{\tau})\Big] = \mathbb{E}_{q(\mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}; \omega)}\left[\log\left(\frac{p(\boldsymbol{\gamma}, \mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}|\boldsymbol{\tau})}{p(\mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}|\boldsymbol{\gamma}, \boldsymbol{\tau})}\right)\right]. \tag{26}$$

Since $p(\boldsymbol{\gamma}|\boldsymbol{\tau})$ is a constant with respect to $q(\mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}; \omega)$, we can pull that out of the expectation

$$\begin{aligned} \log p(\boldsymbol{\gamma}|\boldsymbol{\tau}) &= \mathbb{E}_{q(\mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}; \omega)}\left[\log\left(\frac{p(\boldsymbol{\gamma}, \mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}|\boldsymbol{\tau})}{p(\mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}|\boldsymbol{\gamma}, \boldsymbol{\tau})}\right)\right] \\ &= \mathbb{E}_{q(\mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}; \omega)}\left[\log\left(\frac{p(\boldsymbol{\gamma}, \mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}|\boldsymbol{\tau}) q(\mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}; \omega)}{p(\mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}|\boldsymbol{\gamma}, \boldsymbol{\tau}) q(\mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}; \omega)}\right)\right] \\ &= \mathbb{E}_{q(\mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}; \omega)}\left[\log\left(\frac{p(\boldsymbol{\gamma}, \mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}|\boldsymbol{\tau})}{q(\mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}; \omega)}\right)\right] + \mathbb{E}_{q(\mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}; \omega)}\left[\log\left(\frac{q(\mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}; \omega)}{p(\mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}|\boldsymbol{\gamma}, \boldsymbol{\tau})}\right)\right] \\ &= \mathbb{E}_{q(\mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}; \omega)}\left[\log\left(\frac{p(\boldsymbol{\gamma}, \mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}|\boldsymbol{\tau})}{q(\mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}; \omega)}\right)\right] + \mathbb{D}_{\mathrm{KL}}\left[q\Big(\mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}; \omega\Big) \Big\| p\Big(\mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}|\boldsymbol{\gamma}, \boldsymbol{\tau}\Big)\right] \\ &\geq \mathbb{E}_{q(\mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}; \omega)}\left[\log\left(\frac{p(\boldsymbol{\gamma}, \mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}|\boldsymbol{\tau})}{q(\mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}; \omega)}\right)\right] \\ &= \mathbb{E}_{q(\mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}; \omega)}\left[\log\Big(p\Big(\boldsymbol{\gamma}, \mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}|\boldsymbol{\tau}\Big)\Big)\right] - \mathbb{E}_{q(\mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}; \omega)}\left[\log\Big(q\Big(\mathbf{w}_{\varnothing}, \mathbf{w}, \mathbf{s}; \omega\Big)\Big)\right], \end{aligned} \tag{27}$$

where the second-last inequality is due to the non-negativity of the KL divergence.

D. Derivation of Eq. (18)

From Eq. (17), the probability $p(\boldsymbol{\gamma}_i|\mathbf{w}_i, \mathbf{w}_{\varnothing}; \nu)$ is given as

$$p(\boldsymbol{\gamma}_i|\mathbf{w}_i, \mathbf{w}_{\varnothing}; \nu) = \prod_{u=1}^{d} \mathbf{N}\left([\boldsymbol{\gamma}_i]_u \,\Big|\, \sum_{v=1}^{d} \mathbf{A}_i^{uv} \left[\mathbf{m}_{\lambda}^{i}\right]_v, \sum_{v=1}^{d} \mathbf{A}_i^{uv} \left[\mathbf{v}_{\lambda}^{i}\right]_v\right). \tag{28}$$

Assuming that $p(\boldsymbol{\gamma}_i|\mathbf{w}_i, \mathbf{w}_\varnothing; \nu) \neq 0$, taking log on both sides, we have

$$\log p(\boldsymbol{\gamma}_i|\mathbf{w}_i, \mathbf{w}_\varnothing; \nu) = \sum_{u=1}^{d} \log \mathbf{N}\Bigg([\boldsymbol{\gamma}_i]_u \;\Big|\; \sum_{v=1}^{d} \mathbf{A}_i^{uv} \left[\mathbf{m}_\lambda^i\right]_v, \sum_{v=1}^{d} \mathbf{A}_i^{uv} \left[\mathbf{v}_\lambda^i\right]_v \Bigg) \;. \tag{29}$$

Since $\mathbf{A}_i$ is a permutation matrix, for every row u, there exists exactly one column $v_*(u)$ – here, we make the dependence on the row u explicit – such that

$$\mathbf{A}_i^{uv} = \begin{cases} 1 & \text{if } v = v_*(u) \\ 0 & \text{otherwise} \end{cases} \;.$$

Using the above property, the following three equalities hold for any $u = 1, \ldots, d$

$$\sum_{v=1}^{d} \mathbf{A}_i^{uv} \left[\mathbf{m}_\lambda^i\right]_v = \left[\mathbf{m}_\lambda^i\right]_{v_*(u)} \;, \tag{30}$$

$$\sum_{v=1}^{d} \mathbf{A}_i^{uv} \left[\mathbf{v}_\lambda^i\right]_v = \left[\mathbf{v}_\lambda^i\right]_{v_*(u)} \;, \tag{31}$$

$$\sum_{v=1}^{d} \mathbf{A}_i^{uv} \log \mathbf{N}\Big([\boldsymbol{\gamma}_i]_u \mid [\mathbf{m}_\lambda^i]_v, [\mathbf{v}_\lambda^i]_v \Big) = \log \mathbf{N}\Big([\boldsymbol{\gamma}_i]_u \mid [\mathbf{m}_\lambda^i]_{v_*(u)}, [\mathbf{v}_\lambda^i]_{v_*(u)} \Big) \;, \tag{32}$$

Continuing from Eq. (29), we have

$$\begin{aligned} \log p(\boldsymbol{\gamma}_i|\mathbf{w}_i, \mathbf{w}_\varnothing; \nu) &= \sum_{u=1}^{d} \log \mathbf{N}\Bigg([\boldsymbol{\gamma}_i]_u \;\Big|\; \sum_{v=1}^{d} \mathbf{A}_i^{uv} \left[\mathbf{m}_\lambda^i\right]_v, \sum_{v=1}^{d} \mathbf{A}_i^{uv} \left[\mathbf{v}_\lambda^i\right]_v \Bigg) \\ &= \sum_{u=1}^{d} \log \mathbf{N}\Big([\boldsymbol{\gamma}_i]_u \;\Big|\; \left[\mathbf{m}_\lambda^i\right]_{v_*(u)}, \left[\mathbf{v}_\lambda^i\right]_{v_*(u)} \Big) = \sum_{u=1}^{d} \sum_{v=1}^{d} \mathbf{A}_i^{uv} \log \mathbf{N}\Big([\boldsymbol{\gamma}_i]_u \mid [\mathbf{m}_\lambda^i]_v, [\mathbf{v}_\lambda^i]_v \Big) \;, \end{aligned} \tag{33}$$

where the second equality is due to Eq. (30) and Eq. (31). The third equality is due to Eq. (32).

E. Theoretical Analysis

This section presents the theoretical analysis of our personalized learning framework in Section 3. The analysis comprises two parts. In **Part I**, we will derive an auxiliary result that analyzes the behavior of the distribution[7] $q(\boldsymbol{\gamma}_*|\boldsymbol{\gamma})$ of the personalized solution model $\boldsymbol{\gamma}_*$ for target task $\boldsymbol{\tau}_*$, which is derived from the optimized meta-model representation (Section 3.1). Here, the optimized representation corresponds to the universal maximizer of Eq. (16) or equivalently Eq. (19), which we sought to maximize previously in Section 3.4. This result will then be leveraged next in In **Part II** to derive a non-trivial bound on the generalized performance gap between using a personalized model $\boldsymbol{\gamma}_* \sim q(\boldsymbol{\gamma}_*|\boldsymbol{\gamma})$; and an oracle model $\boldsymbol{\gamma}_*^\circ$ which is defined to be a solution model that has best performance on $\boldsymbol{\tau}_*$, and is unknown to us.

E.1. Part I

First, recall that in part **B** of Section 3.4, the exact posterior[8] $p(\mathbf{w}_\varnothing, \mathbf{w}, \mathbf{s}|\boldsymbol{\gamma})$ of our generative model (Fig. 1) is approximated with $q(\mathbf{w}_\varnothing, \mathbf{w}, \mathbf{s}) = q(\mathbf{w}_\varnothing) p(\mathbf{w}|\mathbf{s}) \prod_{\kappa=1}^{e} p(\mathbf{s}_\kappa)$ where $q(\mathbf{w}_\varnothing)$ is parameterized separately from $p(\mathbf{s}_\kappa)$ and $p(\mathbf{w}|\mathbf{s})$ which are parts of our generative model. We can then show below (Lemma 1) that when the optimization of Eq. (16) converges to a universal maximizer, we can represent $q(\mathbf{w}_\varnothing)$ explicitly in terms of the generative model's components.

Lemma 1. *Suppose that the model evidence lower-bound* $\mathbf{L}\Big(\alpha, \phi, \nu, \omega\Big)$ *defined in Eq. (16) achieves its (global) maximum,*

[7] $\boldsymbol{\gamma}_*$ is also conditioned on the target task's meta data $\boldsymbol{\tau}_*$ as well as the related tasks' meta data $\boldsymbol{\tau} = \{\boldsymbol{\tau}_i\}_{i=1}^n$ but we omitted these terms to avoid cluttering the notations. For the rest of the analysis, conditioning on $\boldsymbol{\tau}_*$ and $\boldsymbol{\tau}$ should therefore be implicitly understood.

[8] The posterior is over the base $\mathbf{w}_\varnothing$ and specific $\mathbf{w} = \{\mathbf{w}_i\}_{i=1}^n$ components, as well as the latter's representative sets $\mathbf{s} = \{\mathbf{s}_\kappa\}_{\kappa=1}^e$ (one set per encoding dimension κ) as detailed previously in Section 3.2.

then the following is true:

$$q(\mathbf{w}_{\varnothing};\omega) \;=\; \frac{1}{\mathbf{Z}}p(\mathbf{w}_{\varnothing};\alpha)\prod_{i=1}^{n}\exp\left(\mathbb{E}_{p(\mathbf{w},\mathbf{s};\phi)}\left[\log p\Big(\boldsymbol{\gamma}_i|\mathbf{w}_i,\mathbf{w}_{\varnothing};\nu\Big)\right]\right) \text{ almost everywhere,} \tag{34}$$

where $\mathbf{Z}=\mathbb{E}_{p(\mathbf{w}_{\varnothing};\alpha)}\left[\prod_{i=1}^{n}\exp\left(\mathbb{E}_{p(\mathbf{w},\mathbf{s};\phi)}\left[\log p\Big(\boldsymbol{\gamma}_i|\mathbf{w}_i,\mathbf{w}_{\varnothing};\nu\Big)\right]\right)\right]$ *is the normalization term.*

Proof. Let us first recall the expression of Eq. (16) below:

$$\begin{aligned}
\mathbf{L}\Big(\alpha,\phi,\nu,\omega\Big) &= -\left(\mathbb{E}_{q(\mathbf{w}_{\varnothing};\omega)}\mathbb{E}_{p(\mathbf{w},\mathbf{s};\phi)}\left[-\sum_{i=1}^{n}\log p\Big(\boldsymbol{\gamma}_i|\mathbf{w}_i,\mathbf{w}_{\varnothing};\nu\Big)\right]+\mathbb{D}_{\mathrm{KL}}\Big(q(\mathbf{w}_{\varnothing};\omega)\Big\|p(\mathbf{w}_{\varnothing};\alpha)\Big)\right)\\
&= -\mathbb{E}_{q(\mathbf{w}_{\varnothing};\omega)}\left[\log\exp\left(\mathbb{E}_{p(\mathbf{w},\mathbf{s};\phi)}\left[-\sum_{i=1}^{n}\log p\Big(\boldsymbol{\gamma}_i|\mathbf{w}_i,\mathbf{w}_{\varnothing};\nu\Big)\right]\right)+\log\frac{q(\mathbf{w}_{\varnothing};\omega)}{p(\mathbf{w}_{\varnothing};\alpha)}\right]\\
&= -\mathbb{E}_{q(\mathbf{w}_{\varnothing};\omega)}\left[\log q(\mathbf{w}_{\varnothing};\omega)-\log\left(p(\mathbf{w}_{\varnothing};\alpha)\prod_{i=1}^{n}\exp\left(\mathbb{E}_{p(\mathbf{w},\mathbf{s};\phi)}\left[\log p\Big(\boldsymbol{\gamma}_i|\mathbf{w}_i,\mathbf{w}_{\varnothing};\nu\Big)\right]\right)\right)\right]\\
&= -\mathbb{D}_{\mathrm{KL}}\left(q(\mathbf{w}_{\varnothing};\omega)\;\middle\|\;\frac{1}{\mathbf{Z}}p(\mathbf{w}_{\varnothing};\alpha)\prod_{i=1}^{n}\exp\left(\mathbb{E}_{p(\mathbf{w},\mathbf{s};\phi)}\left[\log p\Big(\boldsymbol{\gamma}_i|\mathbf{w}_i,\mathbf{w}_{\varnothing};\nu\Big)\right]\right)\right)+\log\mathbf{Z}
\end{aligned} \tag{35}$$

where $\mathbf{Z}=\mathbb{E}_{p(\mathbf{w}_{\varnothing};\alpha)}\left[\prod_{i=1}^{n}\exp\left(\mathbb{E}_{p(\mathbf{w},\mathbf{s};\phi)}\left[\log p\Big(\boldsymbol{\gamma}_i|\mathbf{w}_i,\mathbf{w}_{\varnothing};\nu\Big)\right]\right)\right]$ is the normalization term, which is a constant with respect to ω. Then, due to the non-negativity property of KL divergence, any choices of ω that result in a positive divergence would not be optimal. Therefore, the model evidence lower-bound $\mathbf{L}\Big(\alpha,\phi,\nu,\omega\Big)$ achieves its (global) maximum value if and only if the KL divergence is 0, which occurs if and only if:

$$\begin{aligned}
q(\mathbf{w}_{\varnothing};\omega) &= \frac{1}{\mathbf{Z}}p(\mathbf{w}_{\varnothing};\alpha)\prod_{i=1}^{n}\exp\left(\mathbb{E}_{p(\mathbf{w},\mathbf{s};\phi)}\left[\log p\Big(\boldsymbol{\gamma}_i|\mathbf{w}_i,\mathbf{w}_{\varnothing};\nu\Big)\right]\right) \text{ almost everywhere,}\\
q(\mathbf{w}_{\varnothing};\omega) &\propto p(\mathbf{w}_{\varnothing};\alpha)\left[\prod_{i=1}^{n}\exp\left(\mathbb{E}_{p(\mathbf{w},\mathbf{s};\phi)}\Big[\log p\Big(\boldsymbol{\gamma}_i|\mathbf{w}_i,\mathbf{w}_{\varnothing};\nu\Big)\Big]\right)\right] \text{ almost everywhere,}
\end{aligned} \tag{36}$$

which completes our proof of Lemma 1. □

Leveraging this result, one can derive an immediate expression for $q(\boldsymbol{\gamma}_*|\boldsymbol{\gamma})$ as detailed next in Lemma 2 below.

Lemma 2. *Suppose that the model evidence lower-bound* $\mathbf{L}\Big(\alpha,\phi,\nu,\omega\Big)$ *defined in Eq. (16) achieves its (global) maximum. Then, its induced predictive distribution* $q(\boldsymbol{\gamma}_*|\boldsymbol{\gamma})$ *of our framework coincides with* $\pi(\boldsymbol{\gamma}_*|\boldsymbol{\gamma})$ *where*

$$\pi(\boldsymbol{\gamma}_*|\boldsymbol{\gamma}) \;=\; \frac{\mathbb{E}_{p(\mathbf{w}_{\varnothing};\alpha)}\left[\exp\left(\mathbb{E}_{p(\mathbf{w},\mathbf{s};\phi)}\Big[\sum_{i=1}^{n}\log p\Big(\boldsymbol{\gamma}_i|\mathbf{w}_i,\mathbf{w}_{\varnothing};\nu\Big)\Big]\right)\int_{\mathbf{w}_*}p\Big(\boldsymbol{\gamma}_*|\mathbf{w}_*,\mathbf{w}_{\varnothing};\nu\Big)\mathbb{E}_{p(\mathbf{w},\mathbf{s};\phi)}\Big[p(\mathbf{w}_*|\mathbf{w},\mathbf{s};\phi)\Big]\mathrm{d}\mathbf{w}_*\right]}{\mathbb{E}_{p(\mathbf{w}_{\varnothing};\alpha)}\left[\exp\left(\mathbb{E}_{p(\mathbf{w},\mathbf{s};\phi)}\Big[\sum_{i=1}^{n}\log p\Big(\boldsymbol{\gamma}_i|\mathbf{w}_i,\mathbf{w}_{\varnothing};\nu\Big)\Big]\right)\right]}$$

which does not depend on the parameterization ω *of* $q(\mathbf{w}_{\varnothing};\omega)$.

Proof. Let us first recall our framework's approximate expression of the predictive distribution $q(\boldsymbol{\gamma}_*|\boldsymbol{\gamma})$ for the target model

$\boldsymbol{\gamma}_*$ given observations of existing, relevant models $\boldsymbol{\gamma}$:

$$\begin{aligned} q(\boldsymbol{\gamma}_*|\boldsymbol{\gamma}) &= \int p\Big(\mathbf{w}_\varnothing, \mathbf{w}, \mathbf{s}|\boldsymbol{\gamma}\Big)\Bigg[p\Big(\mathbf{w}_*|\mathbf{w}, \mathbf{s}; \phi\Big)p\Big(\boldsymbol{\gamma}_*|\mathbf{w}_*, \mathbf{w}_\varnothing; \nu\Big)\Bigg]\mathrm{d}\mathbf{w}_*\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{s}\mathrm{d}\mathbf{w}_\varnothing \\ &\simeq \int q\Big(\mathbf{w}_\varnothing, \mathbf{w}, \mathbf{s}; \omega\Big)\Bigg[p\Big(\mathbf{w}_*|\mathbf{w}, \mathbf{s}; \phi\Big)p\Big(\boldsymbol{\gamma}_*|\mathbf{w}_*, \mathbf{w}_\varnothing; \nu\Big)\Bigg]\mathrm{d}\mathbf{w}_*\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{s}\mathrm{d}\mathbf{w}_\varnothing \\ &= \int q\Big(\mathbf{w}_\varnothing; \omega\Big)\Bigg[p\Big(\mathbf{w}, \mathbf{s}; \phi\Big)p\Big(\mathbf{w}_*|\mathbf{w}, \mathbf{s}; \phi\Big)p\Big(\boldsymbol{\gamma}_*|\mathbf{w}_*, \mathbf{w}_\varnothing; \nu\Big)\mathrm{d}\mathbf{w}_*\mathrm{d}\mathbf{w}\mathrm{d}\mathbf{s}\Bigg]\mathrm{d}\mathbf{w}_\varnothing \\ &= \int q\Big(\mathbf{w}_\varnothing; \omega\Big)\mathbb{E}_{p(\mathbf{w},\mathbf{s};\phi)}\Bigg[\int p\Big(\mathbf{w}_*|\mathbf{w}, \mathbf{s}; \phi\Big)p\Big(\boldsymbol{\gamma}_*|\mathbf{w}_\varnothing, \mathbf{w}_*; \nu\Big)\mathrm{d}\mathbf{w}_*\Bigg]\mathrm{d}\mathbf{w}_\varnothing\ , \end{aligned} \tag{37}$$

where the first step is by our variational approximation scheme, the second step follows from the surrogate approximation $p(\mathbf{w}_\varnothing, \mathbf{w}, \mathbf{s}|\boldsymbol{\gamma}) \simeq q(\mathbf{w}_\varnothing, \mathbf{w}, \mathbf{s})$, and the third step follows from our choice of the surrogate $q(\mathbf{w}_\varnothing, \mathbf{w}, \mathbf{s}; \omega) = q(\mathbf{w}_\varnothing; \omega)p(\mathbf{w}|\mathbf{s}; \phi)\prod_{\kappa=1}^{e} p(\mathbf{s}_\kappa; \phi) = q(\mathbf{w}_\varnothing; \omega)p(\mathbf{w}, \mathbf{s}; \phi)$. Finally, plugging the expression of $q(\mathbf{w}_\varnothing)$ in Lemma 1 (at a global maximizer) into Eq. 37 yields the desired form of $\pi(\boldsymbol{\gamma}_*|\boldsymbol{\gamma})$. □

Lemma 2 thus establishes the form of the predictive distribution $q(\boldsymbol{\gamma}_*|\boldsymbol{\gamma})$ at optimality. We will exploit this result in **Part II** below to establish a non-trivial bound on the expected performance gap between an oracle model and a sampled model from $q(\boldsymbol{\gamma}_*|\boldsymbol{\gamma})$ on target task $\boldsymbol{\tau}_*$, which confirms the key result that we stated previously in Theorem 1 (Section 4) of the main text.

E.2. Part II

Let $\boldsymbol{\gamma}_* \in \mathcal{H}$ denotes a model for $\boldsymbol{\tau}_*$ in the hypothesis space $\mathcal{H}$ and $\mathbf{D}_*$ denotes the (unknown) data distribution of $\boldsymbol{\tau}_*$. Let $\ell_{\boldsymbol{\gamma}}(\mathbf{x}, y)$ denotes the non-negative evaluation loss of predicting $\boldsymbol{\gamma}(\mathbf{x})$ when the ground-truth is y. Examples of such evaluation loss function includes the squared error $\ell_{\boldsymbol{\gamma}}(\mathbf{x}, y) = (\boldsymbol{\gamma}(\mathbf{x}) - y)^2$ (for regression) and the 0-1 loss $\ell_{\boldsymbol{\gamma}}(\mathbf{x}, y) = \mathbb{I}(\boldsymbol{\gamma}(\mathbf{x}), y)$ (for classification). The generalized loss of a model $\boldsymbol{\gamma}_*$ on $\boldsymbol{\tau}_*$ is defined as,

$$\mathbf{G}\left(\boldsymbol{\gamma}_*\right) \triangleq \mathbb{E}_{(\mathbf{x},y)\sim\mathbf{D}_*}\Bigg[\ell_{\boldsymbol{\gamma}_*}\left(\mathbf{x}, y\right)\Bigg]\ . \tag{38}$$

We assume there exists an oracle model $\boldsymbol{\gamma}_*^\circ \in \mathcal{H}$ for $\boldsymbol{\tau}_*$ that achieves the minimum generalized loss, which is defined as:

$$\boldsymbol{\gamma}_*^\circ \triangleq \arg\min_{\boldsymbol{\gamma}_* \in \mathcal{H}} \mathbf{G}\left(\boldsymbol{\gamma}_*\right) \text{ or equivalently, } \mathbf{G}\left(\boldsymbol{\gamma}_*^\circ\right) = \min_{\boldsymbol{\gamma}_*} \mathbf{G}\left(\boldsymbol{\gamma}_*\right)\ . \tag{39}$$

We also assume that the evaluation loss admits the triangle inequality such that $\ell_{\boldsymbol{\gamma}}(\mathbf{x}, y) \leq \ell_{\boldsymbol{\gamma}}(\mathbf{x}, \boldsymbol{\gamma}'(\mathbf{x})) + \ell_{\boldsymbol{\gamma}'}(\mathbf{x}, y)$ and $\ell_{\boldsymbol{\gamma}}(\mathbf{x}, \boldsymbol{\gamma}'(\mathbf{x})) \leq \ell_{\boldsymbol{\gamma}}(\mathbf{x}, y) + \ell_{\boldsymbol{\gamma}'}(\mathbf{x}, y)$. It is easy to see that both examples above of the squared and 0-1 losses admit such triangle inequalities. Then, let q be the abbreviation for $q(\boldsymbol{\gamma}^*|\boldsymbol{\gamma})$ (as defined above). The generalized loss of sampling solution model from q for target task $\boldsymbol{\tau}_*$ with distribution $\mathbf{D}_*$ is defined as

$$\mathbf{G}\left(q\right) = \mathbb{E}_{\boldsymbol{\gamma}^*\sim q}\Bigg[\mathbb{E}_{(\mathbf{x},y)\sim\mathbf{D}_*}\Bigg[\ell_{\boldsymbol{\gamma}_*}\left(\mathbf{x}, y\right)\Bigg]\Bigg]\ . \tag{40}$$

To bound the gap between $\mathbf{G}(\boldsymbol{\gamma}_*^\circ)$ and $\mathbf{G}(q)$, we first establish the following immediate results in Lemma 3 and Lemma 4.

Intuition. In lay terms, Lemma 3 re-iterates a well-known result that can be used in Lemma 4 to bound the expected generalized loss $\mathbf{G}(q)$ of a model distribution in terms of auxiliary functions. This in turn can be bound by a small fraction of the oracle's generalized loss $\mathbf{G}(\boldsymbol{\gamma}_*^\circ)$ under a mild assumption regarding the performance of the oracle model on a random input. This is formally stated in Lemma 5 – see its premise assumption and its discussion that follows – which is the stepping stone to establish our key result in Theorem 1 (Section 4).

Lemma 3. *For any function $g(\mathbf{x})$ and any two distributions $p(\mathbf{x})$ and $\pi(\mathbf{x})$ on a certain measurable space, we have*

$$\mathbb{E}_{\mathbf{x}\sim p}\Bigg[g(\mathbf{x})\Bigg] \leq \mathbb{D}_{\mathrm{KL}}\Bigg(p\Big\|\pi\Bigg) + \log\mathbb{E}_{\mathbf{x}\sim\pi}\Bigg[\exp\Bigg(g(\mathbf{x})\Bigg)\Bigg]\ . \tag{41}$$

Proof. This is a well-known result often referred to as the **Change of Measure** inequality (Seldin et al., 2015). For the sake of self-containment, we concisely reproduce the its proof below:

$$\begin{aligned}
\mathbb{E}_{\mathbf{x}\sim p}\left[g(\mathbf{x})\right] &= \mathbb{E}_{\mathbf{x}\sim p}\left[\log\left(\frac{p(\mathbf{x})}{\pi(\mathbf{x})}\times\frac{\pi(\mathbf{x})}{p(\mathbf{x})}\times\exp\Big(g(\mathbf{x})\Big)\right)\right] \\
&= \mathbb{E}_{\mathbf{x}\sim p}\left[\log\frac{p(\mathbf{x})}{\pi(\mathbf{x})}\right] + \mathbb{E}_{\mathbf{x}\sim p}\left[\log\left(\frac{\pi(\mathbf{x})}{p(\mathbf{x})}\times\exp\Big(g(\mathbf{x})\Big)\right)\right] \\
&= \mathbb{D}_{\mathrm{KL}}\Big(p\|\pi\Big) + \mathbb{E}_{\mathbf{x}\sim p}\left[\log\left(\frac{\pi(\mathbf{x})}{p(\mathbf{x})}\times\exp\Big(g(\mathbf{x})\Big)\right)\right] \\
&\leq \mathbb{D}_{\mathrm{KL}}\Big(p\|\pi\Big) + \log\,\mathbb{E}_{\mathbf{x}\sim p}\left[\frac{\pi(\mathbf{x})}{p(\mathbf{x})}\times\exp\Big(g(\mathbf{x})\Big)\right] = \mathbb{D}_{\mathrm{KL}}\Big(p\|\pi\Big) + \log\mathbb{E}_{\mathbf{x}\sim\pi}\left[\exp\Big(g(\mathbf{x})\Big)\right], \qquad (42)
\end{aligned}$$

where the inequality follows from applying Jensen inequality, which completes our proof of Lemma 3.

□

Lemma 4. *For any reference distribution $\pi(\boldsymbol{\gamma}_*|\boldsymbol{\gamma})$, let $\mathbf{F}_{\boldsymbol{\gamma}_*^\circ}(\pi) \triangleq \mathbb{E}_{\mathbf{x}\sim\mathbf{D}_*}\mathbb{E}_{\boldsymbol{\gamma}_*\sim\pi}\left[\exp\left(\ell_{\boldsymbol{\gamma}_*}(\mathbf{x},\boldsymbol{\gamma}_*^\circ(\mathbf{x}))\right)\right]$. We have*

$$\Big|\mathbf{G}(q) - \mathbf{G}(\boldsymbol{\gamma}_*^\circ)\Big| \;\leq\; \mathbb{D}_{\mathrm{KL}}\Big(q\|\pi\Big) \;+\; \log\mathbf{F}_{\boldsymbol{\gamma}_*^\circ}\Big(\pi\Big)\,. \qquad (43)$$

Proof. By definition, $\mathbf{G}(\boldsymbol{\gamma}_*^\circ) \leq \mathbf{G}(\boldsymbol{\gamma}_*)$ for any choice of $\boldsymbol{\gamma}_*$. Hence, $\mathbf{G}(\boldsymbol{\gamma}_*^\circ) = \mathbb{E}_{\boldsymbol{\gamma}_*\sim q}[\mathbf{G}(\boldsymbol{\gamma}_*^\circ)] \leq \mathbb{E}_{\boldsymbol{\gamma}_*\sim q}[\mathbf{G}(\boldsymbol{\gamma}_*)] = \mathbf{G}(q)$. On the other hand, we also have

$$\begin{aligned}
\mathbf{G}(q) &= \mathbb{E}_{\boldsymbol{\gamma}_*\sim q}\mathbb{E}_{(\mathbf{x},y)\sim\mathbf{D}_*}\left[\ell_{\boldsymbol{\gamma}_*}(\mathbf{x},y)\right] \\
&\leq \mathbb{E}_{(\mathbf{x},y)\sim\mathbf{D}_*}\left[\mathbb{E}_{\boldsymbol{\gamma}_*\sim q}\left[\ell_{\boldsymbol{\gamma}_*^\circ}(\mathbf{x},y) + \ell_{\boldsymbol{\gamma}_*}\Big(\mathbf{x},\boldsymbol{\gamma}_*^\circ(\mathbf{x})\Big)\right]\right] \\
&= \mathbb{E}_{(\mathbf{x},y)\sim\mathbf{D}_*}\left[\ell_{\boldsymbol{\gamma}_*^\circ}(\mathbf{x},y)\right] + \mathbb{E}_{\mathbf{x}\sim\mathbf{D}_*}\left[\mathbb{E}_{\boldsymbol{\gamma}_*\sim q}\left[\ell_{\boldsymbol{\gamma}_*}\Big(\mathbf{x},\boldsymbol{\gamma}_*^\circ(\mathbf{x})\Big)\right]\right] \\
&= \mathbf{G}(\boldsymbol{\gamma}_*^\circ) + \mathbb{E}_{\mathbf{x}\sim\mathbf{D}_*}\left[\mathbb{E}_{\boldsymbol{\gamma}_*\sim q}\left[\ell_{\boldsymbol{\gamma}_*}\Big(\mathbf{x},\boldsymbol{\gamma}_*^\circ(\mathbf{x})\Big)\right]\right] \\
&\leq \mathbf{G}(\boldsymbol{\gamma}_*^\circ) + \mathbb{E}_{\mathbf{x}\sim\mathbf{D}_*}\left[\mathbb{D}_{\mathrm{KL}}\Big(q\|\pi\Big) + \log\mathbb{E}_{\boldsymbol{\gamma}_*\sim\pi}\left[\exp\Big(\ell_{\boldsymbol{\gamma}_*}\Big(\mathbf{x},\boldsymbol{\gamma}_*^\circ(\mathbf{x})\Big)\Big)\right]\right] \\
&= \mathbf{G}(\boldsymbol{\gamma}_*^\circ) + \mathbb{D}_{\mathrm{KL}}\Big(q\|\pi\Big) + \mathbb{E}_{\mathbf{x}\sim\mathbf{D}_*}\log\mathbb{E}_{\boldsymbol{\gamma}_*\sim\pi}\left[\exp\Big(\ell_{\boldsymbol{\gamma}_*}\Big(\mathbf{x},\boldsymbol{\gamma}_*^\circ(\mathbf{x})\Big)\Big)\right] \\
&\leq \mathbf{G}(\boldsymbol{\gamma}_*^\circ) + \mathbb{D}_{\mathrm{KL}}\Big(q\|\pi\Big) + \log\,\mathbb{E}_{\mathbf{x}\sim\mathbf{D}_*}\left[\mathbb{E}_{\boldsymbol{\gamma}_*\sim\pi}\left[\exp\Big(\ell_{\boldsymbol{\gamma}_*}\Big(\mathbf{x},\boldsymbol{\gamma}_*^\circ(\mathbf{x})\Big)\Big)\right]\right] \\
&= \mathbf{G}(\boldsymbol{\gamma}_*^\circ) + \mathbb{D}_{\mathrm{KL}}\Big(q\|\pi\Big) + \log\mathbf{F}_{\boldsymbol{\gamma}_*^\circ}\Big(\pi\Big)\,, \qquad (44)
\end{aligned}$$

where the first inequality is due to the triangle inequality of the evaluation loss, the second inequality follows from the change of measure inequality in Lemma 3, and the last inequality follows from Jensen inequality. Thus, combining this with the previously proved statement, we have

$$\mathbf{G}(\boldsymbol{\gamma}_*^\circ) \;\leq\; \mathbf{G}(q) \;\leq\; \mathbf{G}(\boldsymbol{\gamma}_*^\circ) + \mathbb{D}_{\mathrm{KL}}\Big(q\|\pi\Big) + \log\mathbf{F}_{\boldsymbol{\gamma}_*^\circ}\Big(\pi\Big)\,, \qquad (45)$$

which implies $|\mathbf{G}(q) - \mathbf{G}(\boldsymbol{\gamma}_*^\circ)| \leq \mathbb{D}_{\mathrm{KL}}\Big(q\|\pi\Big) + \log\mathbf{F}_{\boldsymbol{\gamma}_*^\circ}(\pi)$ as desired. This completes our proof of Lemma 4. □

Intuition. Next, we show that when the oracle model is highly accurate in the sense that the chance for it to incur a loss worse than a fraction of its generalized loss is vanishingly small – see Eq. (46), $\log \mathbf{F}_{\boldsymbol{\gamma}_*^\circ}(\pi)$ can be upper-bounded by a small fraction of $\mathbf{G}(\boldsymbol{\gamma}_*^\circ)$ and a log data term. This is formally stated in Lemma 5 below.

Lemma 5. *Assume there exist constants $c > 0$ and $r > 1$ such that*

$$\Pr_{(\mathbf{x},y)\sim\mathbf{D}_*}\left(\ell_{\boldsymbol{\gamma}_*^\circ}(\mathbf{x},y) \;>\; \frac{1}{r}\mathbf{G}\Big(\boldsymbol{\gamma}_*^\circ\Big)\right) \;\leq\; c\times\exp\left(\frac{1}{r}\mathbf{G}\Big(\boldsymbol{\gamma}_*^\circ\Big) - 2\sup\nolimits_{\boldsymbol{\gamma},\mathbf{x},y}\Big[\ell_{\boldsymbol{\gamma}}(\mathbf{x},y)\Big]\right). \tag{46}$$

Then, it can be shown that for $\mathbf{F}_{\boldsymbol{\gamma}_*^\circ}(\pi)$ as defined in Lemma 4 above,

$$\log\mathbf{F}_{\boldsymbol{\gamma}_*^\circ}\Big(\pi\Big) \;\leq\; \frac{1}{r}\mathbf{G}(\boldsymbol{\gamma}_*^\circ) \;+\; \log\left(\mathbb{E}_{(\mathbf{x},y)\sim\mathbf{D}_*}\left[\mathbb{E}_{\boldsymbol{\gamma}_*\sim\pi}\left[\exp\Big(\ell_{\boldsymbol{\gamma}_*}(\mathbf{x},y)\Big)\right]\right] + c\right). \tag{47}$$

Proof. To prove this, we first expand the expression of $\mathbf{F}_{\boldsymbol{\gamma}_*^\circ}(\pi)$ using the triangle inequality of the evaluation loss,

$$\begin{aligned}
\log\mathbf{F}_{\boldsymbol{\gamma}_*^\circ}\Big(\pi\Big) &= \log\mathbb{E}_{(\mathbf{x},y)\sim\mathbf{D}_*}\left[\mathbb{E}_{\boldsymbol{\gamma}_*\sim\pi}\left[\exp\Big(\ell_{\boldsymbol{\gamma}_*}\Big(\mathbf{x},\boldsymbol{\gamma}_*^\circ(\mathbf{x})\Big)\Big)\right]\right]\\
&\leq \log\mathbb{E}_{(\mathbf{x},y)\sim\mathbf{D}_*}\left[\mathbb{E}_{\boldsymbol{\gamma}_*\sim\pi}\left[\exp\Big(\ell_{\boldsymbol{\gamma}_*^\circ}\Big(\mathbf{x},y\Big) + \ell_{\boldsymbol{\gamma}_*}\Big(\mathbf{x},y\Big)\Big)\right]\right]\\
&= \log\mathbb{E}_{(\mathbf{x},y)\sim\mathbf{D}_*}\left[\exp\Big(\ell_{\boldsymbol{\gamma}_*^\circ}\Big(\mathbf{x},y\Big)\Big)\times\mathbb{E}_{\boldsymbol{\gamma}_*\sim\pi}\left[\exp\Big(\ell_{\boldsymbol{\gamma}_*}\Big(\mathbf{x},y\Big)\Big)\right]\right] \;\triangleq\; \log\mathbf{F}^\dagger_{\boldsymbol{\gamma}_*^\circ}\Big(\pi\Big).
\end{aligned} \tag{48}$$

Now, let $\psi \triangleq \sup_{\boldsymbol{\gamma},\mathbf{x},y}\ell_{\boldsymbol{\gamma}}(\mathbf{x},y)$ and $\delta = \Pr(\ell_{\boldsymbol{\gamma}_*^\circ}(\mathbf{x},y) > (1/r)\mathbf{G}(\boldsymbol{\gamma}_*^\circ))$. We can now split the space of $(\mathbf{x},y)$ into two parts: (a) $\mathbb{D}_*^{(a)} = \{(\mathbf{x},y) \mid \ell_{\boldsymbol{\gamma}_*^\circ}(\mathbf{x},y) \leq (1/r)\mathbf{G}(\boldsymbol{\gamma}_*^\circ)\}$; and (b) $\mathbb{D}_*^{(b)} = \{(\mathbf{x},y) \mid \ell_{\boldsymbol{\gamma}_*^\circ}(\mathbf{x},y) > (1/r)\mathbf{G}(\boldsymbol{\gamma}_*^\circ)\}$. As such, if $(\mathbf{x},y)$ is to be sampled uniformly in part (a), its sampling distribution is $\mathbf{D}_*^{(a)}(\mathbf{x},y) = \mathbb{I}((\mathbf{x},y)\in\mathbb{D}_*^{(a)})\times\mathbf{D}_*(\mathbf{x},y)/(1-\delta)$. Likewise, if $(\mathbf{x},y)$ is in part (b), its sampling distribution is $\mathbf{D}_*^{(b)}(\mathbf{x},y) = \mathbb{I}((\mathbf{x},y)\in\mathbb{D}_*^{(b)})\times\mathbf{D}_*(\mathbf{x},y)/\delta$. Thus,

$$\begin{aligned}
\mathbf{F}^\dagger_{\boldsymbol{\gamma}_*^\circ}\Big(\pi\Big) &\leq (1-\delta)\exp\left(\frac{1}{r}\mathbf{G}\Big(\boldsymbol{\gamma}_*^\circ\Big)\right)\mathbb{E}_{(\mathbf{x},y)\sim\mathbf{D}_*^{(a)}}\left[\mathbb{E}_{\boldsymbol{\gamma}_*\sim\pi}\left[\exp\Big(\ell_{\boldsymbol{\gamma}_*}\Big(\mathbf{x},y\Big)\Big)\right]\right]\\
&+ \delta\exp\Big(\psi\Big)\mathbb{E}_{(\mathbf{x},y)\sim\mathbf{D}_*^{(b)}}\left[\mathbb{E}_{\boldsymbol{\gamma}_*\sim\pi}\left[\exp\Big(\ell_{\boldsymbol{\gamma}_*}\Big(\mathbf{x},y\Big)\Big)\right]\right]\\
&\leq (1-\delta)\exp\left(\frac{1}{r}\mathbf{G}\Big(\boldsymbol{\gamma}_*^\circ\Big)\right)\mathbb{E}_{(\mathbf{x},y)\sim\mathbf{D}_*^{(a)}}\left[\mathbb{E}_{\boldsymbol{\gamma}_*\sim\pi}\left[\exp\Big(\ell_{\boldsymbol{\gamma}_*}\Big(\mathbf{x},y\Big)\Big)\right]\right] \;+\; \delta\exp\Big(2\psi\Big)\\
&\leq \exp\left(\frac{1}{r}\mathbf{G}\Big(\boldsymbol{\gamma}_*^\circ\Big)\right)\mathbb{E}_{(\mathbf{x},y)\sim\mathbf{D}_*}\left[\mathbb{E}_{\boldsymbol{\gamma}_*\sim\pi}\left[\exp\Big(\ell_{\boldsymbol{\gamma}_*}\Big(\mathbf{x},y\Big)\Big)\right]\right] \;+\; \delta\exp\Big(2\psi\Big)\\
&\leq \exp\left(\frac{1}{r}\mathbf{G}\Big(\boldsymbol{\gamma}_*^\circ\Big)\right)\mathbb{E}_{(\mathbf{x},y)\sim\mathbf{D}_*}\left[\mathbb{E}_{\boldsymbol{\gamma}_*\sim\pi}\left[\exp\Big(\ell_{\boldsymbol{\gamma}_*}\Big(\mathbf{x},y\Big)\Big)\right]\right] \;+\; c\times\exp\left(\frac{1}{r}\mathbf{G}(\boldsymbol{\gamma}_*^\circ) - 2\psi + 2\psi\right)\\
&= \exp\left(\frac{1}{r}\mathbf{G}\Big(\boldsymbol{\gamma}_*^\circ\Big)\right)\left[\mathbb{E}_{(\mathbf{x},y)\sim\mathbf{D}_*}\left[\mathbb{E}_{\boldsymbol{\gamma}_*\sim\pi}\left[\exp\Big(\ell_{\boldsymbol{\gamma}_*}\Big(\mathbf{x},y\Big)\Big)\right]\right] \;+\; c\right],
\end{aligned} \tag{49}$$

where the last inequality follows from our assumption above in Eq. (46). Then, taking logarithm on both sides of Eq. (49) and plugging the result in Eq. (48) yield Eq. (47), which completes our proof of Lemma 5. □

Key Result. Finally, chaining up the above results, we can obtain the following key result which is the main thrust of our theoretical contribution here. It also reveals important insights that explain the rationalities behind the (somewhat artificial) formulation of our personalized learning framework, as discussed below.

Theorem 1. Assume there exist constants $c > 0$ and $r > 1$ such that (same assumption from Lemma 5)

$$\Pr_{(\mathbf{x},y)\sim\mathbf{D}_*}\left(\ell_{\boldsymbol{\gamma}_*^\circ}(\mathbf{x},y) \;>\; \frac{1}{r}\mathbf{G}\Big(\boldsymbol{\gamma}_*^\circ\Big)\right) \;\leq\; c\times\exp\left(\frac{1}{r}\mathbf{G}\Big(\boldsymbol{\gamma}_*^\circ\Big) - 2\sup\nolimits_{\boldsymbol{\gamma},\mathbf{x},y}\Big[\ell_{\boldsymbol{\gamma}}(\mathbf{x},y)\Big]\right). \tag{50}$$

Then, for any reference distribution $\pi(\boldsymbol{\gamma}_*|\boldsymbol{\gamma})$, we have with probability at least $1-\delta$ over the sampling of an m-shot dataset $\{(\mathbf{x}_i,y_i)\}_{i=1}^m$ where $(\mathbf{x}_i,y_i)\sim\mathbf{D}_*$, the following inequality holds simultaneously for all choices of $q(\boldsymbol{\gamma}_*|\boldsymbol{\gamma})$:

$$\left|\mathbf{G}(q) - \mathbf{G}\Big(\boldsymbol{\gamma}_*^\circ\Big)\right| \;\leq\; \mathbb{D}_{\mathrm{KL}}\Big(q\Big\|\pi\Big) + \frac{1}{r}\mathbf{G}\Big(\boldsymbol{\gamma}_*^\circ\Big) + \log\left(\frac{1}{m}\sum_{i=1}^m \mathbb{E}_{\boldsymbol{\gamma}_*\sim\pi}\left[\exp\Big[\ell_{\boldsymbol{\gamma}_*}\Big(\mathbf{x}_i,y_i\Big)\Big]\right] + \mathbf{O}\left(\left[\frac{\log\left(\frac{4m}{\delta}\right)}{2m-1}\right]^{\frac{1}{2}}\right)\right)$$

where (as defined previously) $\boldsymbol{\gamma}_*^\circ$ denotes the oracle solution model of task $\boldsymbol{\tau}_*$.

Proof. First, we will chain up the results of Lemma 4 and Lemma 5 to obtain the following:

$$\begin{aligned}\left|\mathbf{G}(q) - \mathbf{G}(\boldsymbol{\gamma}_*^\circ)\right| \;&\leq\; \mathbb{D}_{\mathrm{KL}}\Big(q\|\pi\Big) \;+\; \log\mathbf{F}_{\boldsymbol{\gamma}_*^\circ}\Big(\pi\Big)\\ &\leq\; \mathbb{D}_{\mathrm{KL}}\Big(q\|\pi\Big) \;+\; \frac{1}{r}\mathbf{G}(\boldsymbol{\gamma}_*^\circ) \;+\; \log\left(\mathbb{E}_{(\mathbf{x},y)\sim\mathbf{D}_*}\left[\mathbb{E}_{\boldsymbol{\gamma}_*\sim\pi}\left[\exp\Big(\ell_{\boldsymbol{\gamma}_*}(\mathbf{x},y)\Big)\right]\right] + c\right). \end{aligned}\tag{51}$$

Now, to bound the last term on the RHS of the above inequality, we define the following auxiliary loss $h_{\boldsymbol{\gamma}_*}(\mathbf{x},y) = \exp(\ell_{\boldsymbol{\gamma}_*}(\mathbf{x},y) - \sup\,\ell_{\boldsymbol{\gamma}_*}(\mathbf{x},y))$ where the supremum is over the choice of $\boldsymbol{\gamma}$ and $(\mathbf{x},y)$. As such, $h_{\boldsymbol{\gamma}_*}(\mathbf{x},y)\in(0,1)$.

Then, applying the PAC-Bayes result in (McAllester, 1999) to relate the generalized and empirical loss of sampling solution model $\boldsymbol{\gamma}_*\sim\pi$ for task $\boldsymbol{\tau}_*$ with data distribution $\mathbf{D}_*$ using instance loss $h_{\boldsymbol{\gamma}_*}(\mathbf{x},y)$, we have with probability at least $1-\delta$ over the choice of $\{(\mathbf{x}_i,y_i)\}_{i=1}^m$,

$$\begin{aligned}\mathbb{E}_{(\mathbf{x},y)\sim\mathbf{D}_*}\left[\mathbb{E}_{\boldsymbol{\gamma}_*\sim\pi}\Big[h_{\boldsymbol{\gamma}_*}\Big(\mathbf{x},y\Big)\Big]\right] \;&\leq\; \frac{1}{m}\sum_{i=1}^m \mathbb{E}_{\boldsymbol{\gamma}_*\sim\pi}\left[h_{\boldsymbol{\gamma}_*}\Big(\mathbf{x}_i,y_i\Big)\right] \;+\; \left(\frac{\mathbb{D}_{\mathrm{KL}}\Big(\pi\|\pi\Big) + \log\left(\frac{4m}{\delta}\right)}{2m-1}\right)^{\frac{1}{2}}\\ &=\; \frac{1}{m}\sum_{i=1}^m \mathbb{E}_{\boldsymbol{\gamma}_*\sim\pi}\left[h_{\boldsymbol{\gamma}_*}\Big(\mathbf{x}_i,y_i\Big)\right] \;+\; \left(\frac{\log\left(\frac{4m}{\delta}\right)}{2m-1}\right)^{\frac{1}{2}}. \end{aligned}\tag{52}$$

Note that we use a special case of the PAC-Bayesian bound here where the target and reference distributions are the same. Thus, multiplying both sides of the above inequality with $\exp(\sup\,\ell_{\boldsymbol{\gamma}_*}(\mathbf{x},y))$ yields

$$\begin{aligned}\mathbb{E}_{(\mathbf{x},y)\sim\mathbf{D}_*}\left[\mathbb{E}_{\boldsymbol{\gamma}_*\sim\pi}\left[\exp\Big(\ell_{\boldsymbol{\gamma}_*}\Big(\mathbf{x},y\Big)\Big)\right]\right] \;&\leq\; \frac{1}{m}\sum_{i=1}^m \mathbb{E}_{\boldsymbol{\gamma}_*\sim\pi}\left[\exp\Big(\ell_{\boldsymbol{\gamma}_*}\Big(\mathbf{x}_i,y_i\Big)\Big)\right]\\ &\quad+\; \exp\Big(\sup\,\ell_{\boldsymbol{\gamma}_*}\Big(\mathbf{x}_i,y_i\Big)\Big)\left(\frac{\log\left(\frac{4m}{\delta}\right)}{2m-1}\right)^{\frac{1}{2}}\\ &=\; \frac{1}{m}\sum_{i=1}^m \mathbb{E}_{\boldsymbol{\gamma}_*\sim\pi}\left[\exp\Big(\ell_{\boldsymbol{\gamma}_*}\Big(\mathbf{x}_i,y_i\Big)\Big)\right] \;+\; \mathbf{O}\left(\left[\frac{\log\left(\frac{4m}{\delta}\right)}{2m-1}\right]^{\frac{1}{2}}\right). \end{aligned}\tag{53}$$

Finally, taking the logarithm of both sides, plugging the result into Eq. (51), and absorbing the constant c into the big-$\mathbf{O}$ notation complete our proof of Theorem 1. Thus, setting $C_m = \mathbf{O}((\log(4m/\delta)/(2m-1))^{1/2})$ reproduces the key result that we mentioned in Theorem 1 in the main text. Now, combining this result with our previously established result in **Part I** reveals an interesting insight into the fine-tuning part of our personalized learning framework as discussed in Corollary 1 below. We note that an informal discussion of this had been provided previously Section 4 above.

Corollary 1. Assuming the same assumption of Theorem 1 regarding the oracle solution model for $\boldsymbol{\tau}_*$ and constructing the reference distribution $\pi(\boldsymbol{\gamma}_*|\boldsymbol{\gamma})$ following its established parameterization in Lemma 2, with arbitrarily high probability,

the gap between the generalized losses of $\boldsymbol{\gamma}_*^\circ$ and $q(\boldsymbol{\gamma}_*|\boldsymbol{\gamma})$ is at most $\mathbf{G}(\boldsymbol{\gamma}_*^\circ)/r$ when $m \to \infty$ and the optimization of Eq. (16) reaches a global optimizer. Furthermore, when m is small, the established bound in Theorem 1 also reveals how the process of fine-tuning a personalized model using the few-shot dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^m$ can be incorporated seamlessly into the learning of the meta-model representation to provably control the bound gap.

Proof. To see this, we first take a closer look at the proved bound of Theorem 1,

$$\left|\mathbf{G}(q) - \mathbf{G}\Big(\boldsymbol{\gamma}_*^\circ\Big)\right| \;\leq\; \mathbb{D}_{\mathrm{KL}}\Big(q\Big\|\pi\Big) + \frac{1}{r}\mathbf{G}\Big(\boldsymbol{\gamma}_*^\circ\Big) + \log\left(\frac{1}{m}\sum_{i=1}^m \mathbb{E}_{\boldsymbol{\gamma}_*\sim\pi}\left[\exp\Big[\ell_{\boldsymbol{\gamma}_*}\Big(\mathbf{x}_i, y_i\Big)\Big]\right] + \mathbf{O}\left(\left[\frac{\log\left(\frac{4m}{\delta}\right)}{2m-1}\right]^{\frac{1}{2}}\right)\right)$$

Now, using the results of **Part I** (see Lemmas 1 and 2) we know that when the optimization of Eq. (16) reaches a universal optimizer, q converges to π in Lemma 2. At that point, $\mathbb{D}_{\mathrm{KL}}(q\|\pi) = 0$ and the first term on the RHS of the above bound disappears. As for the remaining terms, taking limit when $m \to \infty$ on both sides of the reduced bound further zero out the log term, thus leaving only a fraction $1/r$ of the oracle loss.

Secondly, in case m is small, the excess loss term depends heavily on $\log(\frac{1}{m}\sum_{i=1}^m \mathbb{E}_{\boldsymbol{\gamma}_*\sim\pi}[\exp[\ell_{\boldsymbol{\gamma}_*}\Big(\mathbf{x}_i, y_i)]])$ which immediately suggests a loss function for fine-tuning and an appropriate scale to combine with the personalization loss. In particular, it can be seen from the above arguments that optimizing q and π simultaneously via minimizing

$$\mathbf{H}(q,\pi) \;=\; -\mathbf{L}(q) \;+\; \log\left(\frac{1}{m}\sum_{i=1}^m \mathbb{E}_{\boldsymbol{\gamma}_*\sim\pi}\left[\exp\Big[\ell_{\boldsymbol{\gamma}_*}\Big(\mathbf{x}_i, y_i\Big)\Big]\right]\right) \tag{54}$$

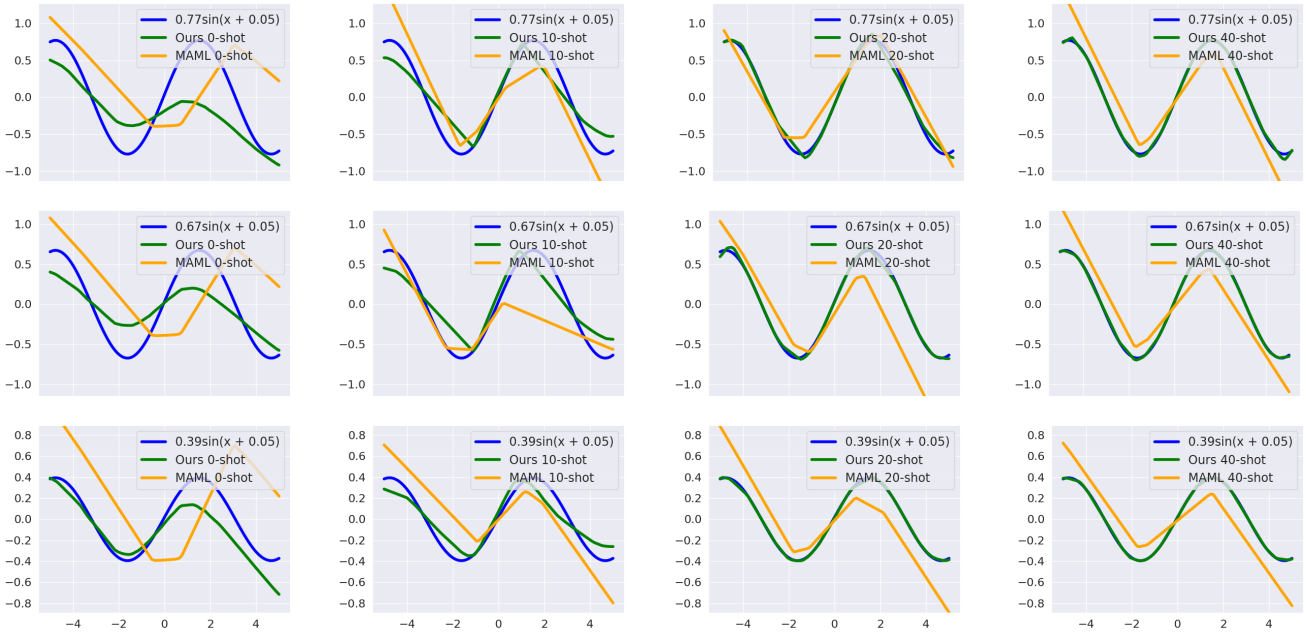
will result in a performance gap bound by $\mathbf{G}(\boldsymbol{\gamma}_*^\circ)/r$ and a minimized excess loss involving the few-shot dataset for fine-tuning. Here, we ignore the constant term that scales with double log, which is negligible. This is true because the parameterizations of q and π are decoupled (see Lemma 1) and the data term in the above loss does not depend on q, which effectively forces $\mathbb{D}_{\mathrm{KL}}(q\|\pi)$ to be zero if (q,π) is a universal minimizer of $\mathbf{H}(q,\pi)$. Thus, optimizing $\mathbf{H}$ instead of the original ELBO objective in Eq. (16) allows us to not only zero out the divergence term but also reduce the data fit further on the few-shot dataset in the new task context (i.e., fine-tuning).

Remark. The bound above also meets our sanity check in the limit of information and quality of the oracle model. To elaborate, in the limit of information, both the divergence and data terms disappear as argued above, thus leaving a fraction $1/r$ of the oracle loss $\mathbf{G}(\boldsymbol{\gamma}_*^\circ)$. Now, if the target task $\boldsymbol{\tau}_*$ can be solved with high accuracy by an oracle model bounded within a model space $\mathcal{H}$, one would expect r (see the assumption in Lemma 5) to be large and c to be small while $\mathbf{G}(\boldsymbol{\gamma}_*^\circ)$ to be vanishingly small. Both of which contribute towards zeroing out the remaining loss terms of our bound in the limit of information and oracle quality, thus resulting in a personalized model distribution whose generalized performance is arbitrarily close to that of an oracle model.

In light of this, the bound in Theorem 1 also suggests an intuition regarding an intricate trade-off between the bound gap (in the limit of information) and the expressiveness of the model space. On one hand, if the model space is highly expressive, it is likely that our assumption for the oracle model will hold with large r which tighten the excess loss but in exchange, it will be harder for the divergence term to zero out via optimizing the meta-model loss in Eq. (16) as there are now more parameters to be optimized. On the other hand, if the model space is reduced in complexity, it is easier for the optimization of Eq. (16) to find a global optimizer and zero out the divergence term but with the reduced complexity, the oracle model (i.e., the best in the defined model space) becomes less accurate and hence, the assumption might only hold with large c and small r. This has the effect of pushing the bound towards being vacuous. When this happens, model fusion become perhaps a less well-defined task since even the oracle model cannot sufficiently solve the task with high accuracy.

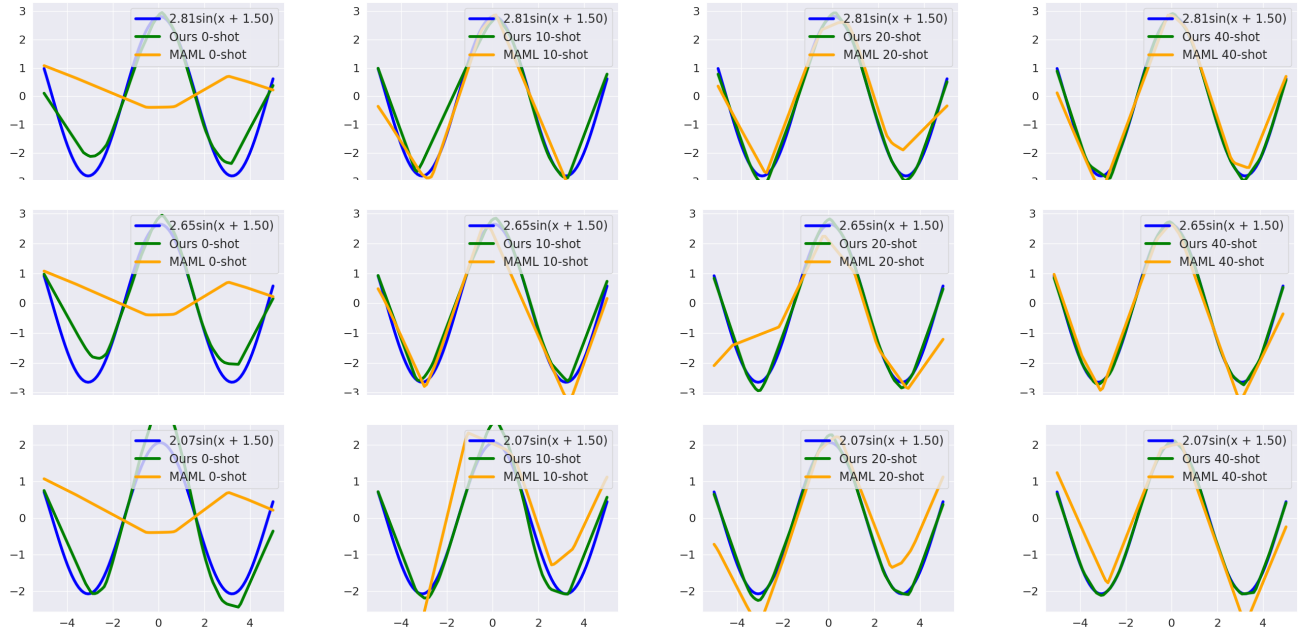
F. Extra Visualized Performance Plots across Different Sine Tasks in Different Sub-Domains

Figures 5, 6 and 7 provide additional visualized fitting performance of our personalized model towards various sine tasks across all 5 polarized sine domains (corresponding to those reported in Appendix A).

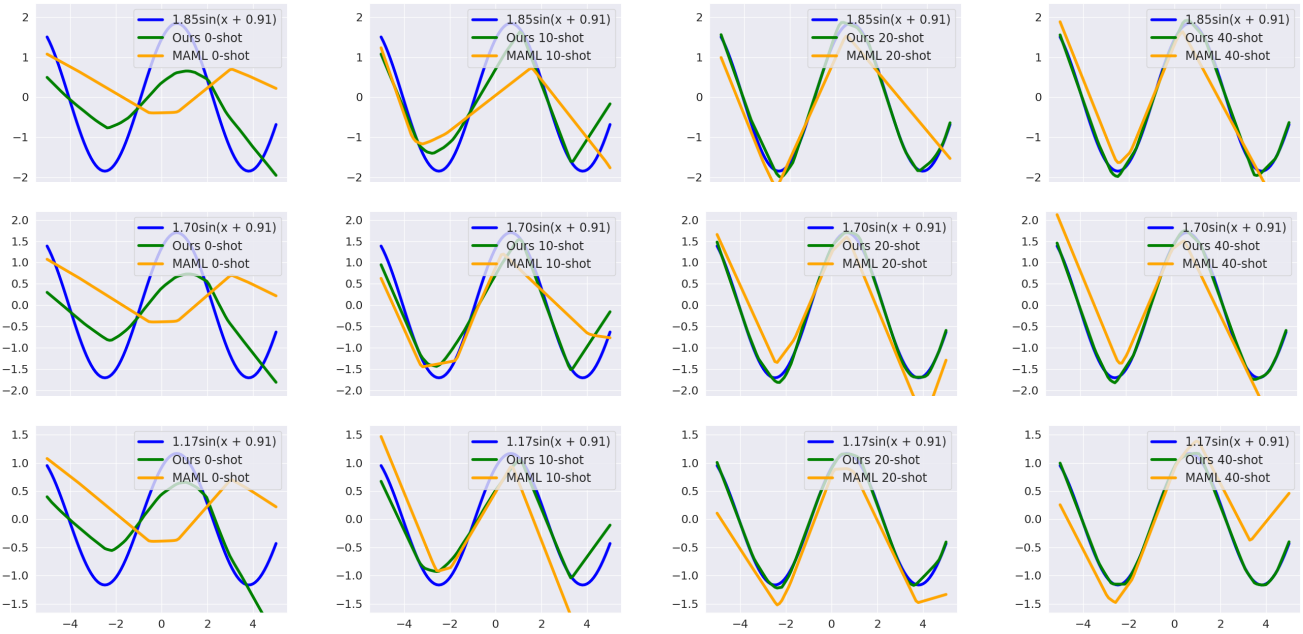


Sub-Domain 1: $a \sim \mathrm{U}[0, 1]$, $b = 0.91$

Figure 5. (Part 1) Additional visual excerpts demonstrating how well the warm-start neural net with architecture [1-100-1] generated by MAML and our method fit 2 unseen sine functions $y = a \sin x + b$ of sub-domain 1 in 0-, 10-, 20- and 40-shot settings.

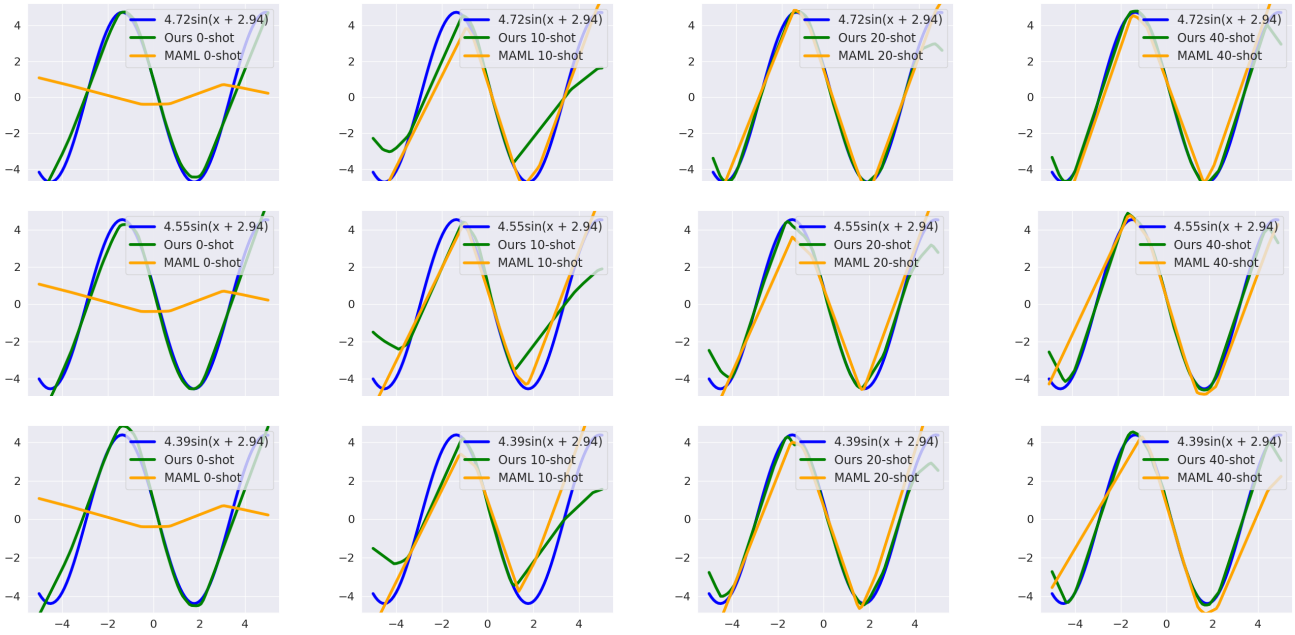


Sub-Domain 3: $a \sim \mathrm{U}[2, 3]$, $b = 1.50$

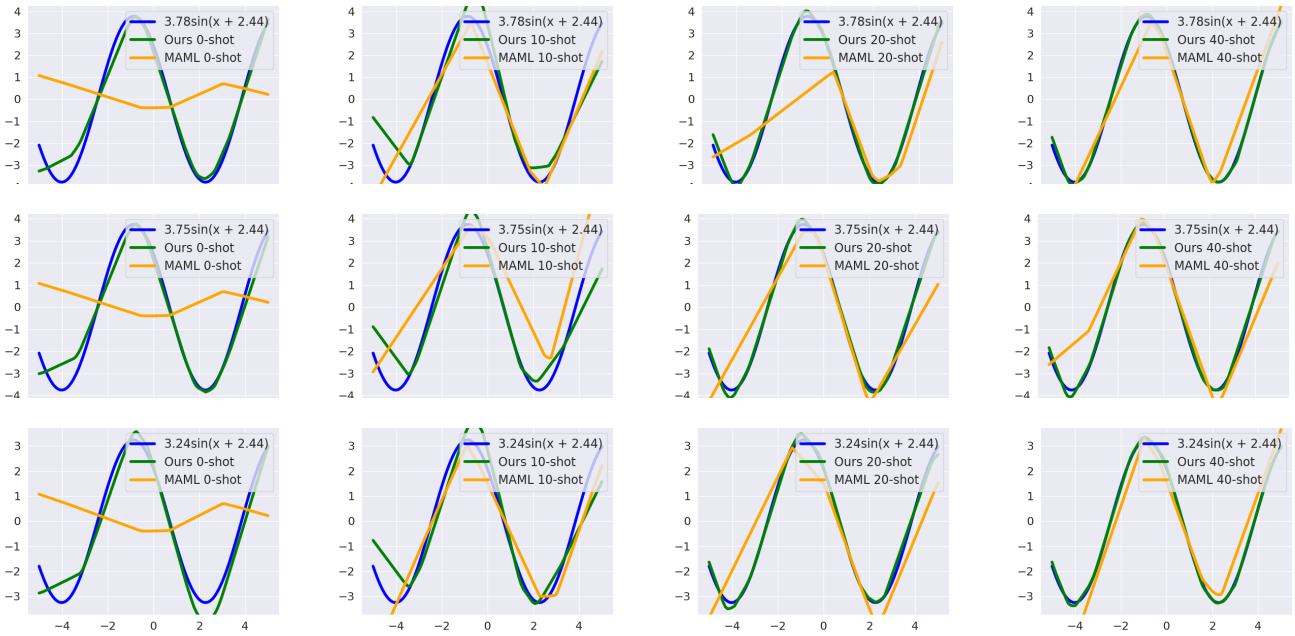


Sub-Domain 2: $a \sim \mathrm{U}[1, 2]$, $b = 0.91$

Figure 6. (Part 2) Additional visual excerpts demonstrating how well the warm-start neural net with architecture [1-100-1] generated by MAML and our method fit 2 unseen sine functions $y = a \sin x + b$ of sub-domains 2 and 3 in 0-, 10-, 20- and 40-shot settings.



Sub-Domain 5: $a \sim \mathrm{U}[4, 5]$, $b = 2.94$



Sub-Domain 4: $a \sim \mathrm{U}[3, 4]$, $b = 2.44$

Figure 7. (Part 3) Additional visual excerpts demonstrating how well the warm-start neural net with architecture [1-100-1] generated by MAML and our method fit 2 unseen sine functions $y = a \sin x + b$ of sub-domains 4 and 5 in 0-, 10-, 20- and 40-shot settings.