In **Experiment 1** we compare the computational complexity of our aggregation method with 10-fold cross-validation. Starting with the same number of ensemble members as in prior experiments, we incrementally increase the number of models to be aggregated. As can be seen in Figure 1 cross-validation (blue) scales computationally less favorably than our approach (orange).

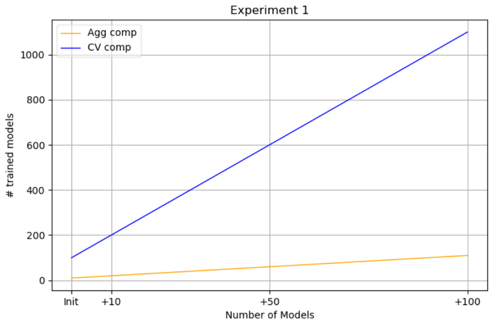

Figure 1: Computational complexity when increasing the number of ensemble models. The x-axis shows the number of ensemble members, the y-axis the number of models that have to be trained. Our method (orange) is computationally more efficient than cross-validation (blue).

Experiment 2 assesses the performance of our aggregation method against conventional ensembling techniques and standard 10-fold cross-validation. We use all Gaussian mixture datasets, averaging performance metrics as in Table 1 of our main paper. Starting with the same initial number of aggregated KuLSIF models as in prior experiments, we incrementally increase the number of ensemble members. As shown in Figure 2 (left), standard model averaging (red) and Bayesian model averaging (green) suffer from performance degradation as the ensemble size increases, our method (orange) maintains stable, superior performance.

In **Experiment 3**, we evaluate the scalability of our aggregation method with increasing data dimensionality on the Gaussian mixture datasets. We employ the KuLSIF and LR DRE methods. As shown in Figure 2 (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.

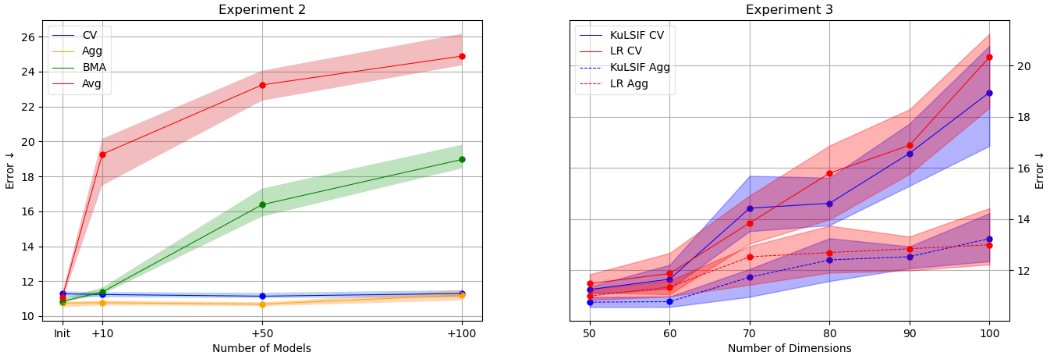

Figure 2: Performance of our method when increasing the number of models/dimensions, median error and 50% confidence intervals. Left: Our aggregation method (orange) remains robust to ensemble members with suboptimal hyperparameters. Other methods (red, green) exhibit increasing approximation error. Right: Our method (dotted lines) scales favorably when increasing the dimensionality of the data.

In **Experiment 4**, 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. We use Kernel Mean Matching (KMM), where selecting an appropriate kernel bandwidth is known to be particularly challenging and CV cannot be used. We compare to the widely used Median Heuristic for setting the kernel bandwidth. As shown in Table 1, aggregating KMM models with varying kernel bandwidths leads to improved prediction accuracy compared to using the Median Heuristic.

	Experiment 4								
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.