A. Comparison with Model-Agnostic Meta-Learning (MAML) (Finn et al., 2017)

In this section, we highlight and explain the key differences between the experiment setting of our proposed method and that of MAML (Finn et al., 2017):

First, MAML assumes access to all local datasets and that they can be centralized for processing. This allows MAML to (in the context of our MNIST experiment) sample pairs of different digit images from each local dataset to train a specialized neural network (NNs) that learns a discriminative pattern for each pair of different digits. Such patterns are then aggregated via the gradient adaptation procedure described in Section 2 above. That is, with access to the full dataset, MAML is able to choose the most relevant task samples and their distribution that are most representative for the target task with limited training data. This implicit assumption is, however, not valid in many practical settings with private data that cannot be accessed by the meta learning algorithm.

Second, MAML learns a fast adaptable initializer $\gamma = \gamma_*$ – see Eq. (1) – by imposing the same parameterization and choice of training algorithm across tasks. This appears restrictive in task/data federated settings where pre-trained models solving previous tasks might have different choices of parameterization and training algorithm since they could be generated at different times and under different computing platforms. In contrast, our proposed algorithm is devised to cater specifically to such situations where we only have access to those pre-trained models (potentially with different parameterizations) via their black-box interface from which we can query their output η for each candidate input $\mathbf{x}$. This also implies MAML has access to information with higher quality than ours since MAML has access to the ground-truth y instead of η.

B. Parameterization of Generative and Inference Networks in Figure 1

This section provides detailed information regarding the parameterization of our generative and inference networks in Figure 1. In particular, for the MNIST experiments, we have:

Generative Network:

$p_\theta(\mathbf{x}|\mathbf{w}, \mathbf{z})$ is parameterized as a 3-layer NN with 2 hidden, linear layers and θ being its defining parameters. The input layer accepts batches of $(|\mathbf{w}| + |\mathbf{z}| + |\mathbf{x}|)$-dimensional inputs where $|\mathbf{w}| = 2$ and $|\mathbf{z}| = 5$ are the latent dimensions of the corresponding components that form the factorized embedding space. $|\mathbf{x}| = 28 \times 28 = 784$ which represents the dimension of the flatten digit image of size 28×28. Each hidden layer has 100 hidden neurons, each of which is activated by a ReLU function. Neurons of the last layer is activated by a sigmoid function to guarantee that the pixel values of the reconstructed image $\mathbf{x}$ are from 0 to 1. Additionally, the output of the first hidden layer is passed through a batch normalization layer to avoid the gradient update from being influenced by a few unusually large components.

$p_\alpha(\eta|\mathbf{z})$ is likewise parameterized as a 3-layer NN with 2 hidden, linear layers and α being its defining parameters. Its input layer accepts batches of $|\mathbf{z}|$-dimensional inputs where $|\mathbf{z}| = 5$. Its output layer returns 10-dimensional output η. The rest of the parameterization is similar to $p_\theta(\mathbf{x}|\mathbf{w}, \mathbf{z})$ above.

$p_\gamma(\mathbf{w}|\tau)$ is parameterized as a 3-layer NN with 1 hidden, linear layer comprising of 100 hidden neurons with γ defines its set of parameters. Each hidden neuron is activated by a ReLU unit. The input layer accepts batches of 10-dimensional inputs. The output of the hidden layer will be normalized via a batch normalization layer. The normalized output will then be fed separately to two parallel linear layers comprising of $|\mathbf{w}| = 2$ neurons each. The outputs of these layers represent the mean and diagonal covariance of a multivariate Gaussian that distributes $\mathbf{w}$.

$p(\tau)$ and $p(\mathbf{z})$ are, on the other hand, fixed to be the empirical distribution over a discrete set $\{\tau_1, \tau_2, \ldots, \tau_p\}$ and an isotropic multivariate Gaussian.

For Mini-ImageNet experiments, we adapt the likelihood network $p_\theta(\mathbf{x}|\mathbf{w}, \mathbf{z})$ to a 4-layer transpose convolutional network.

Inference Network:

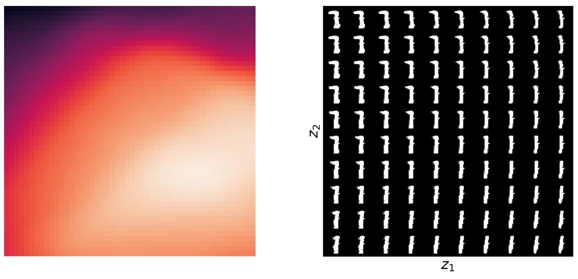
$q_\phi(\mathbf{z}|\mathbf{x}, \eta)$ is parameterized as a 3-layer NN with 2 hidden, linear layer comprising of 100 hidden neurons. Each hidden neuron is activated by a ReLU unit. The input layer accepts batches of $|\mathbf{x}| + |\eta|$-dimensional inputs where as mentioned above $|\mathbf{x}| = 784$ and $|\eta| = 10$. The output of the hidden layer will be normalized via a batch normalization layer. The normalized output will then be fed separately to two parallel linear layers comprising of $|\mathbf{w}| = 2$ neurons each. The outputs of these layers represent the mean and diagonal covariance of a multivariate Gaussian that distributes $\mathbf{z}$.

$q_\phi(\mathbf{w}|\mathbf{x}, \tau)$ is parameterized similar to $q_\phi(\mathbf{z}|\mathbf{x}, \eta)$. The only difference is that the outputs of its final two parallel, linear layers represent the mean and diagonal covariance of a multivariate Gaussian that distributes $\mathbf{w}$ with $|\mathbf{w}| = 2$.

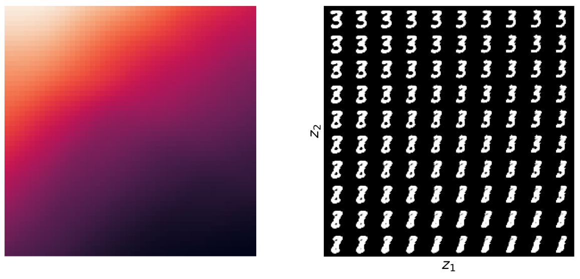
For Mini-ImageNet experiments, we adapt the inference network $q_\phi(\mathbf{z}|\mathbf{x}, \eta)$ and $q_\phi(\mathbf{w}|\mathbf{x}, \tau)$ to be 4-layer convolutional networks. Note that while we abuse the notation ϕ to denote the collective set of parameters defining the above two inference networks, these are parameterized separately and do not share weights.

Regularizer Network:

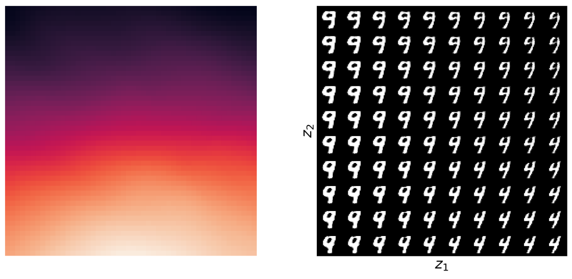
$q_\psi(\eta|\mathbf{w})$ is parameterized as a 3-layer NN with 2 hidden layers and θ being its defining parameters. The input layer accepts batches of $|\mathbf{w}|$-dimensional inputs where $|\mathbf{w}| = 2$. Each hidden layer has 100 hidden neurons, each of which is activated by a ReLU function. Neurons of the last layer is activated by a sigmoid function. Additionally, the output of the first hidden layer is passed through a batch normalization layer to avoid the gradient update from running astray. The final output of this network is $|\eta|$-dimensional where $|\eta| = 10$.



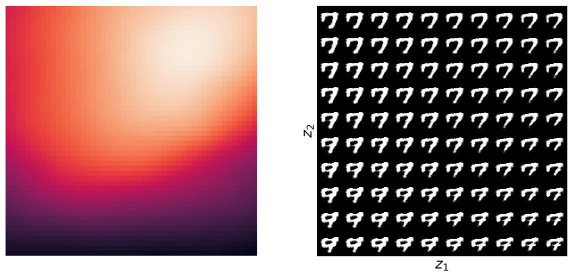
(a)



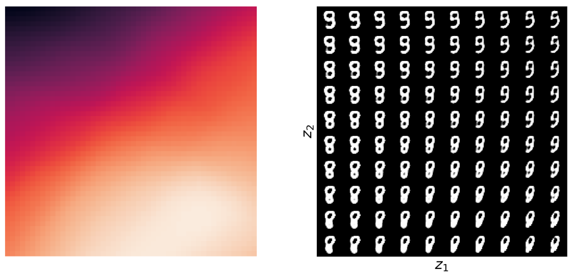
(b)



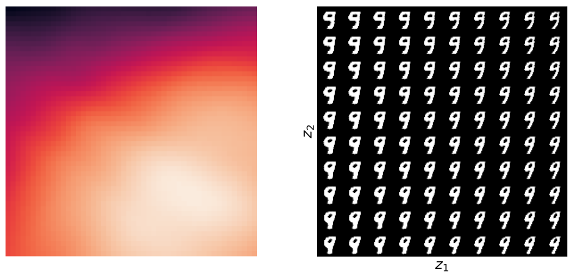
(c)



(d)



(e)



(f)

Figure 3. Visualizing plots of 6 different prototypes' activation and encoded patterns with respect to 6 different samples of $\mathbf{w}$. For each prototype, its encoded patterns are visualized via plotting their reconstructed images across the space of $\mathbf{z}$. The corresponding activation heat-maps of these prototypes are also visualized when they process image input of digit (a) 1, (b) 3, (c) 4, (d) 7, (e) 8 and (f) 9. The plotting procedure of these figures was described in detail earlier in Section 4.2.

C. More Explanations for the Experiments in Section 4.2

Figure 2 provides (1) a visualization of handwriting patterns encoded by a particular prototype $G_{\mathbf{w}}(\eta|\mathbf{x};\theta,\alpha)$, which is built around a task-specific pattern $\mathbf{w}$ – see Eq. (10). These patterns are visualized (via the learned decoder $p_\theta(\mathbf{x}|\mathbf{z},\mathbf{w})$) across the latent space of task-agnostic concept $\mathbf{z}$; and (2) the activation (expressed in form of a heat-map where brighter color indicates stronger activation) of these handwriting patterns which is computed via the atomic term $p_\theta(\mathbf{x}|\mathbf{z},\mathbf{w})p_\alpha(\eta|\mathbf{z})$ that forms the prototype's probabilistic prediction in Eq. (10). By inspection, one can further observe that the activation heat-maps of the prototypes shown above are *visually correct* with respect to input images of digits 3 and 8: The activation is strongest at the encoded handwriting patterns corresponding to reconstructed image variants of digits 3 and 8. Interested readers can find more visual excerpts of such prototypes in Figure 3 above.

Finally, to complete our visual demonstration, we further show below how the activation heat-map of the same prototype (i.e., same $\mathbf{w}$ and $\mathbf{z}$-grid) changes when we change its input $\mathbf{x}$ from one digit to another. Again, by inspection, one can observe that both activation heat-maps are *visually correct*: The activation scores are highest at those encoded handwriting patterns that can be reconstructed into similar image variants of the input images. These visual evidences therefore support our claim that the decomposed prototypes are semantically coherent. That is, they always activate the right handwriting patterns (among their encoded patterns) when processing an arbitrary digit image.

Remark. We also note that (as observed in the visual excerpts below) not all handwriting patterns are captured by a single prototype. In fact, each single prototype is shown to only capture a few patterns that only make up a subset of digits. Thus, to perform well in a new classification context that involves unseen digits, these prototypes need to be combined to form the right patterns that characterize the test digits well. This is achieved via solving Eq. (12) above which is the recomposition step that follows the embedding and decomposition procedure detailed in Algorithm 1 above.

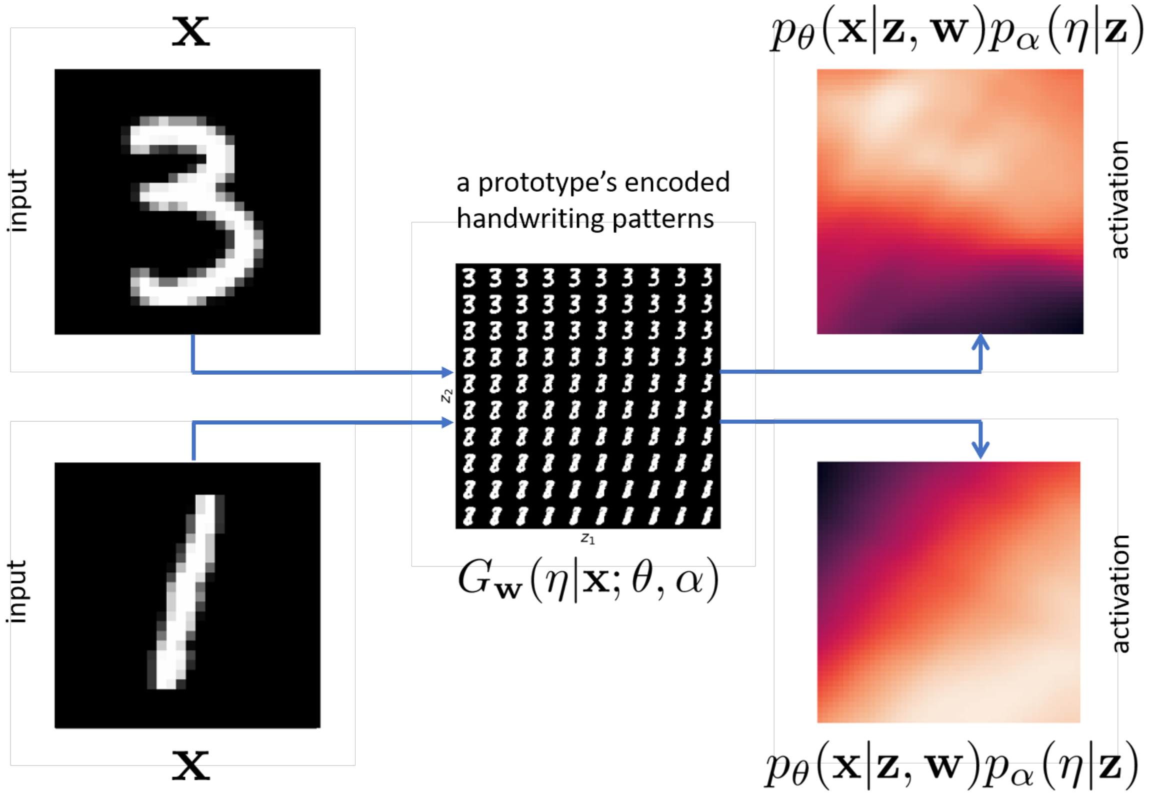

$\mathbf{x}$
input
a prototype's encoded handwriting patterns
z_2
z_1
$G_{\mathbf{w}}(\eta|\mathbf{x};\theta,\alpha)$
$p_\theta(\mathbf{x}|\mathbf{z},\mathbf{w})p_\alpha(\eta|\mathbf{z})$
activation
input
$\mathbf{x}$
activation
$p_\theta(\mathbf{x}|\mathbf{z},\mathbf{w})p_\alpha(\eta|\mathbf{z})$

Figure 4. Visualizing plot of activation changes when we change the image input $\mathbf{x}$ to a fixed prototype. When $\mathbf{x}$ is an image of digit 3, the encoded patterns of the prototype that correspond to an image variant (after being decoded) of digit 3 receive strongest activation (i.e., locations annotated with brightest colors). Likewise, for the same prototype, when $\mathbf{x}$ is an image of digit 1, the activation becomes strongest at patterns that can be decoded in a variant image (slightly distorted) of digit 1.