Experiment 1 assesses the performance of our aggregation method against conventional techniques and standard 10-fold cross-validation, following the protocol outlined in our main paper. We employ the full suite of geometrically constructed Gaussian mixture datasets, averaging performance metrics as in Table 1. Starting with the same initial number of aggregated KuLSIF models as prior experiments, we incrementally increase ensemble size. As shown in Figure 1 (left), standard model averaging (solid red) suffers immediate degradation when incorporating models with suboptimal hyperparameters. Bayesian model averaging (solid green) is more robust but still exhibits rising approximation error with ensemble growth. In contrast, our method (solid brown) maintains stable, superior performance. While cross-validation (solid blue) is less sensitive to hyperparameter expansion, it incurs higher computational cost, as reflected by the less favorable scaling of trained models (dashed blue vs. dashed brown).
In **Experiment 2**, we evaluate the scalability of our aggregation method with increasing data dimensionality, using the same setup as Experiment 1 on all Gaussian mixture datasets for training and evaluation. We employ the KuLSIF and LR DRE methods. As shown in Figure 1 (right), our aggregation method (dashed lines) maintains robust performance across dimensions, whereas the cross-validation approach (solid lines) degrades due to growing difficulty in hyperparameter selection.

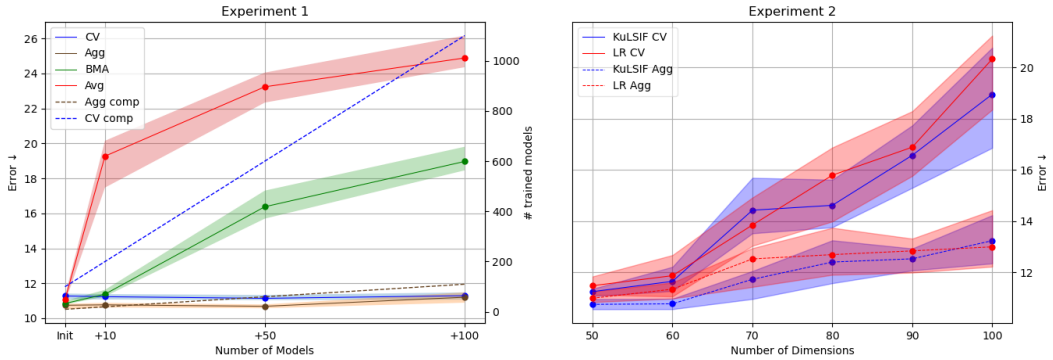

Figure 1: Performance of our method when increasing the number of models/dimensions, median error and 50% confidence intervals. Left: Our aggregation method (solid brown) remains robust to ensemble members with suboptimal parameters. Other methods (red, green) exhibit increasing approximation error. Our method is computationally more efficient than cross-validation (dotted blue vs dotted orange). Right: Our method (dotted lines) scales favorably when increasing the dimensionality of the data.

In **Experiment 3**, we extend our investigation of Ablation 1 (Table 1, main paper) by introducing an experiment that involves tuning a hyperparameter that is not related to regularization. Specifically, we consider Kernel Mean Matching (KMM), a method in which selecting an appropriate kernel bandwidth is known to be particularly challenging. As a baseline, we employ the widely used Median Heuristic for setting the kernel bandwidth. Following the same experimental protocol used for the Gaussian mixture datasets in previous experiments, we evaluate the performance of KMM under different bandwidth settings. As shown in Table 1, aggregating KMM models with varying kernel bandwidths leads to improved prediction accuracy compared to using the Median Heuristic alone.

	Experiment 3								
Methods	c3,d1.70	c2,d1.72	c2d1.59	c1d1.55	c2d1.78	c2d1.55	c3d1.57	c2d1.61	c3d1.46
KMM Med. Heur.	16.783(±0.091)	21.051(±0.104)	20.601(±0.097)	19.833(±0.081)	17.231(±0.089)	19.364(±0.097)	20.911(±0.153)	19.981(±0.099)	18.59(±0.077)
KMM Agg	**13.223**(±**0.071**)	**18.155**(±**0.097**)	**16.532**(±**0.083**)	**15.319**(±**0.067**)	**13.968**(±**0.066**)	**15.473**(±**0.081**)	**17.003**(±**0.114**)	**16.712**(±**0.073**)	**14.538**(±**0.035**)

Table 1: Mean and standard deviation (after ±) of error for Kernel Mean Matching on the geometrically constructed datasets over ten different sample draws from P and Q. Using our aggregation method, we can decrease the error compared to setting the kernel bandwidth by the Median Heuristic.