A Bias–Variance Decomposition for Ensembles over Multiple Synthetic Datasets

Ossi Räisä
University of Helsinki
ossi.raisa@helsinki.fi

Antti Honkela
University of Helsinki
antti.honkela@helsinki.fi

Abstract

Recent studies have highlighted the benefits of generating multiple synthetic datasets for supervised learning, from increased accuracy to more effective model selection and uncertainty estimation. These benefits have clear empirical support, but the theoretical understanding of them is currently very light. We seek to increase the theoretical understanding by deriving bias-variance decompositions for several settings of using multiple synthetic datasets, including differentially private synthetic data. Our theory yields a simple rule of thumb to select the appropriate number of synthetic datasets in the case of mean-squared error and Brier score. We investigate how our theory works in practice with several real datasets, downstream predictors and error metrics. As our theory predicts, multiple synthetic datasets often improve accuracy, while a single large synthetic dataset gives at best minimal improvement, showing that our insights are practically relevant.

1 INTRODUCTION

Synthetic data has recently attracted significant attention for several applications in machine learning. The idea is to generate a dataset that preserves the population-level attributes of the real data. This makes the synthetic data useful for analysis, while also accomplishing a secondary task, such as improving model evaluation (van Breugel et al., 2023b), fairness (van Breugel et al., 2021), data augmentation (Antoniou et al., 2018; Das et al., 2022) or privacy (Liew et al., 1985; Rubin, 1993). For privacy, *differential privacy* (DP) (Dwork et al., 2006) is often combined with synthetic data generation (Hardt et al., 2012; McKenna et al., 2021) to achieve provable privacy protection, since releasing synthetic data without DP can be vulnerable to membership inference attacks (Stadler et al., 2022; van Breugel et al., 2023c; Meeus et al., 2023).

Several lines of work have considered generating multiple synthetic datasets from one real dataset for various purposes, including statistical inference (Rubin, 1993; Raghunathan et al., 2003; Räisä et al., 2023a) and supervised learning (van Breugel et al., 2023a), the latter of which is our focus.

In supervised learning, van Breugel et al. (2023a) proposed an ensemble method of generative models. They propose generating multiple synthetic datasets independently from an arbitrary generative model, training a predictive model separately on each synthetic dataset, and ensembling these models by averaging their predictions. They call this a *deep generative ensemble* (DGE). We drop the word "deep" in this paper and use the term *generative ensemble*, as we do not require the generator to be deep in any sense. The DGE was empirically demonstrated to be beneficial in several ways by van Breugel et al. (2023a), including predictive accuracy, model evaluation, model selection, and uncertainty estimation. Followup work has applied DGEs to improve model evaluation under distribution shifts and for small subgroups (van Breugel et al., 2023b).

However, van Breugel et al. (2023a) have very little theoretical analysis of the generative ensemble. Their theoretical justification assumes the data is generated from a posterior predictive distribution. This assumption can be justified heuristically for deep generative models through a Bayesian interpretation of deep ensembles (Lakshminarayanan et al., 2017; Wilson & Izmailov, 2020; Wilson, 2021). However, this justification only applies to generators with highly multi-modal losses, like neural networks. It also does not provide any insight on how different choices in the setup, like the choice of downstream predictor, affect the performance

Proceedings of the 28[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258.

of the ensemble.

The bias-variance decomposition (Geman et al., 1992) and its generalisations (Ueda & Nakano, 1996; Wood et al., 2023) are classical tools that provide insight into supervised learning problems. The standard bias-variance decomposition from Geman et al. (1992) considers predicting $y \in \mathbb{R}$ given features $x \in \mathcal{X}$, using predictor $g(x; D)$ that receives training data D. The mean-squared error (MSE) of g can be decomposed into bias, variance, and noise terms:

$$\underbrace{\mathop{\mathbb{E}}_{D,y}[(y-g)^2]}_{\text{MSE}} = \underbrace{(f(x) - \mathop{\mathbb{E}}_{D}[g])^2}_{\text{Bias}} + \underbrace{\mathop{\text{Var}}_{D}[g]}_{\text{Variance}} + \underbrace{\mathop{\text{Var}}_{y}[y]}_{\text{Noise}}, \quad (1)$$

where we have shortened $g = g(x; D)$ and $f(x) = \mathbb{E}_y[y]$ is the optimal predictor. x is considered fixed, so all the random quantities in the decomposition are implicitly conditioned on x. While MSE is typically only used with regression problems, the decomposition also applies to the Brier score (Brier, 1950) in classification, which is simply the MSE of class probabilities.

We seek to provide deeper theoretical understanding of generative ensembles through bias-variance decompositions, which provide a more fine-grained view of how different choices in the setup affect the end result.

Contributions.

1. We derive a bias-variance decomposition for the MSE or Brier score of generative ensembles under an i.i.d. assumption in Theorem 2.1. This decomposition is simply a sum of interpretable terms, which makes it possible to predict how various choices affect the error.

2. We derive several practical considerations from our decomposition, including a simple rule of thumb to select the number of synthetic datasets in Section 2.3. In summary, 2 synthetic datasets give 50% of the potential benefit from multiple synthetic datasets, 10 give 90% of the benefit and 100 give 99% of the benefit. This also applies to bagging ensembles like random forests, which is likely to be of independent interest. The benefit is a result of reduced variance, so the theory predicts high-variance models to receive the largest benefit.

3. We generalise the decomposition of Theorem 2.1 to differentially private (DP) generation algorithms that do not split their privacy budget between the multiple synthetic datasets[1] in Theorem 2.3, to non-i.i.d. synthetic data in Appendix B, and to Bregman divergences in Appendix C.

4. We evaluate the performance of a generative ensemble on several datasets, downstream prediction algorithms, and error metrics in Section 3. The results show that out theory applies in practice: multiple synthetic datasets generally decrease all of the error metrics, and our rule of thumb makes accurate predictions when the error can be accurately estimated. In contrast, a single large synthetic dataset provides small benefits at best, and can even increase error.

[1]Theorem 2.1 applies to DP algorithms that split the privacy budget, but it is not clear if multiple synthetic datasets are beneficial with these algorithms, as splitting the privacy budget between more synthetic datasets means that each one requires adding more noise, degrading the quality of the synthetic data.

1.1 Related Work

Ensembling generative models has been independently proposed several times (Wang et al., 2016; Choi et al., 2019; Luzi et al., 2020; Chen et al., 2023; van Breugel et al., 2023a). The inspiration of our work comes from van Breugel et al. (2023a), who proposed ensembling predictive models over multiple synthetic datasets, and empirically studied how this improves several aspects of performance in classification.

Generating multiple synthetic datasets has also been proposed with statistical inference in mind, for both frequentist (Rubin, 1993; Raghunathan et al., 2003; Räisä et al., 2023b), and recently Bayesian (Räisä et al., 2023a) inference. These works use the multiple synthetic datasets to correct statistical inferences for the extra uncertainty from synthetic data generation.

The bias-variance decomposition was originally derived by Geman et al. (1992) for standard regressors using MSE as the loss. James (2003) generalised the decomposition to symmetric losses, and Pfau (2013) generalised it to Bregman divergences.

Ueda & Nakano (1996) were the first to study the MSE bias–variance decomposition with ensembles, and Gupta et al. (2022); Wood et al. (2023) later extended the ensemble decomposition to other losses. All of these also apply to generative ensembles, but they only provide limited insight for them, as they do not separate the synthetic data generation-related terms from the downstream-related terms.

2 BIAS-VARIANCE DECOMPOSITIONS FOR GENERATIVE ENSEMBLES

In this section, we make our main theoretical contributions. We start by defining our setting in Section 2.1. We then derive a bias-variance decomposition for generative ensembles in Section 2.2, and a simple rule of thumb that can be used to select the number of syn-

thetic datasets in Section 2.3. The first decomposition does not apply to some differentially private synthetic data generation methods, so we generalise the decomposition to apply to those in Section 2.4. We also present decompositions that apply non-i.i.d. settings and to Bregman divergences in Appendices B and C, but they are not as informative as the others.

2.1 Problem Setting

We consider multiple synthetic datasets $D_s^{1:m}$, each of which is used to train an instance of a predictive model g. These models are combined into an ensemble $\hat{g}$ by averaging, so

$$\hat{g}(x; D_s^{1:m}) = \frac{1}{m}\sum_{i=1}^{m} g(x; D_s^i). \tag{2}$$

We allow g to be random, so g can for example internally select among several predictive models to use randomly.

We assume that each synthetic dataset D_s^i is generated from a generator with parameters θ_i, and that each generator is run with an independent random seed, so generations are independent given the parameters $\theta_{1:m}$. We also assume that each generator is trained on the real data D_r with an independent random seed, and there are no other dependencies between the training processes besides the real data, so the θ_i are i.i.d. given D_r. This is how synthetic datasets are sampled in DGE (van Breugel et al., 2023a), where θ_i are the parameters of the generative neural network from independent training runs. This also encompasses bootstrapping, where θ_i would be the real dataset[2], and $p(D_s|\theta_i)$ is the bootstrapping.

We will also consider a more general setting that applies to some differentially private synthetic data generators that do not fit into this setting in Section 2.4. In Appendix B, we consider settings without the i.i.d. assumptions.

2.2 Mean-squared Error Decomposition

Next, we present our main theorem. With the conditional i.i.d. assumptions detailed in Section 2.1, all of the $\theta_i|D_r$ distributions are identical and independent, so we use θ as a shorthand for a random variable with this distribution in this section. Similarly, the $D_s^i|\theta_i$ distribution only depends on θ_i, but not i with these assumptions, so we use $D_s|\theta$ as a shorthand.

Theorem 2.1. *Let the parameters for m generators $\theta_i \sim p(\theta|D_r)$, $i = 1, \dots, m$, be i.i.d. Let the synthetic datasets be $D_s^i \sim p(D_s|\theta_i)$ independently, and let $\hat{g}(x; D_s^{1:m}) = \frac{1}{m}\sum_{i=1}^{m} g(x; D_s^i)$. Then the mean-squared error in predicting y from x decomposes into six terms: model variance (MV), synthetic data variance (SDV), real data variance (RDV), synthetic data bias (SDB), model bias (MB), and noise* $\mathrm{Var}_y[y]$*:*

$$\mathrm{MSE} = \frac{1}{m}\mathrm{MV} + \frac{1}{m}\mathrm{SDV} + \mathrm{RDV} + (\mathrm{SDB}+\mathrm{MB})^2 + \mathop{\mathrm{Var}}_{y}[y], \tag{3}$$

where

$$\begin{aligned}
\mathrm{MSE} &= \mathop{\mathbb{E}}_{y, D_r, D_s^{1:m}}[(y - \hat{g}(x; D_s^{1:m}))^2] \\
\mathrm{MV} &= \mathop{\mathbb{E}}_{D_r,\theta} \mathop{\mathrm{Var}}_{D_s|\theta}[g(x; D_s)] \\
\mathrm{SDV} &= \mathop{\mathbb{E}}_{D_r} \mathop{\mathrm{Var}}_{\theta|D_r} \mathop{\mathbb{E}}_{D_s|\theta}[g(x; D_s)] \\
\mathrm{RDV} &= \mathop{\mathrm{Var}}_{D_r} \mathop{\mathbb{E}}_{D_s|D_r}[g(x; D_s)] \\
\mathrm{SDB} &= \mathop{\mathbb{E}}_{D_r}[f(x) - \mathop{\mathbb{E}}_{\theta|D_r}[f_\theta(x)]] \\
\mathrm{MB} &= \mathop{\mathbb{E}}_{D_r,\theta}[f_\theta(x) - \mathop{\mathbb{E}}_{D_s|\theta}[g(x; D_s)]].
\end{aligned} \tag{4}$$

$f(x) = \mathbb{E}_y[y]$ is the optimal predictor for real data, $\theta \sim p(\theta|D_r)$ is a single sample from the identical distribution of the generator parameters θ_i, $D_s \sim p(D_s|\theta)$ is a single sample of the synthetic data generating process given θ, and f_θ is the optimal predictor for the synthetic data generating process with parameters θ. All random quantities are implicitly conditioned on x.

The proofs of all theorems are in Appendix A.

To intuitively explain what each term measures, we split the whole generative ensemble pipeline into three steps:

1. Sample real data D_r,
2. given D_r, train generative model, resulting in θ,
3. given θ, sample synthetic data D_s, train downstream model g and make a prediction on x.

In the full generative ensemble, steps 2 and 3 are repeated m times.

MV measures the variance of step 3, averaged over the randomness in steps 1 and 2. SDV measures the variance of the average prediction from step 3 over the randomness in step 2, and also averages over the randomness in step 1. RDV measures the variance of the average prediction from the combination of steps 2 and 3, over the randomness in step 1.

f_θ represents the optimal predictor if the synthetic data generation in step 3 were the real data generating process. SDB measures how much the average $f(\theta)$, over the randomness in step 2, differs from the actual

[2]The θ_i random variables are i.i.d. if they are deterministically equal.

optimal predictor f on the real data, and averages the difference over the randomness in step 1. MB measures how much the downstream model's average predictions, over the randomness in step 3, differ from f_θ, and averages the difference over the randomness in steps 1 and 2. $\mathrm{Var}_y[y]$ is the noise term also present in the classical bias-variance decomposition in (1), which represents inherent randomness in the prediction problem that cannot be removed by any predictor.

Note that the MV, SDV, RDV, SDB, MB and $\mathrm{Var}_y[y]$ terms do not depend on m. Changing m only affects the impact of MV and SDV on MSE, as seen in (3).

Multidimensional y. For a multidimensional $y \in \mathbb{R}^d$, we have

$$\mathop{\mathbb{E}}_{y,D_r,D_s^{1:m}}\left[||y-\hat{g}||_2^2\right] = \sum_{j=1}^{d} \mathop{\mathbb{E}}_{y,D_r,D_s^{1:m}}\left[(y_j-\hat{g}_j)^2\right], \quad (5)$$

where we have shortened $\hat{g} = \hat{g}(x; D_s^{1:m})$ and y_j and $\hat{g}_j$ index over the dimensions. Since the right hand side is a sum of one-dimensional MSEs, our theory can also be applied to multi-dimensional y.

Relation to Brier Score. The Brier score (Brier, 1950) can be defined in two ways for binary classification. The first way to is pick a class, here class 0, and define

$$\mathrm{BS}_1 = \mathop{\mathbb{E}}_{y,D}[(y_0 - g_0(x;D))^2], \quad (6)$$

where y_0 is the indicator for the true class being 0, and $g_0(x;D)$ is the predicted probability of class 0. The second way is to define

$$\mathrm{BS}_2 = \mathop{\mathbb{E}}_{y,D}[||y - g(x;D)||_2^2], \quad (7)$$

where y is a one-hot-encoding of the true class, and $g(x;D)$ is the vector of predicted probabilities. The second definition also extends to problems with more than two classes. For binary classification, $\mathrm{BS}_2 = 2 \cdot \mathrm{BS}_1$ since probabilities sum to 1, so it does not matter which class is picked for the first definition.

The first definition is an MSE, and the second is a sum of one-dimensional MSEs, so our theory applies to both, including multiclass problems through the extension to multidimensional y.

Practical Considerations We can derive the following practical considerations from Theorem 2.1:

1. There is a simple rule-of-thumb on how many synthetic datasets are beneficial: m synthetic datasets give a $1 - \frac{1}{m}$ fraction of the possible benefit from multiple synthetic datasets. For example, $m = 2$ gives 50% of the benefit, $m = 10$ gives 90% and $m = 100$ gives 99%. This rule-of-thumb can be used to predict the MSE with many synthetic datasets from the results on just two synthetic datasets. More details in Section 2.3.

2. Increasing the number of synthetic datasets m reduces the impact of MV and SDV. This means that high-variance models, like interpolating decision trees or 1-nearest neighbours, benefit the most from multiple synthetic datasets. The size of this benefit can be empirically estimated using the prediction rule of Section 2.3.

3. The quality metrics for a synthetic data generator should include metrics that compare the distribution of the synthetic[3] data to the real distribution, instead of just comparing a synthetic dataset to the real dataset. Examples of the latter are all metrics that return the optimal value for a generator that just returns the real data, like comparisons between the marginals of the real and synthetic datasets, comparisons of downstream ML performance, and metrics like density/coverage (Naeem et al., 2020). One example of the former is the authenticity metric of (Alaa et al., 2022), though it only attempts to capture how well the generator generalises instead of copying the real data.

The last point can be derived by considering two extreme scenarios:

1. Generator is fitted perfectly, and generates from the real data generating distribution. Now

   $$\mathrm{MV} = \mathop{\mathrm{Var}}_{D_r}[g(x;D_r)], \quad \mathrm{MB} = \mathop{\mathbb{E}}_{D_r}[f(x) - g(x;D_r)]$$
   $$\mathrm{SDV} = \mathrm{RDV} = \mathrm{SDB} = 0.$$

   With one synthetic dataset, the result is the standard bias-variance trade-off. With multiple synthetic datasets, the impact of MV can be reduced, reducing the error compared to just using real data.

2. Return the real data: $D_s|D_r = D_r$ deterministically.[4]

   $$\mathrm{RDV} = \mathop{\mathrm{Var}}_{D_r}[g(x;D_r)], \quad \mathrm{MB} = \mathop{\mathbb{E}}_{D_r}[f(x) - g(x;D_r)]$$
   $$\mathrm{MV} = \mathrm{SDV} = \mathrm{SDB} = 0.$$

   The result is the standard bias-variance trade-off. The number of synthetic datasets does not matter, as all of them would be the same anyway.

[3] Specifically, the conditional distribution of synthetic data given real data.

[4] Theorem 2.1 applies in this scenario even with multiple synthetic datasets, as the deterministically identical synthetic datasets are independent as random variables.

While both of these scenarios are unrealistic, they may be approximated by a well-performing algorithm. The generator from Scenario 2 is the optimal generator for metrics that compare the synthetic dataset to the real dataset, while the generator of Scenario 1 is optimal for metrics that compare the synthetic data and real data distributions. Multiple synthetic datasets are only beneficial in Scenario 1, which means that metrics comparing the distributions are more meaningful when multiple synthetic datasets are considered.

2.3 Estimating the effect of Multiple Synthetic Datasets

Next, we consider estimating the variance terms MV and SDV in Theorem 2.1 from a small number of synthetic datasets and a test set. These estimates can then be used to asses if more synthetic datasets should be generated, and how many more are useful.

We can simplify Theorem 2.1 to

$$\text{MSE} = \frac{1}{m}\text{MV} + \frac{1}{m}\text{SDV} + \text{Others}, \tag{8}$$

where Others does not depend on the number of synthetic datasets m. The usefulness of more synthetic datasets clearly depends on the magnitude of MV+SDV compared to Others.

Since MSE depends on m, we can add a subscript to denote the m in question: MSE_m. Now (8) leads to the following corollary.

Corollary 2.2. *In the setting of Theorem 2.1, we have*

$$\text{MSE}_m = \text{MSE}_1 - \left(1 - \frac{1}{m}\right)(\text{MV} + \text{SDV}). \tag{9}$$

Proof. The claim follows by expanding MSE_1 and MSE_m with (8). □

If we have two synthetic datasets, we can estimate MV+ SDV $= 2(\text{MSE}_1 - \text{MSE}_2)$, which gives the estimator

$$\text{MSE}_m = \text{MSE}_1 - 2\left(1 - \frac{1}{m}\right)(\text{MSE}_1 - \text{MSE}_2). \tag{10}$$

If we have more than two synthetic datasets, we can set $x_m = 1 - \frac{1}{m}$ and $y_m = \text{MSE}_m$ in (9):

$$y_m = y_1 + x_m(\text{MV} + \text{SDV}), \tag{11}$$

so we can estimate MV + SDV from linear regression on (x_m, y_m). However, this will likely have a limited effect on the accuracy of the MSE_m estimates, as it will not reduce the noise in estimating MSE_1, which has a significant effect in (9).

From (9), we see that MV + SDV is the maximum reduction in MSE that can be obtained from multiple synthetic datasets. This means that $2(\text{MSE}_1 - \text{MSE}_2)$, or the linear regression estimate from (11), can be used as a diagnostic to check whether generating multiple synthetic datasets is worthwhile.

All terms in (8)-(11) depend on the target features x. We would like our estimates to be useful for typical x, so we will actually want to estimate $\mathbb{E}_x(\text{MSE}_m)$. Equations (8)-(11) remain valid if we take the expectation over x, so we can simply replace the MSE terms with their estimates that are computed from a test set.

Computing the estimates in practice will require that the privacy risk of publishing the test MSE is considered acceptable. The MSE for the estimate can also be computed from a separate validation set to avoid overfitting to the test set, but the risk of overfitting is small in this case, as m has a monotonic effect on the MSE. Both of these caveats can be avoided by choosing m using the rule of thumb that m synthetic datasets give a $1 - \frac{1}{m}$ of the potential benefit of multiple synthetic datasets, which is a consequence of (9).

Note that this MSE estimator can be applied to bagging ensembles (Breiman, 1996) like random forests (Breiman, 2001), since bootstrapping is a very simple form of synthetic data generation.

2.4 Differentially Private Synthetic Data Generators

Generating and releasing multiple synthetic datasets could increase the associated disclosure risk. One solution to this is *differential privacy* (DP) (Dwork et al., 2006; Dwork & Roth, 2014), which is a property of an algorithm that formally bounds the privacy leakage that can result from releasing the output of that algorithm. DP gives a quantitative upper bound on the privacy leakage, which is known as the privacy budget. Achieving DP requires adding extra noise to some point in the algorithm, lowering the utility of the result.

If the synthetic data is to be generated with DP, there are two possible ways to handle the required noise addition. The first is splitting the privacy budget across the m synthetic datasets, and run the DP generation algorithm separately m times. Theorem 2.1 applies in this setting. However, it is not clear if multiple synthetic datasets are beneficial in this case, as splitting the privacy budget requires adding more noise to each synthetic dataset. This also means that the rule of thumb from Section 2.3 will not apply. Most DP synthetic data generation algorithms would fall into this category (Aydore et al., 2021; Chen et al., 2020; Harder et al., 2021; Hardt et al., 2012; Liu et al., 2021; McKenna et al., 2019, 2021) if used to generate

multiple synthetic datasets.

The second possibility is generating all synthetic datasets based on a single application of a DP mechanism. Specifically, a noisy summary $\tilde{s}$ of the real data is released under DP. The parameters $\theta_{1:m}$ are then sampled i.i.d. conditional on $\tilde{s}$, and the synthetic datasets are sampled conditionally on the $\theta_{1:m}$. This setting includes algorithms that release a posterior distribution under DP, and use the posterior to generate synthetic data, like the NAPSU-MQ algorithm (Räisä et al., 2023b) and DP variational inference (DPVI) (Jälkö et al., 2017, 2021).[5]

The synthetic datasets are not i.i.d. given the real data in the second setting, so the setting described in Section 2.1 and assumed in Theorem 2.1 does not apply. However, the synthetic datasets are i.i.d. given the noisy summary $\tilde{s}$, so we obtain a similar decomposition as before.

Theorem 2.3. *Let the parameters for m generators $\theta_i \sim p(\theta|\tilde{s})$, $i = 1, \dots, m$, be i.i.d. given a DP summary $\tilde{s}$. Let the synthetic datasets be $D_s^i \sim p(D_s|\theta_i)$ independently, and let $\hat{g}(x; D_s^{1:m}) = \frac{1}{m}\sum_{i=1}^m g(x; D_s^i)$. Then*

$$\begin{aligned}\text{MSE} = \frac{1}{m}\text{MV} + \frac{1}{m}\text{SDV} + \text{RDV} + \text{DPVAR} \\ + (\text{SDB} + \text{MB})^2 + \operatorname*{Var}_y[y],\end{aligned} \tag{12}$$

where

$$\begin{aligned}\text{MSE} &= \mathop{\mathbb{E}}_{y, D_r, \tilde{s}, D_s^{1:m}}[(y - \hat{g}(x; D_s^{1:m}))^2] \\ \text{MV} &= \mathop{\mathbb{E}}_{D_r, \tilde{s}, \theta} \operatorname*{Var}_{D_s|\theta}[g(x; D_s)] \\ \text{SDV} &= \mathop{\mathbb{E}}_{D_r, \tilde{s}} \operatorname*{Var}_{\theta|\tilde{s}} \mathop{\mathbb{E}}_{D_s|\theta}[g(x; D_s)] \\ \text{RDV} &= \operatorname*{Var}_{D_r} \mathop{\mathbb{E}}_{D_s|D_r}[g(x; D_s)]\end{aligned} \tag{13}$$

$$\begin{aligned}\text{DPVAR} &= \mathop{\mathbb{E}}_{D_r} \operatorname*{Var}_{\tilde{s}|D_r} \mathop{\mathbb{E}}_{D_s|\tilde{s}}[g(x; D_s)] \\ \text{SDB} &= \mathop{\mathbb{E}}_{D_r, \tilde{s}}[f(x) - \mathop{\mathbb{E}}_{\theta|\tilde{s}}[f_\theta(x)]] \\ \text{MB} &= \mathop{\mathbb{E}}_{D_r, \tilde{s}, \theta}[f_\theta(x) - \mathop{\mathbb{E}}_{D_s|\theta}[g(x; D_s)]].\end{aligned} \tag{14}$$

$f(x) = \mathbb{E}_y[y]$ is the optimal predictor for real data, $\theta \sim p(\theta|D_r)$ is a single sample from the distribution of the generator parameters, $D_s \sim p(D_s|\theta)$ is a single sample of the synthetic data generating process, and f_θ is the optimal predictor for the synthetic data generating process with parameters θ. All random quantities are implicitly conditioned on x.

The takeaways from Theorem 2.3 are mostly the same as from Theorem 2.1, and the estimator from Section 2.3 also applies. The main difference is the DPVAR term in Theorem 2.3, which accounts for the added DP noise. As expected, the impact of DPVAR cannot be reduced with additional synthetic datasets.

[5] In DPVI, $\tilde{s}$ would be the variational approximation to the posterior.

3 EXPERIMENTS

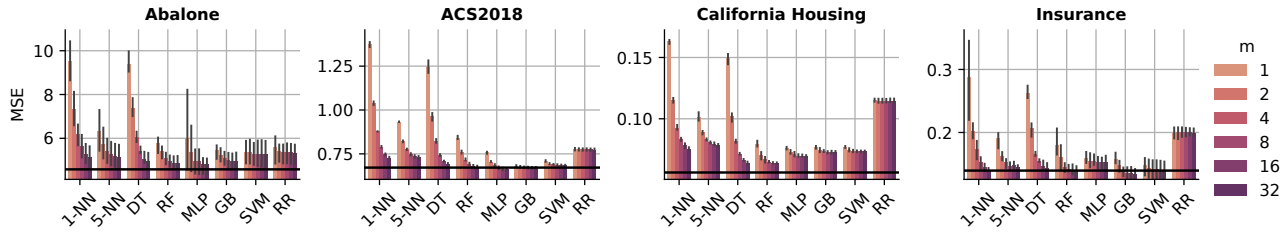
In this section, we describe our experiments. The common theme in all of them is generating synthetic data and evaluating the performance of several downstream prediction algorithms trained on the synthetic data. The performance evaluation uses a test set of real data, which is split from the whole dataset before generating synthetic data. Our code is available at `https://github.com/DPBayes/generative-ensemble-bias-variance-decomposition`.

The downstream algorithms we consider are nearest neighbours with 1 or 5 neighbours (1-NN and 5-NN), decision tree (DT), random forest (RF), gradient boosted trees (GB), a multilayer perceptron (MLP) and a support vector machine (SVM) for both classification and regression. We also use linear regression (LR) and ridge regression (RR) on regression tasks, and logistic regression (LogR) on classification tasks, though we omit linear regression from the main text plots, as its results are nearly identical to ridge regression. Decision trees and 1-NN have a very high variance, as both interpolate the training data with the hyperparameters we use. Linear, ridge, and logistic regression have fairly small variance in contrast. Appendix D contains more details on the experimental setup, including details on the datasets, and downstream algorithm hyperparameters.

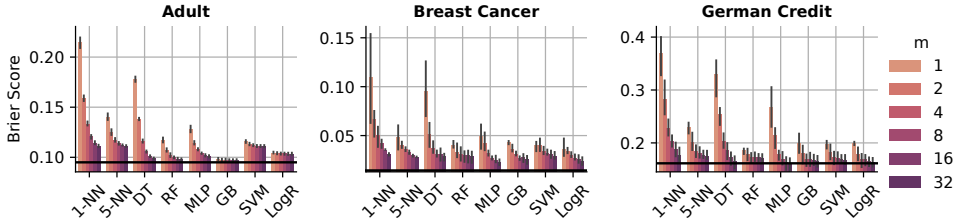
In each of our figures comparing prediction performance, we have included a horizontal black line showing the performance of the best downstream predictor without synthetic data. The downstream models include a random forest and gradient boosted trees, which are ensemble methods, so this line serves as a baseline for ensembles without synthetic data. Another relevant baseline is the performance of an ensemble with only one synthetic dataset, which is given by the results of random forests and gradient boosted trees with $m = 1$.

3.1 Effect of Multiple Synthetic Datasets

As our first experiment, we evaluate the performance of the synthetic data ensemble on 7 datasets, containing 4 regression and 3 classification tasks. See Appendix D.1 for details on the datasets. We use the synthetic data generators DDPM (Kotelnikov et al., 2023) and synthpop (Nowok et al., 2016), which we selected after a preliminary experiment described in Appendix E.1. We only plot the results from synthpop in the main text to save space, and defer the results of DDPM to Appendix E.2. We generate 32 synthetic datasets, of



(a) Regression datasets



(b) Classification datasets

Figure 1: MSE on regression datasets (a) or Brier score on classification datasets (b) of the ensemble of downstream predictors, with varying number of synthetic datasets m from synthpop. Increasing the number of synthetic datasets generally decreases both metrics, especially for decision trees and 1-NN. The predictors are nearest neighbours with 1 or 5 neighbours (1-NN and 5-NN), decision tree (DT), random forest (RF), a multilayer perceptron (MLP), gradient boosted trees (GB), a support vector machine (SVM), ridge regression (RR) and logistic regression (LogR). The black line is the MSE of the best predictor on real data. Tables S4 to S10 in the Appendix contain the numbers from the plots.

which between 1 and 32 are used to train the ensemble. The results are averaged over 3 runs with different train-test splits. We compute error bars as 95% confidence intervals obtained from bootstrapping over the repeats.

On the regression datasets, our error metric is MSE, which is the subject of Theorem 2.1. The results in Figure 1a show that a larger number of synthetic datasets generally decreases MSE. The decrease is especially clear with downstream algorithms that have a high variance like decision trees and 1-NN. Low-variance algorithms like ridge regression have very little if any decrease from multiple synthetic datasets. This is consistent with Theorem 2.1, where the number of synthetic datasets only affects the variance-related terms.

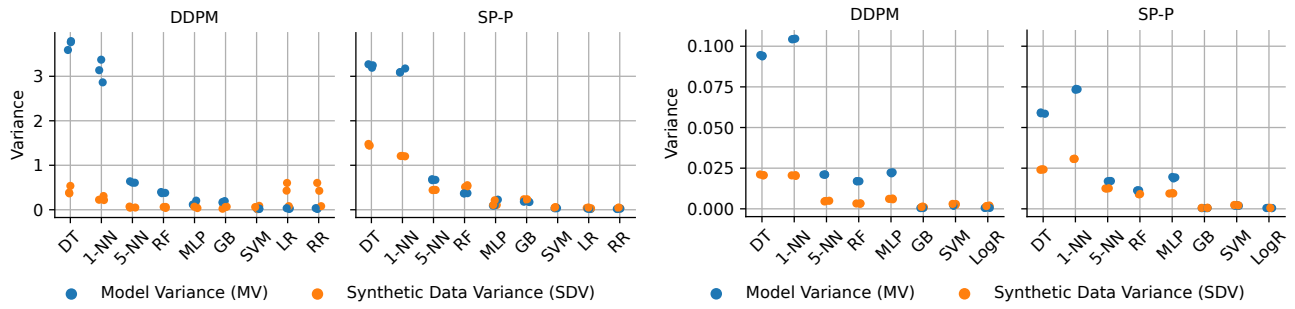
On the classification datasets, we consider 4 error metrics. Brier score (Brier, 1950) is MSE of the class probability predictions, so Theorem 2.1 applies to it. Cross entropy is a Bregman divergence, so Theorem C.2 from Appendix C applies to it. We also included accuracy and area under the ROC curve (AUC) even though our theory does not apply to them, as they are common and interpretable error metrics, so it is interesting to see how multiple synthetic datasets affect them. We use their complements in the plots, so that lower is better for all plotted metrics. We only present the Brier score results in the main text in Figure 1b, and defer the rest to Figures S3, S4 and S5 in Appendix E.2.

Because Theorem C.2 only applies to cross entropy when averaging log probabilities instead of probabilities, we compare both ways of averaging in the Appendix E.2, but only include probability averaging in the main text.

The results on the classification datasets in Figure 1b are similar to the regression experiment. A larger number of synthetic datasets generally decreases the score, especially for the high-variance models.

In Figure 2, we estimate the MV and SDV terms of the decomposition in Theorem 2.1. The results show that MV depends mostly on the downstream predictor, while SDV also depends on the synthetic data generator. The results also confirm our claims on the model variances: decision trees and 1-NN have a high variance, while linear, ridge and logistic regression have a low variance. See Appendix E.4, for details, and Figure S11 for results on all datasets.

In Appendix E.5 we compare an alternative to the ensemble of multiple synthetic datasets: generating a single large synthetic datasets with an equal number of datapoints as all the multiple synthetic datasets combined. One could expect generating a larger synthetic dataset to also reduce variances while saving on the computational cost of training multiple generative models. However, in the cases we examined in Figure S12, a single larger dataset gave at best a small improvement, and sometimes even increased the error. In contrast, adding more synthetic datasets often decreases error



(a) Abalone (b) Adult

Figure 2: Estimating the MV and SDV terms from the decomposition. Decision trees have high variances on all datasets, while linear, ridge and logistic regression have low variances. MV depends mostly on the predictor, while SDV depends on both the predictor and synthetic data generation algorithm. The points are the averages of estimated MV and SDV, averaged over the test data, from 3 repeats with different train-test splits. See Figure S11 in the Appendix for results on all datasets.

and never increases it. As a result, as a default choice, we recommend generating as many synthetic datasets as possible that have the same size as the original. We do not recommend making the synthetic datasets smaller, since the computational saving is likely small, and the additional generators that could be trained with the savings have diminishing returns due to the rule-of-thumb that m synthetic datasets give a $(1 - \frac{1}{m})$ fraction of the benefit.

3.2 Predicting Performance from Two Synthetic Datasets

Next, we evaluate the predictions our rule of thumb from Section 2.3 makes. To recap, our rule of thumb predicts that the maximal benefit from multiple synthetic datasets is $2(\text{MSE}_1 - \text{MSE}_2)$, and m synthetic datasets achieve a $1 - \frac{1}{m}$ fraction of this benefit, as shown in (10).

To evaluate the predictions from the rule, we estimate the MSE on regression tasks and Brier score on classification tasks for one and two synthetic datasets from the test set. The setup is otherwise identical to Section 3.1, and the train-test splits are the same. We plot the predictions from the rule, and compare them with the measured test errors with more than two synthetic datasets.

Figure 3 contains the results for the ACS 2018 datasets, and Figures S6 to S8 in the Appendix contain the results for the other datasets. The predictions are very accurate on ACS 2018, and reasonably accurate on the other datasets. The variance of the prediction depends heavily on the variance of the errors computed from the test data.

We also evaluated the rule on random forests without synthetic data, as it also applies to them. In this setting, the number of trees in the random forest is analogous to the number of synthetic datasets. We use the same datasets as in the previous experiments and use the same train-test splits of the real data. The results are in Figure S9 in the Appendix. The prediction is accurate when the test error is accurate, but can have high variance.

3.3 Differentially Private Synthetic Data

In this experiment, we evaluate the performance of the generative ensemble in the setting of Theorem 2.3, where m synthetic datasets are generated from a single noisy summary $\tilde{s}$ of the real data. We compare with splitting the privacy budget between m synthetic datasets.

The algorithms we use are AIM (McKenna et al., 2022), which needs to split the privacy budget between the m synthetic datasets, and NAPSU-MQ (Räisä et al., 2023b), which uses a single noisy summary. The dataset is the Adult dataset with a reduced set of columns, which is needed to keep the runtime of NAPSU-MQ reasonable. The downstream task is classification, so we use the same 4 metrics, and both probability and log-probability averaging, as in the non-DP classification experiment. We use fairly strict privacy parameters of $\epsilon = 1.5$, $\delta = n^{-2} \approx 4.7 \cdot 10^{-7}$. See Appendix D for the full details of the setup.

The results for Brier score are in Figure 4, and the results for all metrics are in Figure S10 in the Appendix. Increasing m often always improves the results, which is expected for NAPSU-MQ due to Theorem 2.3, but somewhat surprising for AIM, which splits the privacy budget. For AIM, $m = 4$ looks like a sensible default choice for the well-performing downstream predictors,

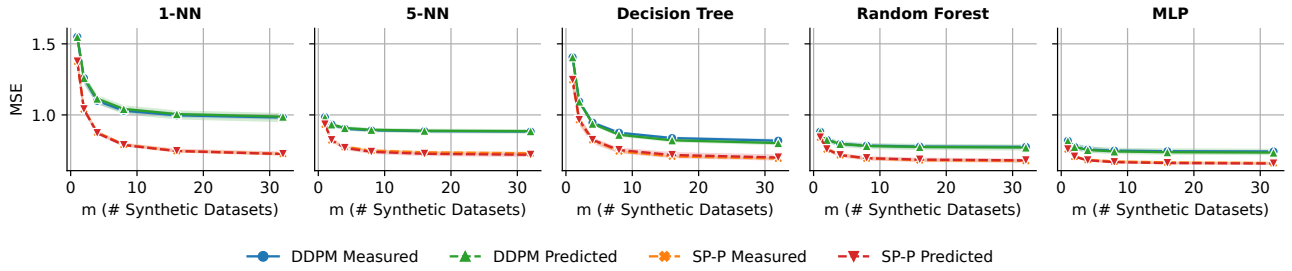


Figure 3: MSE or Brier score prediction on the ACS 2018 dataset. The predictions are very accurate on this dataset. The solid lines for DDPM and synthpop (SP-P) show the same error MSE or Brier score as Figure 1, while the dashed lines show predicted MSE or Brier score. 1-NN and 5-NN are nearest neighbours with 1 or 5 neighbours. We omitted downstream algorithms with uninteresting flat curves. See Figure S6 in the Appendix for the full figure, and Figures S7 and S8 for the other datasets. Tables S11 and S12 in the Appendix contain the numbers from the plots.

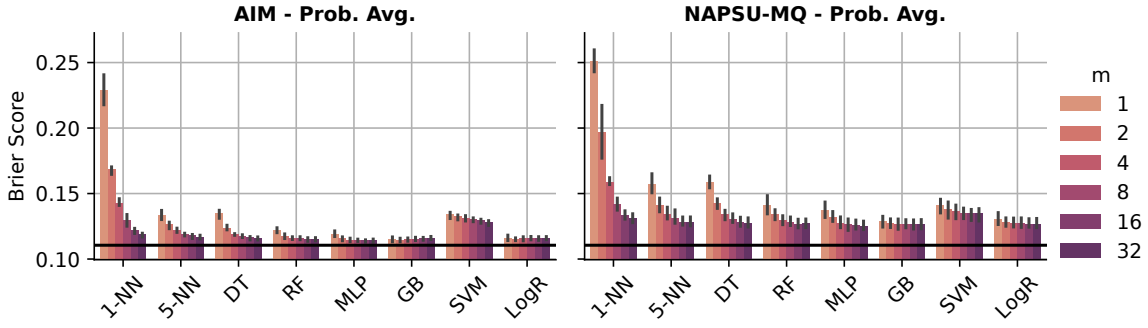


Figure 4: Brier score of the ensemble of downstream predictors with varying numbers of synthetic datasets m, generated with the DP methods AIM or NAPSU-MQ from the Adult dataset with a reduced set of features. Increasing the number of datasets generally decreases the score, even with AIM, which splits the privacy budget between m synthetic datasets. The privacy parameters are $\epsilon = 1.5$, $\delta = n^{-2} \approx 4.7 \cdot 10^{-7}$. The predictors are the same as in Figure 1. The black lines show the loss of the best non-DP downstream predictor trained on real data. Table S13 contains the numbers from the plots, and Figure S10 contains plots of the other error metrics.

since performance does not greatly change either way with $m > 4$. Once again, the high-variance predictors 1-NN, 5-NN and decision trees benefit the most from multiple synthetic datasets. AIM always outperforms NAPSU-MQ, which is likely a result of AIM being able to handle a more complicated set of input queries than NAPSU-MQ.

4 DISCUSSION

Limitations. Our main theory assumes that the synthetic datasets are generated i.i.d. given either the real data or a DP summary, which for example leaves out generative ensembles that explicitly encourage diversity between synthetic datasets in some way. We generalise the bias-variance decompositions to non-i.i.d. settings in Appendix B. The implications of the decompositions, like the rule-of-thumb on the number of synthetic datasets, do not apply in general if i.i.d. assumption is removed, but they can still apply with additional assumptions. We give an example in Appendix B.

Conclusion. We derived bias-variance decompositions for using synthetic data in several cases: for MSE or Brier score with i.i.d. synthetic datasets given the real data (Section 2.2) and MSE with i.i.d. synthetic datasets given a DP summary of the real data (Section 2.4). We generalised these decompositions to non-i.i.d. synthetic datasets (Appendix B) and Bregman divergences (Appendix C). The decompositions make actionable predictions, such as yielding a simple rule of thumb that can be used to select the number of synthetic datasets. We empirically examined the performance of generative ensembles on several real datasets and downstream predictors, and found that the predictions of the theory generally hold in practice (Section 3). These findings significantly increase the theoretical understanding of generative ensembles, which is very limited in prior literature.

Acknowledgments

This work was supported by the Research Council of Finland (Flagship programme: Finnish Center for Ar-

tificial Intelligence, FCAI as well as Grants 356499 and 359111), the Strategic Research Council at the Research Council of Finland (Grant 358247) as well as the European Union (Project 101070617). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the granting authority can be held responsible for them. The authors wish to thank the Finnish Computing Competence Infrastructure (FCCI) for supporting this project with computational and data storage resources.

References

Alaa, A., Breugel, B. V., Saveliev, E. S., and van der Schaar, M. How Faithful is your Synthetic Data? Sample-level Metrics for Evaluating and Auditing Generative Models. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 290–306. PMLR, 2022.

Antoniou, A., Storkey, A., and Edwards, H. Data Augmentation Generative Adversarial Networks, arXiv:1711.04340, 2018. `http://arxiv.org/abs/1711.04340`.

Aydore, S., Brown, W., Kearns, M., Kenthapadi, K., Melis, L., Roth, A., and Siva, A. A. Differentially Private Query Release Through Adaptive Projection. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 457–467. PMLR, 2021.

Bregman, L. M. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200–217, 1967.

Breiman, L. Bagging predictors. *Machine Learning*, 24 (2):123–140, 1996.

Breiman, L. Random Forests. *Machine Learning*, 45 (1):5–32, 2001.

Brier, G. W. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

Bun, M. and Steinke, T. Concentrated Differential Privacy: Simplifications, Extensions, and Lower Bounds. In *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part I*, pp. 635–658, 2016.

Chen, D., Orekondy, T., and Fritz, M. GS-WGAN: A gradient-sanitized approach for learning differentially private generators. In *Advances in Neural Information Processing Systems*, volume 33, pp. 12673–12684, 2020.

Chen, M., Quan, Y., Xu, Y., and Ji, H. Self-Supervised Blind Image Deconvolution via Deep Generative Ensemble Learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(2):634–647, 2023.

Choi, H., Jang, E., and Alemi, A. A. WAIC, but Why? Generative Ensembles for Robust Anomaly Detection, arXiv:1810.01392, 2019. `http://arxiv.org/abs/1810.01392`.

Das, H. P., Tran, R., Singh, J., Yue, X., Tison, G., Sangiovanni-Vincentelli, A., and Spanos, C. J. Conditional Synthetic Data Generation for Robust Machine Learning Applications with Limited Pandemic Data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 11792–11800, 2022.

Ding, F., Hardt, M., Miller, J., and Schmidt, L. Retiring Adult: New Datasets for Fair Machine Learning. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Dwork, C. and Roth, A. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. D. Calibrating Noise to Sensitivity in Private Data Analysis. In *Third Theory of Cryptography Conference*, volume 3876 of *Lecture Notes in Computer Science*, pp. 265–284. Springer, 2006.

Geman, S., Bienenstock, E., and Doursat, R. Neural Networks and the Bias/Variance Dilemma. *Neural Computation*, 4(1):1–58, 1992.

Gneiting, T. and Raftery, A. E. Strictly Proper Scoring Rules, Prediction, and Estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

Gupta, N., Smith, J., Adlam, B., and Mariet, Z. E. Ensembles of Classifiers: A Bias-Variance Perspective. *Transactions on Machine Learning Research*, 2022.

Harder, F., Adamczewski, K., and Park, M. DP-MERF: Differentially Private Mean Embeddings with Random Features for Practical Privacy-preserving Data Generation. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 1819–1827. PMLR, 2021.

Hardt, M., Ligett, K., and McSherry, F. A Simple and Practical Algorithm for Differentially Private Data Release. In *Advances in Neural Information Processing Systems*, volume 25, pp. 2348–2356, 2012.

Hofmann, H. Statlog (German credit data). UCI Machine Learning Repository, 1994. https://archive.ics.uci.edu/dataset/144/statlog+german+credit+data.

Jälkö, J., Dikmen, O., and Honkela, A. Differentially Private Variational Inference for Non-conjugate Models. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2017.

Jälkö, J., Lagerspetz, E., Haukka, J., Tarkoma, S., Honkela, A., and Kaski, S. Privacy-preserving data sharing via probabilistic modeling. *Patterns*, 2(7): 100271, 2021.

James, G. M. Variance and Bias for General Loss Functions. *Machine Learning*, 51(2):115–135, 2003.

Kimpara, D., Frongillo, R., and Waggoner, B. Proper Losses for Discrete Generative Models. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Kohavi, R. and Becker, B. Adult. UCI Machine Learning Repository, 1996. https://archive.ics.uci.edu/dataset/2/adult.

Kotelnikov, A., Baranchuk, D., Rubachev, I., and Babenko, A. TabDDPM: Modelling Tabular Data with Diffusion Models. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Liew, C. K., Choi, U. J., and Liew, C. J. A data distortion by probability distribution. *ACM Transactions on Database Systems*, 10(3):395–411, 1985.

Liu, T., Vietri, G., and Wu, S. Z. Iterative Methods for Private Synthetic Data: Unifying Framework and New Methods. In *Advances in Neural Information Processing Systems*, volume 34, pp. 690–702, 2021.

Luzi, L., Balestriero, R., and Baraniuk, R. G. Ensembles of Generative Adversarial Networks for Disconnected Data, arXiv:2006.14600, 2020. http://arxiv.org/abs/2006.14600.

McKenna, R., Sheldon, D., and Miklau, G. Graphical-model based estimation and inference for differential privacy. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4435–4444. PMLR, 2019.

McKenna, R., Miklau, G., and Sheldon, D. Winning the NIST Contest: A scalable and general approach to differentially private synthetic data. *Journal of Privacy and Confidentiality*, 11(3), 2021.

McKenna, R., Mullins, B., Sheldon, D., and Miklau, G. AIM: An adaptive and iterative mechanism for differentially private synthetic data. *Proceedings of the VLDB Endowment*, 15(11):2599–2612, 2022.

Meeus, M., Guépin, F., Cretu, A.-M., and de Montjoye, Y.-A. Achilles' Heels: Vulnerable Record Identification in Synthetic Data Publishing. In *28th European Symposium on Research in Computer Security*, 2023.

Naeem, M. F., Oh, S. J., Uh, Y., Choi, Y., and Yoo, J. Reliable Fidelity and Diversity Metrics for Generative Models. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 7176–7185. PMLR, 2020.

Nash, W., Sellers, T., Talbot, S., Cawthorn, A., and Ford, W. Abalone. UCI Machine Learning Repository, 1995. https://archive.ics.uci.edu/dataset/1/abalone.

Nowok, B., Raab, G. M., and Dibben, C. Synthpop: Bespoke Creation of Synthetic Data in R. *Journal of Statistical Software*, 74:1–26, 2016.

Pfau, D. A generalized bias-variance decomposition for Bregman divergences. 2013. http://www.davidpfau.com/assets/generalized_bvd_proof.pdf.

Raghunathan, T. E., Reiter, J. P., and Rubin, D. B. Multiple imputation for statistical disclosure limitation. *Journal of Official Statistics*, 19(1):1, 2003.

Räisä, O., Jälkö, J., and Honkela, A. On Consistent Bayesian Inference from Synthetic Data, arXiv:2305.16795, 2023a. http://arxiv.org/abs/2305.16795.

Räisä, O., Jälkö, J., Kaski, S., and Honkela, A. Noise-aware statistical inference with differentially private synthetic data. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 3620–3643. PMLR, 2023b.

Rubin, D. B. Discussion: Statistical disclosure limitation. *Journal of Official Statistics*, 9(2):461–468, 1993.

Stadler, T., Oprisanu, B., and Troncoso, C. Synthetic Data – Anonymisation Groundhog Day. In *31st USENIX Security Symposium (USENIX Security 22)*, pp. 1451–1468, 2022.

Ueda, N. and Nakano, R. Generalization error of ensemble estimators. In *Proceedings of International Conference on Neural Networks (ICNN'96)*, volume 1, pp. 90–95 vol.1, 1996.

van Breugel, B., Kyono, T., Berrevoets, J., and van der Schaar, M. DECAF: Generating Fair Synthetic Data Using Causally-Aware Generative Networks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 22221–22233, 2021.

van Breugel, B., Qian, Z., and van der Schaar, M. Synthetic data, real errors: How (not) to publish and use synthetic data. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 34793–34808. PMLR, 2023a.

van Breugel, B., Seedat, N., Imrie, F., and van der Schaar, M. Can You Rely on Your Model Evaluation? Improving Model Evaluation with Synthetic Test Data. In *Thirty-Seventh Conference on Neural Information Processing Systems*, 2023b.

van Breugel, B., Sun, H., Qian, Z., and van der Schaar, M. Membership Inference Attacks against Synthetic Data through Overfitting Detection. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, pp. 3493–3514. PMLR, 2023c.

Wang, Y., Zhang, L., and van de Weijer, J. Ensembles of Generative Adversarial Networks, arXiv:1612.00991, 2016. `http://arxiv.org/abs/1612.00991`.

Wilson, A. G. Deep Ensembles as Approximate Bayesian Inference, 2021. `https://cims.nyu.edu/~andrewgw/deepensembles/`.

Wilson, A. G. and Izmailov, P. Bayesian Deep Learning and a Probabilistic Perspective of Generalization. In *Advances in Neural Information Processing Systems*, volume 33, pp. 4697–4708, 2020.

Wolberg, W., Mangasarian, O., Street, N., and Street, W. Breast cancer Wisconsin (diagnostic). UCI Machine Learning Repository, 1995. `https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic`.

Wood, D., Mu, T., Webb, A. M., Reeve, H. W. J., Lujan, M., and Brown, G. A Unified Theory of Diversity in Ensemble Learning. *Journal of Machine Learning Research*, 24(359):1–49, 2023.

Xu, L., Skoularidou, M., Cuesta-Infante, A., and Veeramachaneni, K. Modeling Tabular data using Conditional GAN. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Not Applicable]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Yes]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes]
   (d) Information about consent from data providers/curators. [No] None of the providers of the datasets we used provide information about consent.
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A MISSING PROOFS

Theorem 2.1. *Let the parameters for m generators $\theta_i \sim p(\theta|D_r)$, $i = 1, \ldots, m$, be i.i.d. Let the synthetic datasets be $D_s^i \sim p(D_s|\theta_i)$ independently, and let $\hat{g}(x; D_s^{1:m}) = \frac{1}{m}\sum_{i=1}^m g(x; D_s^i)$. Then the mean-squared error in predicting y from x decomposes into six terms: model variance (MV), synthetic data variance (SDV), real data variance (RDV), synthetic data bias (SDB), model bias (MB), and noise $\operatorname{Var}_y[y]$:*

$$\text{MSE} = \frac{1}{m}\text{MV} + \frac{1}{m}\text{SDV} + \text{RDV} + (\text{SDB} + \text{MB})^2 + \operatorname*{Var}_y[y], \tag{3}$$

where

$$\begin{aligned}
\text{MSE} &= \operatorname*{\mathbb{E}}_{y, D_r, D_s^{1:m}}[(y - \hat{g}(x; D_s^{1:m}))^2] \\
\text{MV} &= \operatorname*{\mathbb{E}}_{D_r,\theta} \operatorname*{Var}_{D_s|\theta}[g(x; D_s)] \\
\text{SDV} &= \operatorname*{\mathbb{E}}_{D_r} \operatorname*{Var}_{\theta|D_r} \operatorname*{\mathbb{E}}_{D_s|\theta}[g(x; D_s)] \\
\text{RDV} &= \operatorname*{Var}_{D_r} \operatorname*{\mathbb{E}}_{D_s|D_r}[g(x; D_s)] \\
\text{SDB} &= \operatorname*{\mathbb{E}}_{D_r}[f(x) - \operatorname*{\mathbb{E}}_{\theta|D_r}[f_\theta(x)]] \\
\text{MB} &= \operatorname*{\mathbb{E}}_{D_r,\theta}[f_\theta(x) - \operatorname*{\mathbb{E}}_{D_s|\theta}[g(x; D_s)]].
\end{aligned} \tag{4}$$

$f(x) = \mathbb{E}_y[y]$ is the optimal predictor for real data, $\theta \sim p(\theta|D_r)$ is a single sample from the identical distribution of the generator parameters θ_i, $D_s \sim p(D_s|\theta)$ is a single sample of the synthetic data generating process given θ, and f_θ is the optimal predictor for the synthetic data generating process with parameters θ. All random quantities are implicitly conditioned on x.

Proof. With m synthetic datasets $D_s^{1:m}$ and model $\hat{g}(x, D_s^{1:m})$ that combines the synthetic datasets, the classical bias-variance decomposition gives

$$\operatorname*{\mathbb{E}}_{y, D_r, D_s^{1:m}}[(y - \hat{g}(x; D_s^{1:m}))^2] = (f(x) - \operatorname*{\mathbb{E}}_{D_r, D_s^{1:m}}[\hat{g}(x; D_s^{1:m})])^2 + \operatorname*{Var}_{D_r, D_s^{1:m}}[\hat{g}(x; D_s^{1:m})] + \operatorname*{Var}_y[y]. \tag{15}$$

Using the independence of the synthetic datasets, these can be decomposed further:

$$\operatorname*{\mathbb{E}}_{D_r, D_s^{1:m}}[\hat{g}(x; D_s^{1:m})] = \operatorname*{\mathbb{E}}_{D_r} \operatorname*{\mathbb{E}}_{D_s^{1:m}|D_r}\left[\frac{1}{m}\sum_{i=1}^m g(x, D_s^i)\right] = \operatorname*{\mathbb{E}}_{D_r} \operatorname*{\mathbb{E}}_{D_s|D_r}[g(x, D_s)] = \operatorname*{\mathbb{E}}_{D_r, D_s}[g(x, D_s)], \tag{16}$$

$$\begin{aligned}
\operatorname*{Var}_{D_r, D_s^{1:m}}[\hat{g}(x; D_s^{1:m})] &= \operatorname*{\mathbb{E}}_{D_r} \operatorname*{Var}_{D_s^{1:m}|D_r}\left[\frac{1}{m}\sum_{i=1}^m g(x; D_s^i)\right] + \operatorname*{Var}_{D_r} \operatorname*{\mathbb{E}}_{D_s^{1:m}|D_r}\left[\frac{1}{m}\sum_{i=1}^m g(x; D_s^i)\right] \\
&= \frac{1}{m^2} \operatorname*{\mathbb{E}}_{D_r} \operatorname*{Var}_{D_s^{1:m}|D_r}\left[\sum_{i=1}^m g(x; D_s^i)\right] + \operatorname*{Var}_{D_r} \operatorname*{\mathbb{E}}_{D_s|D_r}[g(x; D_s)] \\
&= \frac{1}{m} \operatorname*{\mathbb{E}}_{D_r} \operatorname*{Var}_{D_s|D_r}[g(x; D_s)] + \operatorname*{Var}_{D_r} \operatorname*{\mathbb{E}}_{D_s|D_r}[g(x; D_s)],
\end{aligned} \tag{17}$$

and

$$\operatorname*{Var}_{D_s|D_r}[g(x; D_s)] = \operatorname*{\mathbb{E}}_{\theta|D_r} \operatorname*{Var}_{D_s|\theta}[g(x; D_s)] + \operatorname*{Var}_{\theta|D_r} \operatorname*{\mathbb{E}}_{D_s|\theta}[g(x, D_s)]. \tag{18}$$

The bias can be decomposed with $f_\theta(x)$:

$$\begin{aligned}
f(x) - \operatorname*{\mathbb{E}}_{D_r, D_s}[g(x; D_s)] &= \operatorname*{\mathbb{E}}_{D_r,\theta}[f(x) - f_\theta(x) + f_\theta(x) - \operatorname*{\mathbb{E}}_{D_s|\theta}[g(x; D_s)]] \\
&= \operatorname*{\mathbb{E}}_{D_r}[f(x) - \operatorname*{\mathbb{E}}_{\theta|D_r}[f_\theta(x)]] + \operatorname*{\mathbb{E}}_{D_r,\theta}[f_\theta(x) - \operatorname*{\mathbb{E}}_{D_s|\theta}[g(x; D_s)]]
\end{aligned} \tag{19}$$

Combining all of these gives the claim. □

Theorem 2.3. *Let the parameters for m generators $\theta_i \sim p(\theta|\tilde{s})$, $i = 1, \dots, m$, be i.i.d. given a DP summary $\tilde{s}$. Let the synthetic datasets be $D_s^i \sim p(D_s|\theta_i)$ independently, and let $\hat{g}(x; D_s^{1:m}) = \frac{1}{m}\sum_{i=1}^m g(x; D_s^i)$. Then*

$$\begin{aligned}\mathrm{MSE} = \frac{1}{m}\mathrm{MV} + \frac{1}{m}\mathrm{SDV} + \mathrm{RDV} + \mathrm{DPVAR} \\ + (\mathrm{SDB} + \mathrm{MB})^2 + \operatorname*{Var}_y[y],\end{aligned} \tag{12}$$

where

$$\begin{aligned}\mathrm{MSE} &= \mathop{\mathbb{E}}_{y, D_r, \tilde{s}, D_s^{1:m}}[(y - \hat{g}(x; D_s^{1:m}))^2] \\ \mathrm{MV} &= \mathop{\mathbb{E}}_{D_r, \tilde{s}, \theta} \mathop{\mathrm{Var}}_{D_s|\theta}[g(x; D_s)] \\ \mathrm{SDV} &= \mathop{\mathbb{E}}_{D_r, \tilde{s}} \mathop{\mathrm{Var}}_{\theta|\tilde{s}} \mathop{\mathbb{E}}_{D_s|\theta}[g(x; D_s)] \\ \mathrm{RDV} &= \mathop{\mathrm{Var}}_{D_r} \mathop{\mathbb{E}}_{D_s|D_r}[g(x; D_s)]\end{aligned} \tag{13}$$

$$\begin{aligned}\mathrm{DPVAR} &= \mathop{\mathbb{E}}_{D_r} \mathop{\mathrm{Var}}_{\tilde{s}|D_r} \mathop{\mathbb{E}}_{D_s|\tilde{s}}[g(x; D_s)] \\ \mathrm{SDB} &= \mathop{\mathbb{E}}_{D_r, \tilde{s}}[f(x) - \mathop{\mathbb{E}}_{\theta|\tilde{s}}[f_\theta(x)]] \\ \mathrm{MB} &= \mathop{\mathbb{E}}_{D_r, \tilde{s}, \theta}[f_\theta(x) - \mathop{\mathbb{E}}_{D_s|\theta}[g(x; D_s)]].\end{aligned} \tag{14}$$

$f(x) = \mathbb{E}_y[y]$ is the optimal predictor for real data, $\theta \sim p(\theta|D_r)$ is a single sample from the distribution of the generator parameters, $D_s \sim p(D_s|\theta)$ is a single sample of the synthetic data generating process, and f_θ is the optimal predictor for the synthetic data generating process with parameters θ. All random quantities are implicitly conditioned on x.

Proof. Using $\tilde{s}$ in place of the real data in Theorem 2.1 gives

$$\mathrm{MSE} = \frac{1}{m}\mathrm{MV} + \frac{1}{m}\mathrm{SDV} + \mathop{\mathrm{Var}}_{D_r, \tilde{s}} \mathop{\mathbb{E}}_{D_s|\tilde{s}}[g(x; D_s)] + (\mathrm{SDB} + \mathrm{MB})^2 + \operatorname*{Var}_y[y], \tag{20}$$

where

$$\mathrm{MSE} = \mathop{\mathbb{E}}_{y, D_r, \tilde{s}, D_s^{1:m}}[(y - \hat{g}(x; D_s^{1:m}))^2] \tag{21}$$

$$\mathrm{MV} = \mathop{\mathbb{E}}_{D_r, \tilde{s}, \theta} \mathop{\mathrm{Var}}_{D_s|\theta}[g(x; D_s)] \tag{22}$$

$$\mathrm{SDV} = \mathop{\mathbb{E}}_{D_r, \tilde{s}} \mathop{\mathrm{Var}}_{\theta|\tilde{s}} \mathop{\mathbb{E}}_{D_s|\theta}[g(x; D_s)] \tag{23}$$

$$\mathrm{SDB} = \mathop{\mathbb{E}}_{D_r, \tilde{s}}[f(x) - \mathop{\mathbb{E}}_{\theta|\tilde{s}}[f_\theta(x)]] \tag{24}$$

$$\mathrm{MB} = \mathop{\mathbb{E}}_{D_r, \tilde{s}, \theta}[f_\theta(x) - \mathop{\mathbb{E}}_{D_s|\theta}[g(x; D_s)]]. \tag{25}$$

We can additionally decompose

$$\begin{aligned}\mathop{\mathrm{Var}}_{D_r, \tilde{s}} \mathop{\mathbb{E}}_{D_s|\tilde{s}}[g(x; D_s)] &= \mathop{\mathbb{E}}_{D_r} \mathop{\mathrm{Var}}_{\tilde{s}|D_r} \mathop{\mathbb{E}}_{D_s|\tilde{s}}[g(x; D_s)] + \mathop{\mathrm{Var}}_{D_r} \mathop{\mathbb{E}}_{\tilde{s}|D_r} \mathop{\mathbb{E}}_{D_s|\tilde{s}}[g(x; D_s)] \\ &= \mathop{\mathbb{E}}_{D_r} \mathop{\mathrm{Var}}_{\tilde{s}|D_r} \mathop{\mathbb{E}}_{D_s|\tilde{s}}[g(x; D_s)] + \mathop{\mathrm{Var}}_{D_r} \mathop{\mathbb{E}}_{D_s|D_r}[g(x; D_s)]\end{aligned} \tag{26}$$

This reveals the DP-related variance term

$$\mathrm{DPVAR} = \mathop{\mathbb{E}}_{D_r} \mathop{\mathrm{Var}}_{\tilde{s}|D_r} \mathop{\mathbb{E}}_{D_s|\tilde{s}}[g(x; D_s)] \tag{27}$$

so we have

$$\mathrm{MSE} = \frac{1}{m}\mathrm{MV} + \frac{1}{m}\mathrm{SDV} + \mathrm{RDV} + \mathrm{DPVAR} + (\mathrm{SDB} + \mathrm{MB})^2 + \operatorname*{Var}_y[y]. \tag{28}$$

□

B NON-I.I.D. SYNTHETIC DATA

Here, we consider the case of non-i.i.d synthetic datasets, and allow each synthetic dataset to have a different predictor g_i. We get a similar decomposition as in Theorem 2.1, but the terms of the decomposition are now averages over the possibly different synthetic data distributions, and there is an additional covariance term.

Theorem B.1. *Let the parameters for m generators $\theta_i \sim p(\theta_i|D_r)$, $i = 1, \dots, m$, be potentially non-i.i.d. given the real data D_r. Let the synthetic datasets be $D_s^i \sim p(D_s^i|\theta_i)$, and let $\hat{g}(x; D_s^{1:m}) = \frac{1}{m}\sum_{i=1}^m g_i(x; D_s^i)$. Then*

$$\text{MSE} = \frac{1}{m}\overline{\text{MV}} + \frac{1}{m}\overline{\text{SDV}} + \text{COV} + \overline{\text{RDV}} + (\overline{\text{SDB}} + \overline{\text{MB}})^2 + \operatorname*{Var}_y[y], \tag{29}$$

where

$$\text{MSE} = \mathop{\mathbb{E}}_{y, D_r, D_s^{1:m}}[(y - \hat{g}(x; D_s^{1:m}))^2] \tag{30}$$

$$\overline{\text{MV}} = \frac{1}{m}\sum_{i=1}^m \mathop{\mathbb{E}}_{D_r, \theta_i} \operatorname*{Var}_{D_s^i|\theta_i}[g_i(x; D_s^i)] \tag{31}$$

$$\overline{\text{SDV}} = \frac{1}{m}\sum_{i=1}^m \mathop{\mathbb{E}}_{D_r} \operatorname*{Var}_{\theta_i|D_r} \mathop{\mathbb{E}}_{D_s^i|\theta_i}[g_i(x; D_s^i)] \tag{32}$$

$$\text{COV} = \frac{1}{m^2}\sum_{i \neq j} \mathop{\mathbb{E}}_{D_r}\left[\operatorname*{Cov}_{D_s^i, D_s^j|D_r}[g_i(x; D_s^i), g_j(x; D_s^j)]\right] \tag{33}$$

$$\overline{\text{RDV}} = \operatorname*{Var}_{D_r}\left[\frac{1}{m}\sum_{i=1}^m \mathop{\mathbb{E}}_{D_s^i|D_r}[g_i(x; D_s^i)]\right] \tag{34}$$

$$\overline{\text{SDB}} = \frac{1}{m}\sum_{i=1}^m \mathop{\mathbb{E}}_{D_r}\left[f(x) - \mathop{\mathbb{E}}_{\theta_i|D_r}[f_{\theta_i}(x)]\right] \tag{35}$$

$$\overline{\text{MB}} = \frac{1}{m}\sum_{i=1}^m \mathop{\mathbb{E}}_{D_r, \theta_i}\left[f_{\theta_i}(x) - \mathop{\mathbb{E}}_{D_s^i|\theta_i}[g_i(x, D_s^i)]\right]. \tag{36}$$

$f(x) = \mathbb{E}_y[y]$ is the optimal predictor for real data and f_{θ_i} is the optimal predictor for the synthetic data generating process given parameters θ_i. All random quantities are implicitly conditioned on x.

Proof. The classical bias-variance decomposition gives

$$\mathop{\mathbb{E}}_{y, D_r, D_s^{1:m}}[(y - \hat{g}(x; D_s^{1:m}))^2] = (f(x) - \mathop{\mathbb{E}}_{D_r, D_s^{1:m}}[\hat{g}(x; D_s^{1:m})])^2 + \operatorname*{Var}_{D_r, D_s^{1:m}}[\hat{g}(x; D_s^{1:m})] + \operatorname*{Var}_y[y]. \tag{37}$$

For the bias,

$$\begin{aligned}
&f(x) - \mathop{\mathbb{E}}_{D_r, D_s^{1:m}}[\hat{g}(x; D_s^{1:m})] \\
&= f(x) - \mathop{\mathbb{E}}_{D_r} \mathop{\mathbb{E}}_{D_s^{1:m}|D_r} \left[\frac{1}{m}\sum_{i=1}^{m} g_i(x, D_s^i)\right] \\
&= \mathop{\mathbb{E}}_{D_r} \mathop{\mathbb{E}}_{D_s^{1:m}|D_r} \left[\frac{1}{m}\sum_{i=1}^{m} (f(x) - g_i(x, D_s^i))\right] \\
&= \mathop{\mathbb{E}}_{D_r} \mathop{\mathbb{E}}_{D_s^{1:m}|D_r} \left[\frac{1}{m}\sum_{i=1}^{m} (f(x) - f_{\theta_i}(x) + f_{\theta_i}(x) - g_i(x, D_s^i))\right] \\
&= \mathop{\mathbb{E}}_{D_r} \mathop{\mathbb{E}}_{\theta_{1:m}|D_r} \left[\frac{1}{m}\sum_{i=1}^{m} (f(x) - f_{\theta_i}(x))\right] + \mathop{\mathbb{E}}_{D_r} \mathop{\mathbb{E}}_{D_s^{1:m}|D_r} \left[\frac{1}{m}\sum_{i=1}^{m} (f_{\theta_i}(x) - g_i(x, D_s^i))\right] \\
&= \mathop{\mathbb{E}}_{D_r} \left[\frac{1}{m}\sum_{i=1}^{m} (f(x) - \mathop{\mathbb{E}}_{\theta_i|D_r}[f_{\theta_i}(x)])\right] + \mathop{\mathbb{E}}_{D_r, \theta_{1:m}} \left[\frac{1}{m}\sum_{i=1}^{m} (f_{\theta_i}(x) - \mathop{\mathbb{E}}_{D_s^i|\theta_i}[g_i(x, D_s^i)])\right] \\
&= \frac{1}{m}\sum_{i=1}^{m} \mathop{\mathbb{E}}_{D_r} \left[f(x) - \mathop{\mathbb{E}}_{\theta_i|D_r}[f_{\theta_i}(x)]\right] + \frac{1}{m}\sum_{i=1}^{m} \mathop{\mathbb{E}}_{D_r, \theta_i} \left[f_{\theta_i}(x) - \mathop{\mathbb{E}}_{D_s^i|\theta_i}[g_i(x, D_s^i)]\right].
\end{aligned} \tag{38}$$

For the variance,

$$\begin{aligned}
\mathop{\mathrm{Var}}_{D_r, D_s^{1:m}}[\hat{g}(x; D_s^{1:m})] &= \mathop{\mathbb{E}}_{D_r} \mathop{\mathrm{Var}}_{D_s^{1:m}|D_r} \left[\frac{1}{m}\sum_{i=1}^{m} g_i(x; D_s^i)\right] + \mathop{\mathrm{Var}}_{D_r} \mathop{\mathbb{E}}_{D_s^{1:m}|D_r} \left[\frac{1}{m}\sum_{i=1}^{m} g_i(x; D_s^i)\right] \\
&= \frac{1}{m^2} \mathop{\mathbb{E}}_{D_r} \mathop{\mathrm{Var}}_{D_s^{1:m}|D_r} \left[\sum_{i=1}^{m} g_i(x; D_s^i)\right] + \mathop{\mathrm{Var}}_{D_r} \left[\frac{1}{m}\sum_{i=1}^{m} \mathop{\mathbb{E}}_{D_s^i|D_r}[g_i(x; D_s^i)]\right],
\end{aligned} \tag{39}$$

$$\mathop{\mathrm{Var}}_{D_s^{1:m}|D_r} \left[\sum_{i=1}^{m} g_i(x; D_s^i)\right] = \sum_{i=1}^{m} \mathop{\mathrm{Var}}_{D_s^i|D_r}[g_i(x; D_s^i)] + \sum_{i\neq j} \mathop{\mathrm{Cov}}_{D_s^i, D_s^j|D_r}[g_i(x; D_s^i), g_j(x; D_s^j)] \tag{40}$$

and

$$\mathop{\mathrm{Var}}_{D_s^i|D_r}[g_i(x; D_s^i)] = \mathop{\mathbb{E}}_{\theta_i|D_r} \mathop{\mathrm{Var}}_{D_s^i|\theta_i}[g_i(x; D_s^i)] + \mathop{\mathrm{Var}}_{\theta_i|D_r} \mathop{\mathbb{E}}_{D_s^i|\theta}[g_i(x, D_s^i)]. \tag{41}$$

□

If the synthetic datasets are identically distributed, but not necessarily independent, and the predictors are identical, Theorem B.1 simplifies to

$$\mathrm{MSE} = \frac{1}{m}\mathrm{MV} + \frac{1}{m}\mathrm{SDV} + \left(1 - \frac{1}{m}\right)\mathrm{COV} + \mathrm{RDV} + (\mathrm{SDB} + \mathrm{MB})^2 + \mathop{\mathrm{Var}}_{y}[y], \tag{42}$$

where

$$\begin{aligned}
\mathrm{MSE} &= \mathop{\mathbb{E}}_{y, D_r, D_s^{1:m}}[(y - \hat{g}(x; D_s^{1:m}))^2] \\
\mathrm{MV} &= \mathop{\mathbb{E}}_{D_r, \theta} \mathop{\mathrm{Var}}_{D_s|\theta}[g(x; D_s)] \\
\mathrm{SDV} &= \mathop{\mathbb{E}}_{D_r} \mathop{\mathrm{Var}}_{\theta|D_r} \mathop{\mathbb{E}}_{D_s|\theta}[g(x; D_s)] \\
\mathrm{COV} &= \mathop{\mathbb{E}}_{D_r} \left[\mathop{\mathrm{Cov}}_{D_s^1, D_s^2|D_r}[g(x; D_s^1), g(x; D_s^2)]\right] \\
\mathrm{RDV} &= \mathop{\mathrm{Var}}_{D_r} \mathop{\mathbb{E}}_{D_s|D_r}[g(x; D_s)] \\
\mathrm{SDB} &= \mathop{\mathbb{E}}_{D_r}[f(x) - \mathop{\mathbb{E}}_{\theta|D_r}[f_\theta(x)]] \\
\mathrm{MB} &= \mathop{\mathbb{E}}_{D_r, \theta}[f_\theta(x) - \mathop{\mathbb{E}}_{D_s|\theta}[g(x; D_s)]].
\end{aligned} \tag{43}$$

which is Theorem 2.1 with the additional covariance term.

In the noisy summary case, we get an analogue of Theorem 2.3.

Theorem B.2. *Let the parameters for m generators $\theta_i \sim p(\theta_i|\tilde{s})$, $i = 1, \ldots, m$, be potentially non-i.i.d. given a DP summary $\tilde{s}$ the real data D_r, let the synthetic datasets be $D_s^i \sim p(D_s^i|\theta_i)$, and let $\hat{g}(x; D_s^{1:m}) = \frac{1}{m}\sum_{i=1}^m g_i(x; D_s^i)$. Then*

$$\text{MSE} = \frac{1}{m}\overline{\text{MV}} + \frac{1}{m}\overline{\text{SDV}} + \text{COV} + \overline{\text{RDV}} + \overline{\text{DPVAR}} + (\overline{\text{SDB}} + \overline{\text{MB}})^2 + \underset{y}{\text{Var}}[y], \tag{44}$$

where

$$\text{MSE} = \underset{y, D_r, \tilde{s}, D_s^{1:m}}{\mathbb{E}}[(y - \hat{g}(x; D_s^{1:m}))^2] \tag{45}$$

$$\overline{\text{MV}} = \frac{1}{m}\sum_{i=1}^m \underset{D_r, \tilde{s}, \theta_i}{\mathbb{E}} \underset{D_s^i|\theta_i}{\text{Var}}[g_i(x; D_s^i)] \tag{46}$$

$$\overline{\text{SDV}} = \frac{1}{m}\sum_{i=1}^m \underset{D_r, \tilde{s}}{\mathbb{E}} \underset{\theta_i|\tilde{s}}{\text{Var}} \underset{D_s^i|\theta_i}{\mathbb{E}}[g_i(x; D_s^i)] \tag{47}$$

$$\text{COV} = \frac{1}{m^2}\sum_{i \neq j} \underset{D_r, \tilde{s}}{\mathbb{E}}\left[\underset{D_s^i, D_s^j|\tilde{s}}{\text{Cov}}[g_i(x; D_s^i), g_j(x; D_s^j)]\right] \tag{48}$$

$$\overline{\text{RDV}} = \underset{D_r}{\text{Var}}\left[\frac{1}{m}\sum_{i=1}^m \underset{D_s^i|D_r}{\mathbb{E}}[g_i(x; D_s^i)]\right] \tag{49}$$

$$\overline{\text{DPVAR}} = \underset{D_r}{\mathbb{E}} \underset{\tilde{s}|D_r}{\text{Var}}\left[\frac{1}{m}\sum_{i=1}^m \underset{D_s^i|\tilde{s}}{\mathbb{E}}[g_i(x; D_s^i)]\right] \tag{50}$$

$$\overline{\text{SDB}} = \frac{1}{m}\sum_{i=1}^m \underset{D_r, \tilde{s}}{\mathbb{E}}\left[f(x) - \underset{\theta_i|\tilde{s}}{\mathbb{E}}[f_{\theta_i}(x)]\right] \tag{51}$$

$$\overline{\text{MB}} = \frac{1}{m}\sum_{i=1}^m \underset{D_r, \tilde{s}, \theta_i}{\mathbb{E}}\left[f_{\theta_i}(x) - \underset{D_s^i|\theta_i}{\mathbb{E}}[g_i(x, D_s^i)]\right]. \tag{52}$$

$f(x) = \mathbb{E}_y[y]$ is the optimal predictor for real data and f_{θ_i} is the optimal predictor for the synthetic data generating process given parameters θ_i. All random quantities are implicitly conditioned on x.

Proof. Using $\tilde{s}$ in place of D_r in Theorem B.1 gives

$$\text{MSE} = \frac{1}{m}\overline{\text{MV}} + \frac{1}{m}\overline{\text{SDV}} + \text{COV} + \underset{D_r, \tilde{s}}{\text{Var}}\left[\frac{1}{m}\sum_{i=1}^m \underset{D_s^i|\tilde{s}}{\mathbb{E}}[g_i(x; D_s^i)]\right] + (\overline{\text{SDB}} + \overline{\text{MB}})^2 + \underset{y}{\text{Var}}[y]. \tag{53}$$

The variance over D_r and $\tilde{s}$ can be decomposed

$$\begin{aligned}
&\underset{D_r, \tilde{s}}{\text{Var}}\left[\frac{1}{m}\sum_{i=1}^m \underset{D_s^i|\tilde{s}}{\mathbb{E}}[g_i(x; D_s^i)]\right] \\
&= \underset{D_r}{\mathbb{E}} \underset{\tilde{s}|D_r}{\text{Var}}\left[\frac{1}{m}\sum_{i=1}^m \underset{D_s^i|\tilde{s}}{\mathbb{E}}[g_i(x; D_s^i)]\right] + \underset{D_r}{\text{Var}} \underset{\tilde{s}|D_r}{\mathbb{E}}\left[\frac{1}{m}\sum_{i=1}^m \underset{D_s^i|\tilde{s}}{\mathbb{E}}[g_i(x; D_s^i)]\right] \\
&= \underset{D_r}{\mathbb{E}} \underset{\tilde{s}|D_r}{\text{Var}}\left[\frac{1}{m}\sum_{i=1}^m \underset{D_s^i|\tilde{s}}{\mathbb{E}}[g_i(x; D_s^i)]\right] + \underset{D_r}{\text{Var}}\left[\frac{1}{m}\sum_{i=1}^m \underset{D_s^i|D_r}{\mathbb{E}}[g_i(x; D_s^i)]\right].
\end{aligned} \tag{54}$$

□

Implication of Non-i.i.d. Theory From Theorems B.1 and B.2, it is clear that the implications of the i.i.d. theory do not always hold. For example, when increasing the number of synthetic datasets with each generator

being a different model, all of the terms in Theorems B.1 and B.2 that are averages over the generators can change.

However, we can recover some implications with additional assumptions. For example, if we assume that the generators and downstream predictors are always the same, possibly correlated, but the covariance term COV does depend on the number of synthetic datasets, we can derive a similar MSE prediction rule as in Section 2.3 from Equation (42).

C BIAS-VARIANCE DECOMPOSITION FOR BREGMAN DIVERGENCES

C.1 Background: Bregman Divergences

A Bregman divergence (Bregman, 1967) $D_F \colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a loss function

$$D_F(y, g) = F(y) - F(g) - \nabla F(g)^T (y - g) \tag{55}$$

where $F \colon \mathbb{R}^d \to \mathbb{R}$ is a strictly convex differentiable function. Many common error metrics, like MSE and cross entropy, can be expressed as expected values of a Bregman divergence. In fact, proper scoring rules[6] can be characterised via Bregman divergences (Gneiting & Raftery, 2007; Kimpara et al., 2023). Table S1 shows how the metrics we consider are expressed as Bregman divergences (Gupta et al., 2022).

Pfau (2013) derive the following bias-variance decomposition for Bregman divergences:

$$\underbrace{\mathbb{E}[D(y, g)]}_{\text{Error}} = \underbrace{\mathbb{E}[D(y, \mathbb{E}\, y)]}_{\text{Noise}} + \underbrace{D(\mathbb{E}\, y, \mathcal{E} g)}_{\text{Bias}} + \underbrace{\mathbb{E}\, D(\mathcal{E} g, g)}_{\text{Variance}} \tag{56}$$

y is the true value, and g is the predicted value. All of the random quantities are conditioned on x. $\mathcal{E}$ is a *central prediction*:

$$\mathcal{E} g = \operatorname*{argmin}_z \mathbb{E}\, D(z, g). \tag{57}$$

The variance term can be used to define a generalisation of variance:

$$\mathcal{V} g = \mathbb{E}\, D(\mathcal{E} g, g) \tag{58}$$

$\mathcal{E}$ and $\mathcal{V}$ can also be defined conditionally on some random variable Z by making the expectations conditional on Z in the definitions. These obey generalised laws of total expectation and variance (Gupta et al., 2022):

$$\mathcal{E} g = \mathcal{E}_Z[\mathcal{E}_{g|Z}[g]] \tag{59}$$

and

$$\mathcal{V} g = \mathop{\mathbb{E}}_Z[\mathcal{V}_{g|Z}[g]] + \mathcal{V}_Z[\mathcal{E}_{g|Z}[g]]. \tag{60}$$

The convex dual of g is $g^* = \nabla F(g)$. The central prediction $\mathcal{E} g$ can also be expressed as an expectation over the convex dual (Gupta et al., 2022):

$$\mathcal{E} g = (\mathbb{E}\, g^*)^* \tag{61}$$

Gupta et al. (2022) study the bias-variance decomposition of Bregman divergence on a generic ensemble. They show that if the ensemble aggregates prediction by averaging them, bias is not preserved, and can increase. As a solution, they consider *dual averaging*, that is

$$\hat{g} = \left(\frac{1}{m} \sum_{i=1}^{m} g_i^* \right)^* \tag{62}$$

for models $g_1, \ldots, g_m$ forming the ensemble $\hat{g}$. They show that the bias is preserved in the dual averaged ensemble, and derive a bias-variance decomposition for them. For mean squared error, the dual average is simply the standard average, but for cross entropy, it corresponds to averaging log probabilities.

[6] Proper scoring rules are error metrics that are minimised by predicting the correct probabilities.

Table S1: Common error metrics as Bregman divergences. g denotes a prediction in regression and p denotes predicted class probabilities in classification. $g^{(j)}$ and $p^{(j)}$ denote the predictions of different ensemble members. y is the correct value in regression, and a one-hot encoding of the correct class in classification. The binary classification Brier score only looks at probabilities for one class. If the multiclass Brier score is used with two classes, it is twice the binary Brier score.

Error Metric	D_F	$F(t)$	Dual Average
MSE	$(y-g)^2$	t^2	$\frac{1}{m}\sum_{j=1}^m g^{(j)}$
Brier Score (2 classes)	$(y_0-p_0)^2$	t^2	$\frac{1}{m}\sum_{j=1}^m p^{(j)}$
Brier Score (Multiclass)	$\sum_i (y_i-p_i)^2$	$\sum_i t_i^2$	$\frac{1}{m}\sum_{j=1}^m p^{(j)}$
Cross Entropy	$-\sum_i y_i \ln p_i$	$\sum_i t_i \ln t_i$	$\operatorname{softmax}\left(\frac{1}{m}\sum_{j=1}^m \ln p^{(j)}\right)$

C.2 Bregman Divergence Decomposition for Synthetic Data

We extend the Bregman divergence decomposition for ensembles from Gupta et al. (2022) to generative ensembles. To prove Theorem C.2, we use the following lemma.

Lemma C.1 (Gupta et al. 2022, Proposition 5.3)**.** *Let $X_1, \dots, X_m$ be i.i.d. random variables and let $\hat{X} = (\sum_{i=1}^m X_i^*)^*$ be their dual average. Then $\mathcal{E}\hat{X} = \mathcal{E}X$, $\mathcal{V}\hat{X} \leq \mathcal{V}X$ and for any independent Y, $D(\mathbb{E}\,Y, \mathcal{E}\hat{X}) = D(\mathbb{E}\,Y, \mathcal{E}X)$.*

Theorem C.2. *When the synthetic datasets $D_s^{1:m}$ are i.i.d. given the real data D_r and $\hat{g}(x; D_s^{1:m}) = (\frac{1}{m}\sum_{i=1}^m g(x; D_s^i)^*)^*$,*

$$\text{Error} \leq \text{MV} + \text{SDV} + \text{RDV} + \text{Bias} + \text{Noise} \tag{63}$$

where

$$\text{Error} = \mathop{\mathbb{E}}_{y, D_r, D_s^{1:m}}[D(y, \hat{g})] \tag{64}$$

$$\text{MV} = \mathop{\mathbb{E}}_{D_r} \mathop{\mathbb{E}}_{\theta|D_r} \mathcal{V}_{D_s|\theta}[g] \tag{65}$$

$$\text{SDV} = \mathop{\mathbb{E}}_{D_r} \mathcal{V}_{\theta|D_r} \mathcal{E}_{D_s|\theta}[g] \tag{66}$$

$$\text{RDV} = \mathcal{V}_{D_r} \mathcal{E}_{D_s|D_r}[g] \tag{67}$$

$$\text{Bias} = D\left(\mathop{\mathbb{E}}_{y} y, \mathcal{E}_{D_r}\mathcal{E}_{D_s|D_r}[g]\right) \tag{68}$$

$$\text{Noise} = \mathop{\mathbb{E}}_{y}\left[D(y, \mathop{\mathbb{E}}_{y} y)\right] \tag{69}$$

Proof. Plugging the ensemble $\hat{g}$ into the decomposition (56) gives

$$\mathop{\mathbb{E}}_{y, D_r, D_s^{1:m}}[D(y, \hat{g})] = \mathop{\mathbb{E}}_{y}[D(y, \mathop{\mathbb{E}}_{y} y)] + D(\mathop{\mathbb{E}}_{y} y, \mathcal{E}_{D_r, D_s^{1:m}} \hat{g}) + \mathcal{V}_{D_r, D_s^{1:m}}[\hat{g}] \tag{70}$$

Applying the generalised laws of expectation and variance, and Lemma C.1 to the variance term, we obtain:

$$\mathcal{V}_{D_r, D_s^{1:m}}[\hat{g}] = \mathop{\mathbb{E}}_{D_r} \mathcal{V}_{D_s^{1:m}|D_r}[\hat{g}] + \mathcal{V}_{D_r}\mathcal{E}_{D_s^{1:m}|D_r}[\hat{g}] \tag{71}$$

For the second term on the right:

$$\mathcal{E}_{D_s^{1:m}|D_r}[\hat{g}] = \mathcal{E}_{D_s|D_r}[g], \tag{72}$$

which gives the RDV:

$$\mathcal{V}_{D_r}\mathcal{E}_{D_s^{1:m}|D_r}[\hat{g}] = \mathcal{V}_{D_r}\mathcal{E}_{D_s|D_r}[g]. \tag{73}$$

For the first term on the right:

$$\mathop{\mathbb{E}}_{D_r} \mathcal{V}_{D_s^{1:m}|D_r}[\hat{g}] \leq \mathop{\mathbb{E}}_{D_r} \mathcal{V}_{D_s|D_r}[g] \tag{74}$$

Table S2: Details on the datasets used in the experiments. # Cat. and # Num. are the numbers of categorical and numerical features, not counting the target variable. For datasets with removed rows, the table shows the number of rows after the removals.

Dataset	# Rows	# Cat.	# Num.	Task
Abalone	4177	1	7	Regression
ACS 2018	50000	5	2	Regression
Adult	45222	8	4	Classification
Reduced Adult	46043	7	2	Classification
Breast Cancer	569	0	30	Classification
California Housing	20622	0	8	Regression
German Credit	1000	13	7	Classification
Insurance	1338	3	3	Regression

and

$$\mathcal{V}_{D_s|D_r}[g] = \mathop{\mathbb{E}}_{\theta|D_r} \mathcal{V}_{D_s|\theta}[g] + \mathcal{V}_{\theta|D_r}\mathcal{E}_{D_s|\theta}[g], \tag{75}$$

which give MV and SDV.

For the bias

$$\begin{aligned} D(\mathop{\mathbb{E}}_{y} y, \mathcal{E}_{D_r, D_s^{1:m}}[\hat{g}]) &= D(\mathop{\mathbb{E}}_{y} y, \mathcal{E}_{D_r}\mathcal{E}_{D_s^{1:m}|D_r}[\hat{g}]) \\ &= D(\mathop{\mathbb{E}}_{y} y, \mathcal{E}_{D_r}\mathcal{E}_{D_s|D_r}[g]). \end{aligned} \tag{76}$$

Putting everything together proves the claim. □

This decomposition is not as informative as the other two in Section 2, as is only gives an upper bound, and does not explicitly depend on the number of synthetic datasets.

Implication for Non-Synthetic Data Ensembles The theory of Gupta et al. (2022) assumes that each member of the ensemble is trained with an independently sampled real dataset. In practice, this would mean that one needs to split the training data between each ensemble member to apply their theory, so their theory does not apply to any of the ways ensembles are usually trained. In contrast, our theory applies to bagging, as discussed in Section 2.1, so our Theorem C.2 implies a generalisation of Proposition 5.3 of Gupta et al. (2022) to bagging.

D EXPERIMENTAL DETAILS

D.1 Datasets

In our experiments, we use 7 tabular datasets. For four of them, the downstream prediction task is regression, and for the other three, the prediction task is binary classification. Table S2 lists some general information on the datasets. We use 25% of the real data as a test set, with the remaining 75% being used to generate the synthetic data, for all of the datasets. All experiments are repeated several times, with different randomly selected train-test splits for each repeat.

Abalone (Nash et al., 1995, CC BY 4.0) The abalone dataset contains information on abalones, with the task of predicting the number of rings on the abalone from other information like weight and size.

ACS 2018 (`https://www.census.gov/programs-surveys/acs/microdata/documentation.2018.html`, license: `https://www.census.gov/data/developers/about/terms-of-service.html`) This dataset contains several variables from the American community survey (ACS) of 2018, with the task of predicting a person's income from the other features. Specifically, the variables we selected are AGEP (age), COW (employer type), SCHL (education), MAR (marital status), WKHP (working hours), SEX, RAC1P (race), and the target PINCP (income). We take a subset of 50000 datapoints from the California data, and log-transform the target variable. We used the folktables package (Ding et al., 2021) to download the given subset of the data.

Adult (Kohavi & Becker, 1996, CC BY 4.0) The UCI Adult dataset contains general information on people, with the task of predicting whether their income is over $50000. We drop rows with any missing values.

Reduced Adult (Kohavi & Becker, 1996, CC BY 4.0) This dataset is the UCI Adult dataset with a reduced set of features. The subset of features is age, workclass, education, marital-status, race, gender, capital-gain, capital-loss and hours-per-week.[7] We binarise capital-gain and capital-loss to indicate whether the original value is positive or not for both synthetic data generation and downstream prediction. We discretise age and hours-per-week to 5 categories for synthetic data generation, and converted back to continuous values for the downstream prediction, as our differentially private synthetic data generators require discrete data. We only remove rows with missing values in the included columns, so the number of rows is larger with the reduced set of features.

Breast Cancer (Wolberg et al., 1995, CC BY 4.0) The breast cancer dataset contains features derived from images of potential tumors, with the task of predicting whether the potential tumor is benign or malignant.

California Housing (`https://scikit-learn.org/stable/datasets/real_world.html#california-housing-dataset`, license unknown) The california housing dataset contains information on housing districts, specifically census block groups, in California. The task is predicting the median house value in the district. We removed outlier rows where the average number of rooms is at least 50, or the average occupancy is at least 30. According to the dataset description, these likely correspond to districts with many empty houses. We log-transformed the target variable, as well as the population and median income features.

German Credit (Hofmann, 1994, CC BY 4.0) The German credit dataset contains information on a bank's customers, with the task of predicting whether the customers are "good" or "bad".

Insurance (`https://www.kaggle.com/datasets/mirichoi0218/insurance/data`, Database Contents License (DbCL) v1.0) The insurance dataset contains general information on people, like age, gender and BMI, as well as the amount they charged their medical insurance, which is the variable to predict. We take a log transform of the target variable before generating synthetic data.

D.2 Downstream Prediction Algorithms

We use the scikit-learn[8] implementations of all of the downstream algorithms, which includes probability predictions for all algorithms on the classification tasks. We standardise the data before training for all downstream algorithms except the tree-based algorithms, specifically decision tree, random forest, and gradient boosted trees. This standardisation is done just before downstream training, so the input to the synthetic data generation algorithms is not standardised. We use the default hyperparameters of scikit-learn for all downstream algorithms except MLP, where we increased the maximum number of iterations to 1000, as the default was not enough to converge on some datasets. In particular, this means that decision trees are trained to interpolate the training data, resulting in high variance of the predictions.

D.3 DP Experiment

Both AIM (McKenna et al., 2022) and NAPSU-MQ (Räisä et al., 2023b) generate synthetic data based on noisy values of marginal queries on the real data, which count how many rows of the data have given values for given variables. AIM includes a mechanism that chooses a subset of queries to measure under DP from a potentially very large workload. We set the workload to be all marginal queries over two variables. NAPSU-MQ does not include a query selection mechanism, so we first run AIM with $\epsilon = 0.5$, $\delta = \frac{1}{2}n^{-2}$ to select a subset of marginals and then run NAPSU-MQ with the selected marginals with $\epsilon = 1$, $\delta = \frac{1}{2}n^{-2}$, which results in the same privacy bounds ($\epsilon = 1.5$, $\delta = n^{-2}$; Dwork & Roth, 2014) that we used for AIM. We split the privacy budget between the m synthetic datasets in AIM by dividing the zero-concentrated DP (Bun & Steinke, 2016) parameter that AIM uses internally by m, which keeps the total privacy budget fixed.

[7] This subset was used in the NAPSU-MQ experiments of Räisä et al. (2023b)

[8] `https://scikit-learn.org/stable/index.html`, BSD 3-Clause License

We used the implementation of the authors for AIM[9], and used the Twinify library[10] for NAPSU-MQ. We used the default hyperparameters for AIM when generating synthetic data. For selecting the queries for NAPSU-MQ, we set the maximum junction tree size[11] hyperparameter to 0.001 MiB to ensure that NAPSU-MQ does not run for an unreasonable amount of time with the selected queries. The default is 80 MiB, so the synthetic data generated by AIM is based on a much more comprehensive set of queries than the synthetic data from NAPSU-MQ. For posterior inference in NAPSU-MQ, we used MCMC with 1000 kept samples, 500 warmup samples, and 2 chains.

D.4 Computational Resources

We ran the synthetic data generation algorithms DDPM, TVAE and CTGAN on a single GPU, synthpop, AIM and NAPSU-MQ on CPU, and all downstream analysis on CPU, all in a cluster environment.

E EXTRA RESULTS

E.1 Synthetic Data Generation Algorithms

We compare several synthetic data generation algorithms to see which algorithms are most interesting for subsequent experiments. We use the California housing dataset, where the downstream task is regression. The algorithms we compare are DDPM (Kotelnikov et al., 2023), TVAE (Xu et al., 2019), CTGAN (Xu et al., 2019) and synthpop (Nowok et al., 2016). DDPM, TVAE and CTGAN are a diffusion model, a variational autoencoder and a GAN that are designed for tabular data. We use the implementations from the synthcity library[12] for these. Synthpop generates synthetic data by sampling one column from the real data, and generating the other columns by sequentially training a predictive model on the real data, and predicting the next column from the already generated ones. We use the implementation from the authors (Nowok et al., 2016).[13]

We use the default hyperparameters for all of the algorithms. Synthpop and DDPM have a setting that could potentially affect the randomness in the synthetic data generation, so we include both possibilities for these settings in this experiment. For synthpop, this setting is whether the synthetic data is generated from a Bayesian posterior predictive distribution, which synthpop calls "proper" synthetic data. For DDPM, this setting is whether the loss function is MSE or KL divergence. In the plots, the two variants of synthpop are called "SP-P" and "SP-IP" for the proper and improper variants, and the variants of DDPM are "DDPM" and "DDPM-KL".

The results of the comparison are shown in Figure S1. We see that synthpop and DDPM with MSE loss generally outperform the other generation algorithms, so we select them for the subsequent experiments. There is very little difference between the two variants of synthpop, so we choose the "proper" variant due to its connection with the Bayesian reasoning for using multiple synthetic datasets.

[9] `https://github.com/ryan112358/private-pgm/blob/master/mechanisms/aim.py`, Apache-2.0 license

[10] `https://github.com/DPBayes/twinify`, Apache-2.0 license

[11] The junction tree size of the selected queries determines how difficult the selected queries are the probabilistic graphical model algorithms NAPSU-MQ and AIM use. Their runtime is roughly linear in the junction tree size (McKenna et al., 2022).

[12] `https://github.com/vanderschaarlab/synthcity`, Apache-2.0 license

[13] License: GPL-3

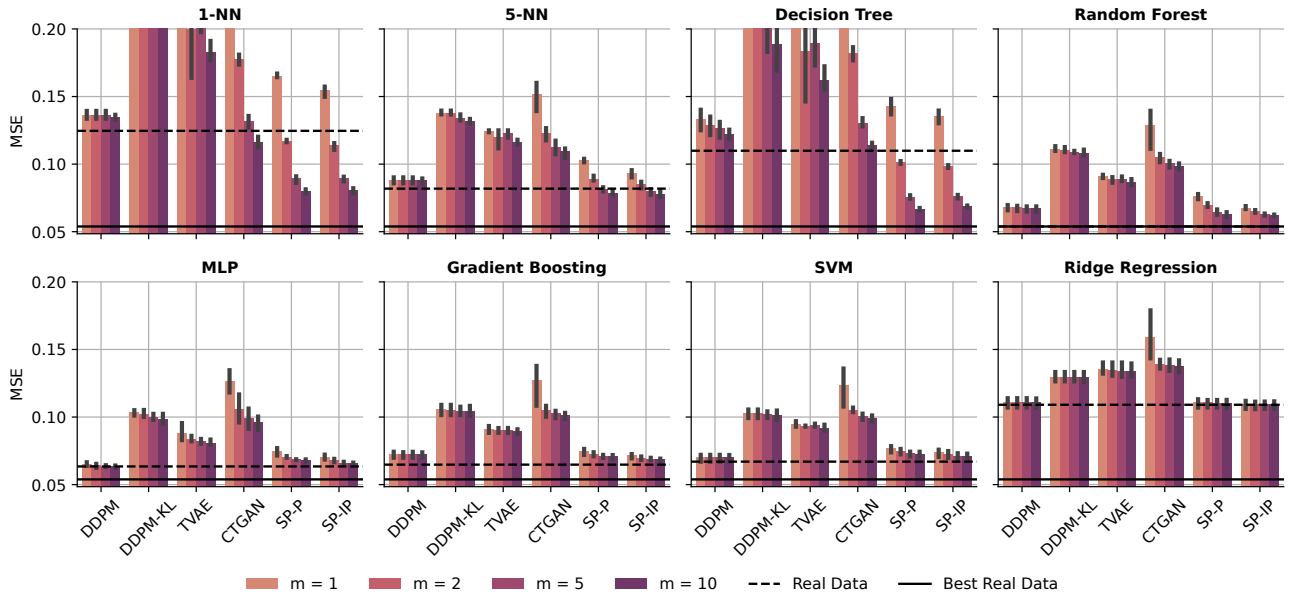


Figure S1: Comparison of synthetic data generation algorithms for several prediction algorithms on the California housing dataset, with 1 to 10 synthetic datasets. DDPM and synthpop achieve smaller MSE in the downstream predictions, so they were selected for further experiments. SP-P and SP-IP are the proper and improper variants of synthpop, and DDPM-KL is DDPM with KL divergence loss. 1-NN and 5-NN are nearest neighbours with 1 and 5 neighbours. The dashed black lines show the performance of each prediction algorithm on the real data, and the solid black line shows the performance of the best predictor, random forest, on the real data. The results are averaged over 3 repeats, with different train-test splits. The error bars are 95% confidence intervals formed by bootstrapping over the repeats. Linear regression was omitted, as it had nearly identical results as ridge regression. Table S3 in the Appendix contains the numbers in the plots, including ridge regression.

E.2 Extra Plots

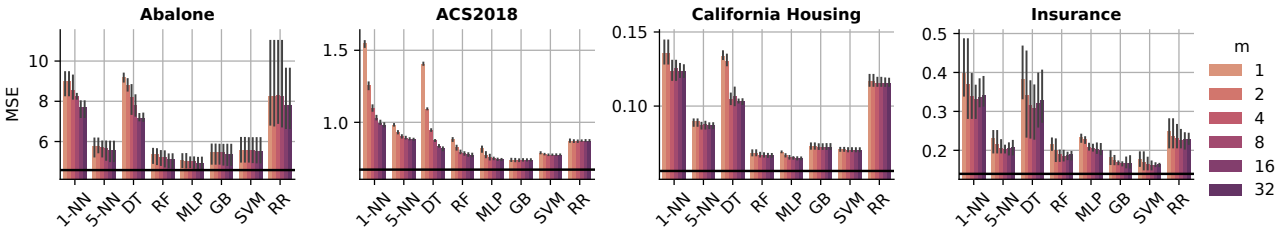


Figure S2: MSE on regression datasets of the ensemble of downstream predictors, with varying number of synthetic datasets m from DDPM. Increasing the number of synthetic datasets generally decreases MSE, especially for decision trees and 1-NN. The predictors are nearest neighbours with 1 or 5 neighbours (1-NN and 5-NN), decision tree (DT), random forest (RF), a multilayer perceptron (MLP), gradient boosted trees (GB), a support vector machine (SVM) and ridge regression (RR). The black line is the MSE of the best predictor on real data. The results are averaged over 3 repeats. The error bars are 95% confidence intervals formed by bootstrapping over the repeats. We omitted linear regression from the plots, as it had almost identical results to ridge regression. Tables S4 to S7 contain the numbers from the plots, including linear regression.

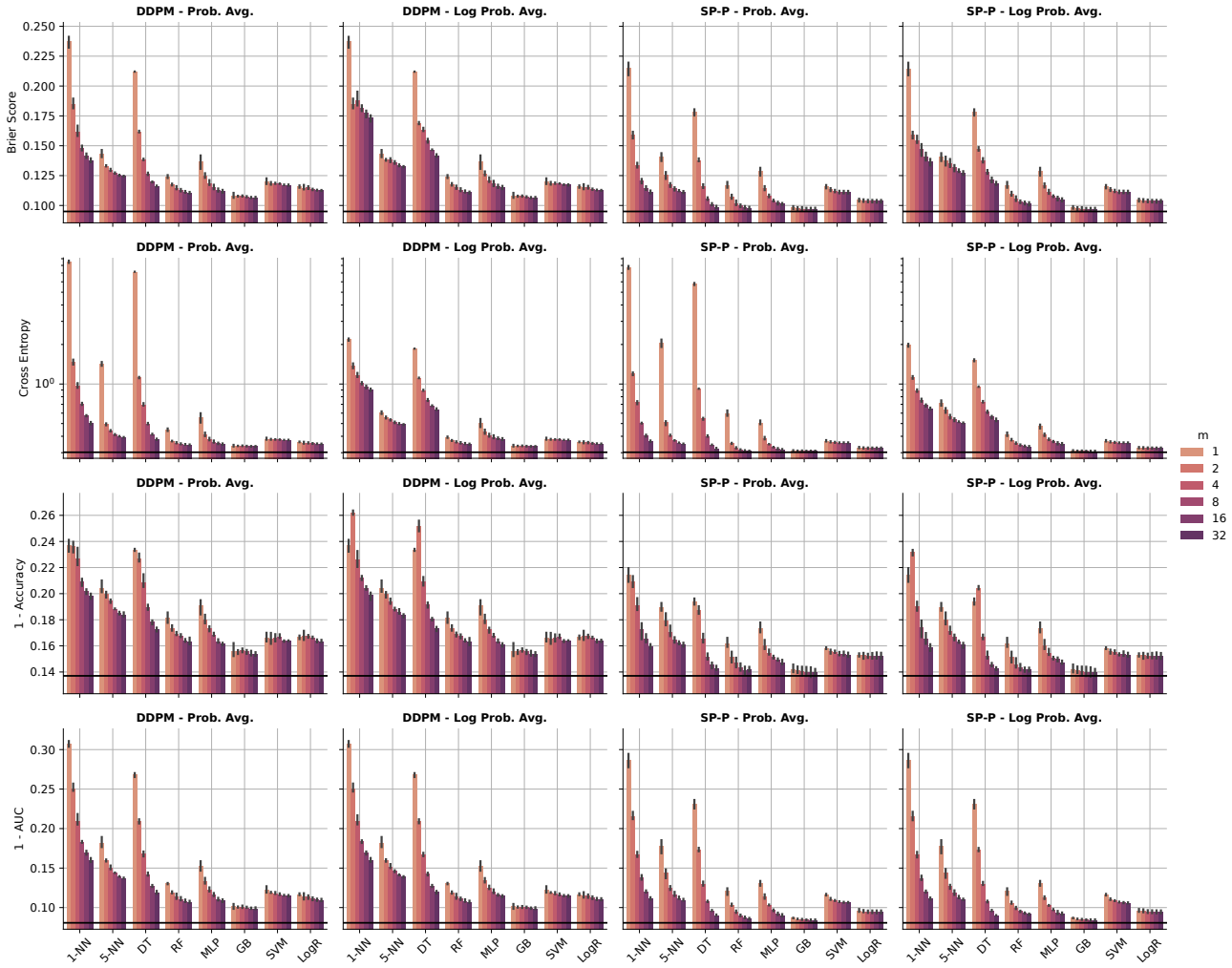


Figure S3: All error metrics on the Adult dataset. Note the logarithmic scale on the cross entropy y-axis. The predictors are nearest neighbours with 1 or 5 neighbours (1-NN and 5-NN), decision tree (DT), random forest (RF), a multilayer perceptron (MLP), gradient boosted trees (GB), a support vector machine (SVM) and logistic regression (LogR). The black line show the loss of the best downstream predictor trained on real data. The results are averaged over 3 repeats with different train-test splits. The error bars are 95% confidence intervals formed by bootstrapping over the repeats.

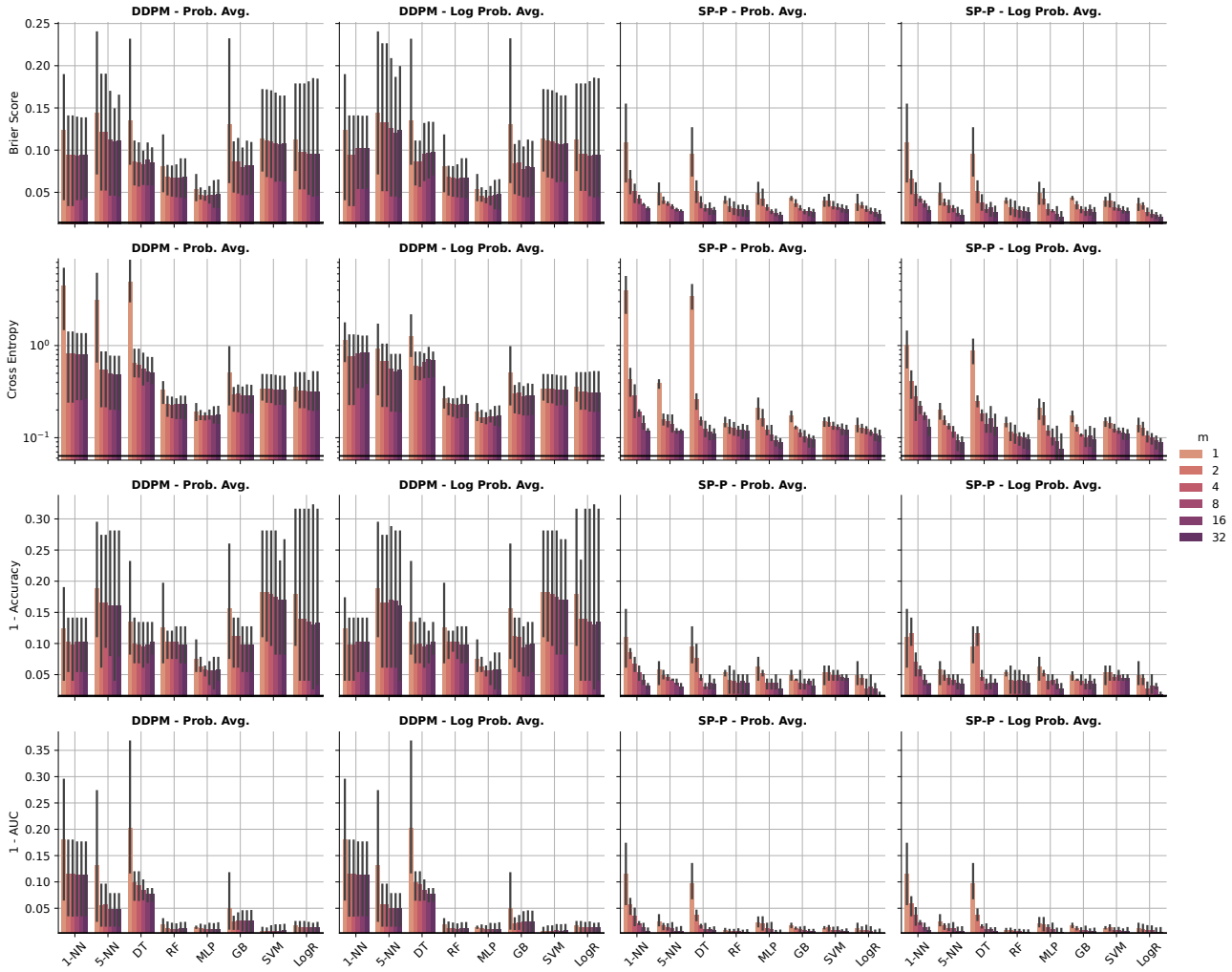


Figure S4: All error metrics on the breast cancer dataset. Note the he logarithmic scale on the cross entropy y-axis. The predictors are nearest neighbours with 1 or 5 neighbours (1-NN and 5-NN), decision tree (DT), random forest (RF), a multilayer perceptron (MLP), gradient boosted trees (GB), a support vector machine (SVM) and logistic regression (LogR). The black line show the loss of the best downstream predictor trained on real data. The results are averaged over 3 repeats with different train-test splits. The error bars are 95% confidence intervals formed by bootstrapping over the repeats.

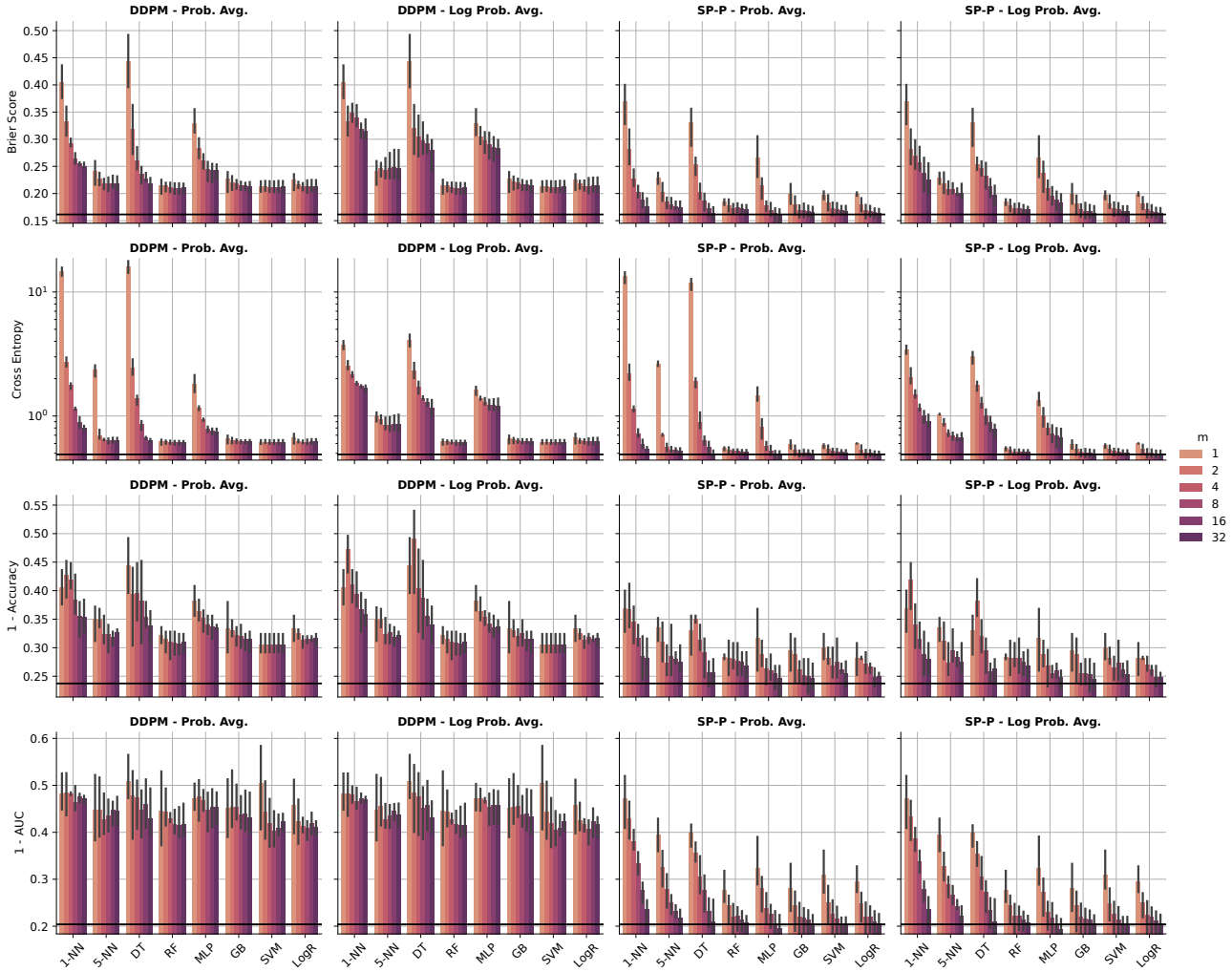
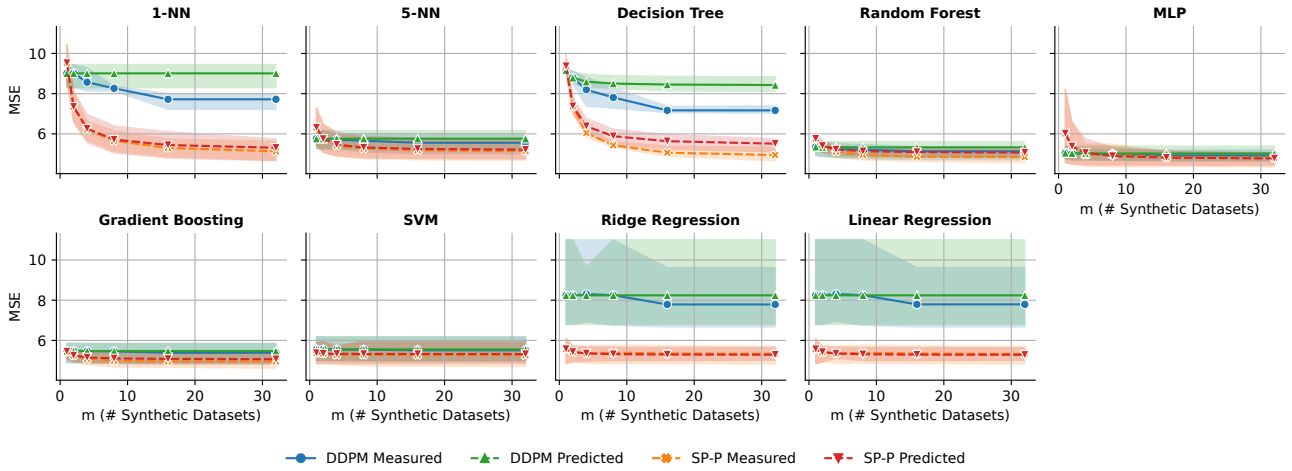
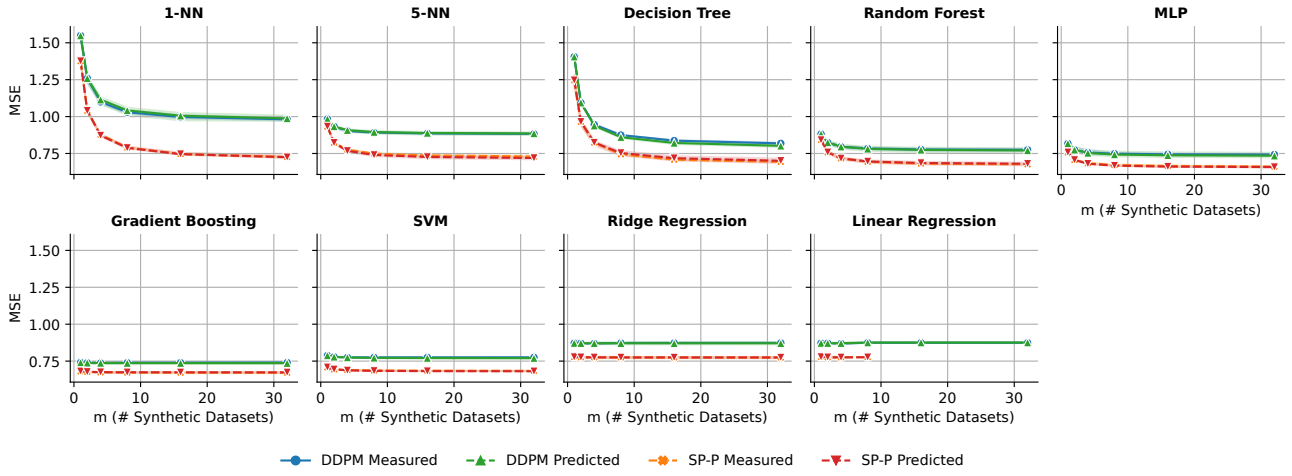


Figure S5: All error metrics on the German credit dataset. Note the he logarithmic scale on the cross entropy y-axis. The predictors are nearest neighbours with 1 or 5 neighbours (1-NN and 5-NN), decision tree (DT), random forest (RF), a multilayer perceptron (MLP), gradient boosted trees (GB), a support vector machine (SVM) and logistic regression (LogR). The black line show the loss of the best downstream predictor trained on real data. The results are averaged over 3 repeats with different train-test splits. The error bars are 95% confidence intervals formed by bootstrapping over the repeats.

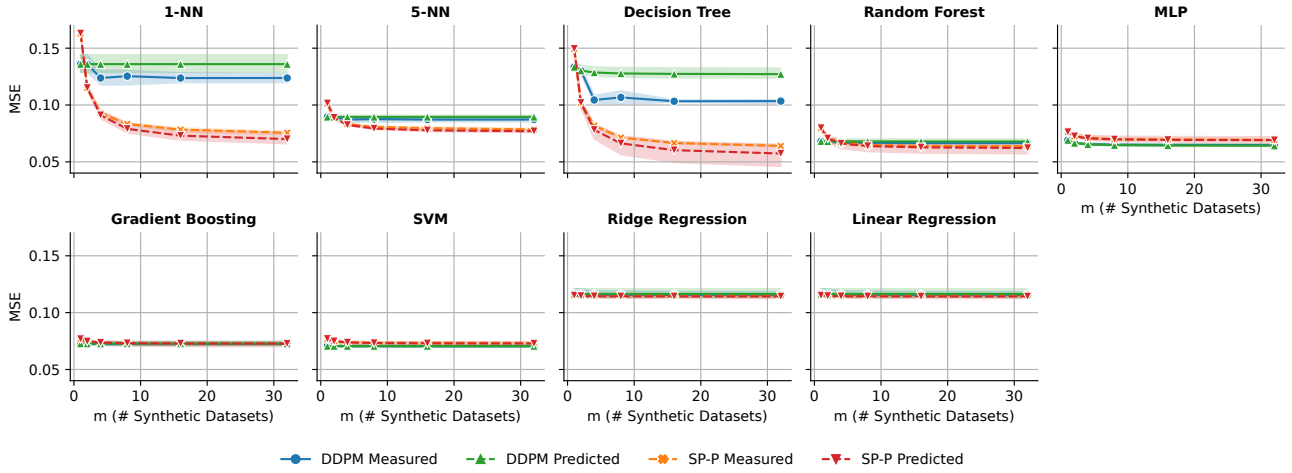


(a) Abalone

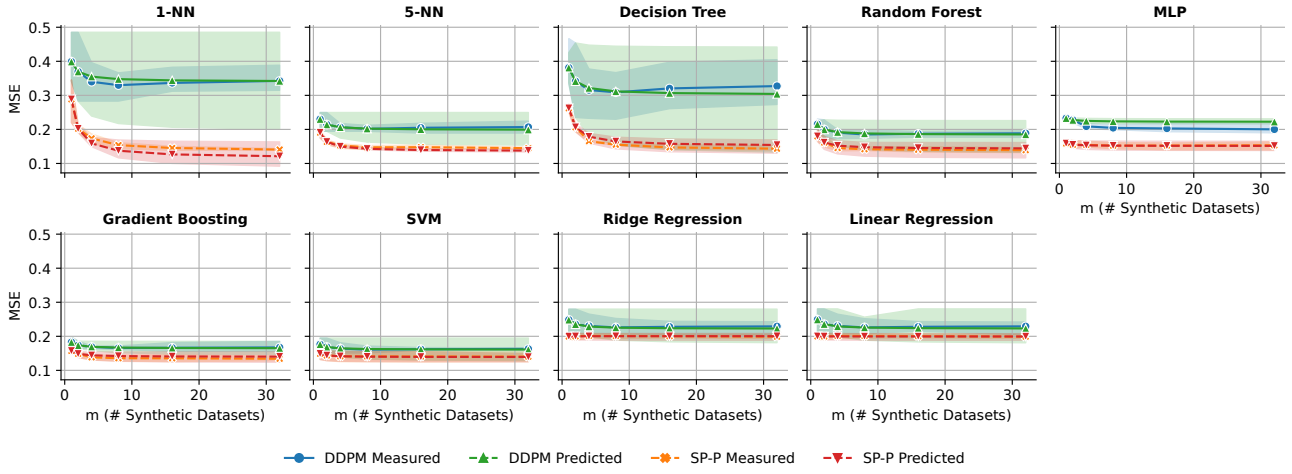


(b) ACS 2018

Figure S6: MSE prediction on the first two regression datasets. The predictions for synthpop are very accurate, and the predictions for DDPM are accurate for most cases. The linear regression measured MSE line for synthpop with ACS 2018 data is cut off due to excluding repeats with extremely large MSE ($\geq 10^6$). 1-NN and 5-NN are nearest neighbours with 1 or 5 neighbours. The results are averaged over 3 repeats with different train-test splits. The error bands are 95% confidence intervals formed by bootstrapping over the repeats.

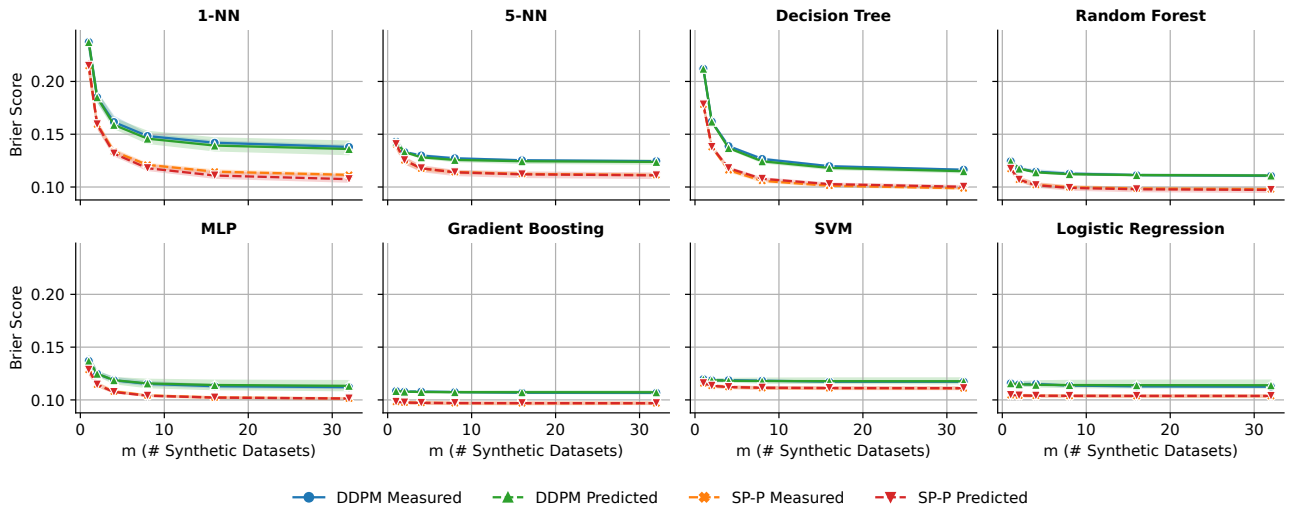


(a) California Housing

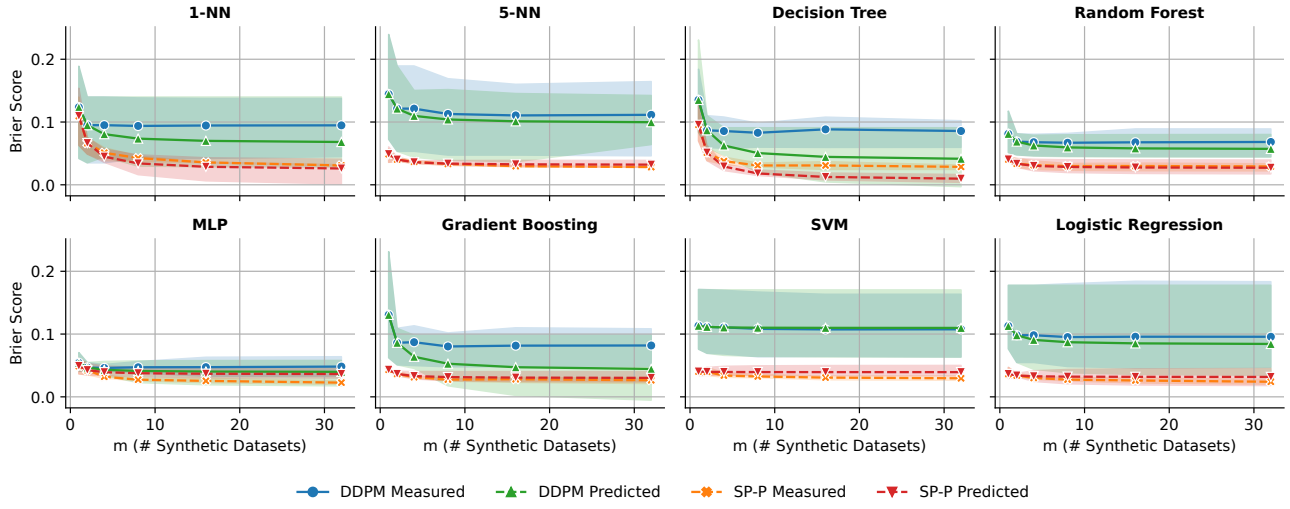


(b) Insurance

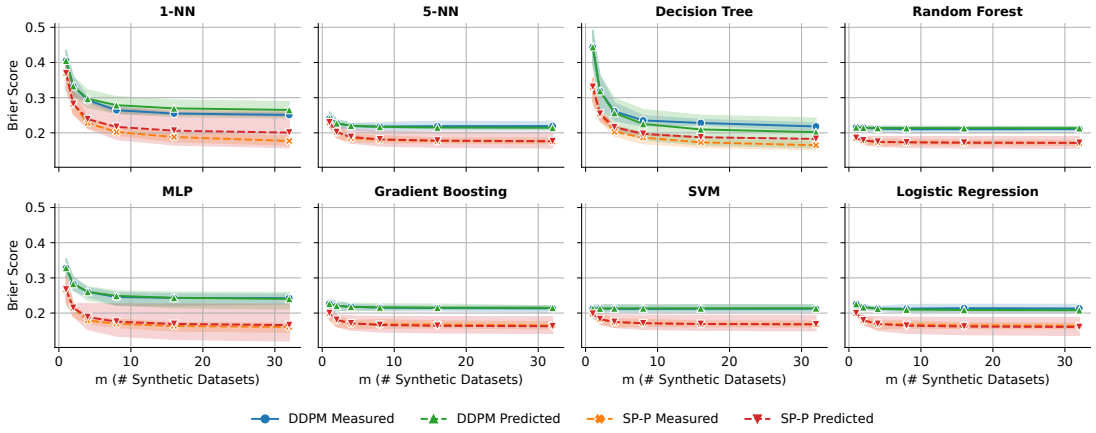
Figure S7: MSE prediction on the last two regression datasets. The predictions for synthpop are very accurate, and the predictions for DDPM are accurate for most cases. 1-NN and 5-NN are nearest neighbours with 1 or 5 neighbours. The results are averaged over 3 repeats with different train-test splits. The error bands are 95% confidence intervals formed by bootstrapping over the repeats.



(a) Adult

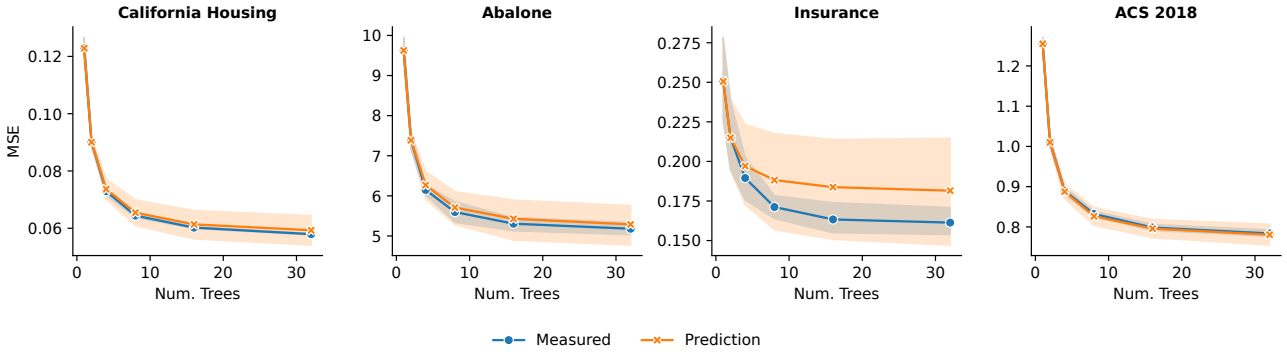


(b) Breast Cancer

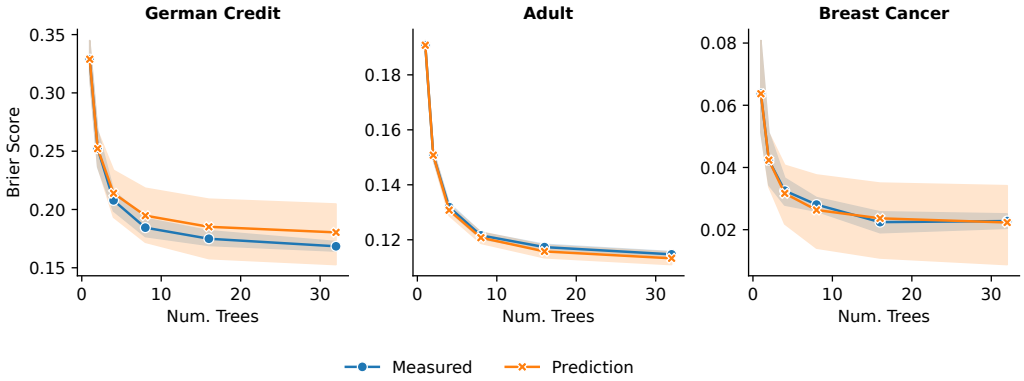


(c) German credit

Figure S8: Brier score prediction on three classification datasets. The predictions are accurate, but can have high variance. 1-NN and 5-NN are nearest neighbours with 1 or 5 neighbours. The results are averaged over 3 repeats with different train-test splits. The error bands are 95% confidence intervals formed by bootstrapping over the repeats.



(a) Regression



(b) Classification

Figure S9: Random forest performance prediction on the regression datasets in (a) and classification datasets in (b). The prediction is reasonably accurate on the datasets with accurate estimates of the error. On the other datasets, the prediction can have high variance. The lines show averages over 3 different train-test splits and 3 repeats of model training per split. The error bands are 95% confidence intervals formed by bootsrapping over the repeats and different splits.

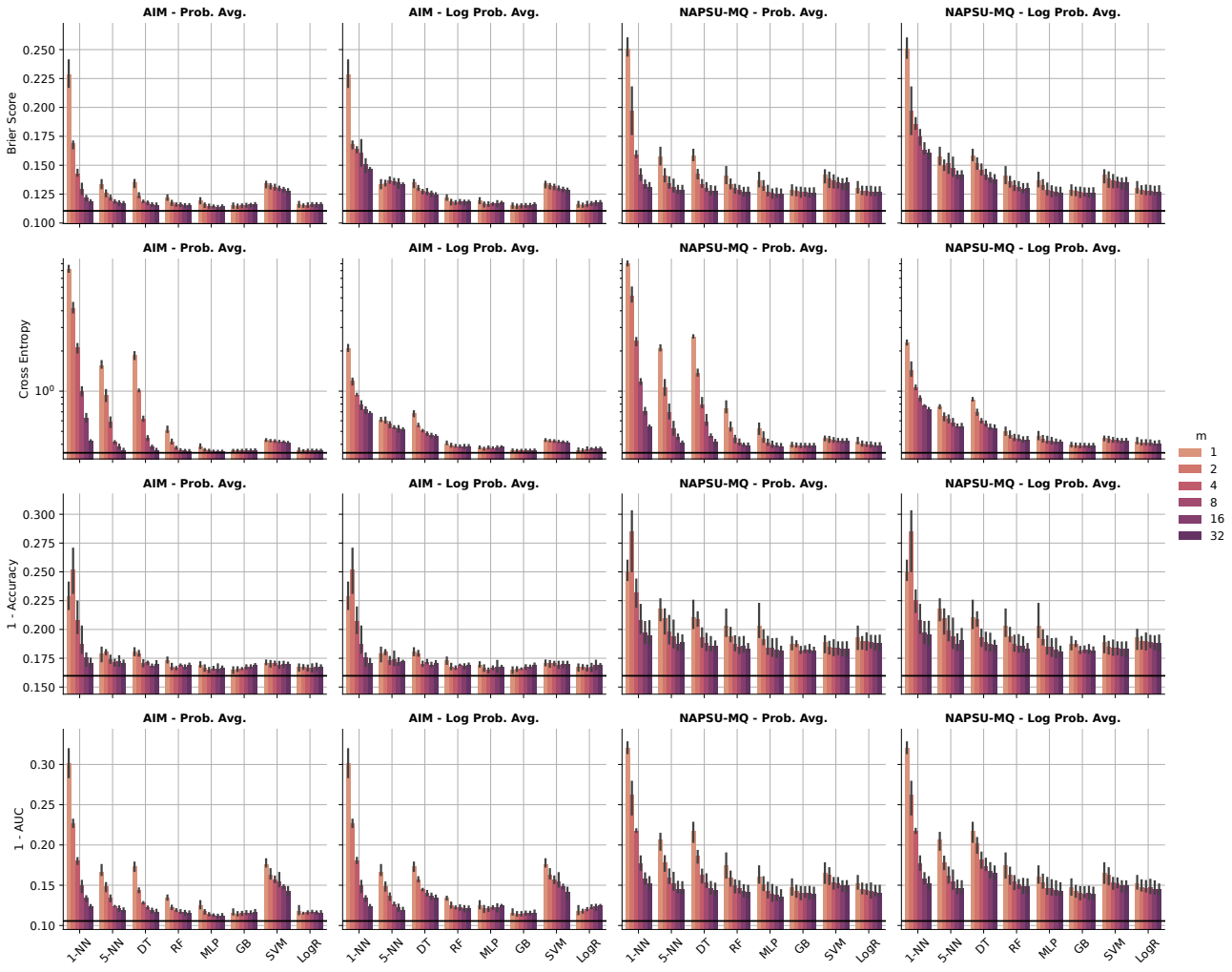


Figure S10: All error metrics on the Adult dataset with reduced features and DP synthetic data generation. Note the he logarithmic scale on the cross entropy y-axis. The privacy parameters are $\epsilon = 1.5$, $\delta = n^{-2} \approx 4.7 \cdot 10^{-7}$. The predictors are nearest neighbours with 1 or 5 neighbours (1-NN and 5-NN), decision tree (DT), random forest (RF), a multilayer perceptron (MLP), gradient boosted trees (GB), a support vector machine (SVM) and logistic regression (LogR). The black line show the loss of the best downstream predictor trained on real data. The results are averaged over 3 repeats with different train-test splits. The error bars are 95% confidence intervals formed by bootstrapping over the repeats.

E.3 Result Tables

Table S3: Table of synthetic data generator comparison results from Figure S1. The numbers are the mean MSE ± standard deviation from 3 repeats.

Downstream	m Generator	1	2	5	10
1-NN	CTGAN	0.2621 ± 0.0228	0.1775 ± 0.0037	0.1316 ± 0.0044	0.1164 ± 0.0040
	DDPM	0.1365 ± 0.0029	0.1365 ± 0.0029	0.1365 ± 0.0029	0.1343 ± 0.0019
	DDPM-KL	0.2447 ± 0.0082	0.2447 ± 0.0082	0.2238 ± 0.0240	0.2124 ± 0.0085
	SP-IP	0.1544 ± 0.0040	0.1139 ± 0.0028	0.0892 ± 0.0018	0.0805 ± 0.0019
	SP-P	0.1654 ± 0.0014	0.1172 ± 0.0012	0.0893 ± 0.0027	0.0801 ± 0.0015
	TVAE	0.2195 ± 0.0036	0.2002 ± 0.0316	0.2114 ± 0.0123	0.1829 ± 0.0073
5-NN	CTGAN	0.1518 ± 0.0110	0.1230 ± 0.0047	0.1125 ± 0.0050	0.1092 ± 0.0040
	DDPM	0.0885 ± 0.0023	0.0885 ± 0.0023	0.0885 ± 0.0023	0.0881 ± 0.0020
	DDPM-KL	0.1379 ± 0.0015	0.1379 ± 0.0015	0.1342 ± 0.0023	0.1320 ± 0.0023
	SP-IP	0.0936 ± 0.0030	0.0852 ± 0.0027	0.0797 ± 0.0021	0.0780 ± 0.0021
	SP-P	0.1029 ± 0.0013	0.0891 ± 0.0020	0.0814 ± 0.0015	0.0788 ± 0.0014
	TVAE	0.1240 ± 0.0008	0.1201 ± 0.0073	0.1227 ± 0.0028	0.1161 ± 0.0018
Decision Tree	CTGAN	0.2926 ± 0.0398	0.1818 ± 0.0048	0.1301 ± 0.0034	0.1136 ± 0.0024
	DDPM	0.1332 ± 0.0076	0.1285 ± 0.0068	0.1266 ± 0.0069	0.1219 ± 0.0033
	DDPM-KL	0.2395 ± 0.0041	0.2314 ± 0.0043	0.2024 ± 0.0272	0.1885 ± 0.0186
	SP-IP	0.1355 ± 0.0048	0.0981 ± 0.0010	0.0762 ± 0.0014	0.0689 ± 0.0008
	SP-P	0.1432 ± 0.0058	0.1015 ± 0.0011	0.0758 ± 0.0013	0.0669 ± 0.0010
	TVAE	0.2086 ± 0.0063	0.1838 ± 0.0328	0.1894 ± 0.0149	0.1620 ± 0.0091
Random Forest	CTGAN	0.1286 ± 0.0151	0.1050 ± 0.0032	0.1004 ± 0.0026	0.0988 ± 0.0024
	DDPM	0.0681 ± 0.0020	0.0678 ± 0.0022	0.0674 ± 0.0022	0.0673 ± 0.0021
	DDPM-KL	0.1113 ± 0.0018	0.1106 ± 0.0019	0.1089 ± 0.0008	0.1082 ± 0.0022
	SP-IP	0.0676 ± 0.0012	0.0649 ± 0.0010	0.0626 ± 0.0008	0.0619 ± 0.0007
	SP-P	0.0762 ± 0.0020	0.0696 ± 0.0014	0.0642 ± 0.0020	0.0629 ± 0.0018
	TVAE	0.0912 ± 0.0013	0.0888 ± 0.0024	0.0889 ± 0.0017	0.0866 ± 0.0022
MLP	CTGAN	0.1267 ± 0.0082	0.1054 ± 0.0105	0.0991 ± 0.0075	0.0958 ± 0.0049
	DDPM	0.0650 ± 0.0014	0.0642 ± 0.0016	0.0638 ± 0.0007	0.0635 ± 0.0009
	DDPM-KL	0.1032 ± 0.0020	0.1020 ± 0.0028	0.0999 ± 0.0023	0.0985 ± 0.0036
	SP-IP	0.0702 ± 0.0018	0.0682 ± 0.0014	0.0661 ± 0.0011	0.0658 ± 0.0007
	SP-P	0.0747 ± 0.0029	0.0705 ± 0.0006	0.0685 ± 0.0003	0.0681 ± 0.0006
	TVAE	0.0882 ± 0.0065	0.0834 ± 0.0023	0.0818 ± 0.0019	0.0809 ± 0.0021
Gradient Boosting	CTGAN	0.1270 ± 0.0162	0.1053 ± 0.0044	0.1026 ± 0.0030	0.1012 ± 0.0023
	DDPM	0.0725 ± 0.0023	0.0725 ± 0.0023	0.0725 ± 0.0023	0.0724 ± 0.0022
	DDPM-KL	0.1054 ± 0.0038	0.1054 ± 0.0038	0.1045 ± 0.0031	0.1044 ± 0.0037
	SP-IP	0.0715 ± 0.0018	0.0698 ± 0.0013	0.0689 ± 0.0012	0.0685 ± 0.0011
	SP-P	0.0745 ± 0.0023	0.0724 ± 0.0015	0.0711 ± 0.0012	0.0708 ± 0.0018
	TVAE	0.0912 ± 0.0026	0.0900 ± 0.0018	0.0901 ± 0.0018	0.0893 ± 0.0018
SVM	CTGAN	0.1235 ± 0.0144	0.1051 ± 0.0017	0.1005 ± 0.0024	0.0993 ± 0.0022
	DDPM	0.0700 ± 0.0027	0.0700 ± 0.0027	0.0700 ± 0.0027	0.0700 ± 0.0027
	DDPM-KL	0.1026 ± 0.0032	0.1026 ± 0.0032	0.1017 ± 0.0024	0.1013 ± 0.0034
	SP-IP	0.0736 ± 0.0028	0.0725 ± 0.0026	0.0712 ± 0.0022	0.0711 ± 0.0020
	SP-P	0.0768 ± 0.0025	0.0745 ± 0.0020	0.0728 ± 0.0018	0.0724 ± 0.0020
	TVAE	0.0944 ± 0.0022	0.0934 ± 0.0004	0.0938 ± 0.0011	0.0919 ± 0.0023
Ridge Regression	CTGAN	0.1587 ± 0.0182	0.1391 ± 0.0032	0.1381 ± 0.0042	0.1372 ± 0.0044
	DDPM	0.1110 ± 0.0036	0.1110 ± 0.0036	0.1110 ± 0.0036	0.1109 ± 0.0035
	DDPM-KL	0.1296 ± 0.0035	0.1296 ± 0.0035	0.1295 ± 0.0035	0.1293 ± 0.0037
	SP-IP	0.1098 ± 0.0032	0.1095 ± 0.0031	0.1095 ± 0.0030	0.1096 ± 0.0030
	SP-P	0.1109 ± 0.0034	0.1106 ± 0.0030	0.1105 ± 0.0030	0.1105 ± 0.0033
	TVAE	0.1353 ± 0.0044	0.1343 ± 0.0053	0.1341 ± 0.0055	0.1336 ± 0.0053
Linear Regression	CTGAN	0.1587 ± 0.0182	0.1391 ± 0.0032	0.1381 ± 0.0042	0.1372 ± 0.0044
	DDPM	0.1110 ± 0.0036	0.1110 ± 0.0036	0.1110 ± 0.0036	0.1109 ± 0.0035
	DDPM-KL	0.1296 ± 0.0035	0.1296 ± 0.0035	0.1295 ± 0.0035	0.1293 ± 0.0037
	SP-IP	0.1098 ± 0.0032	0.1095 ± 0.0031	0.1095 ± 0.0030	0.1096 ± 0.0030
	SP-P	0.1109 ± 0.0034	0.1105 ± 0.0030	0.1105 ± 0.0030	0.1105 ± 0.0033
	TVAE	0.1352 ± 0.0044	0.1343 ± 0.0053	0.1341 ± 0.0055	0.1336 ± 0.0053

Table S4: Table of results from the Abalone dataset. The numbers are the mean MSE ± standard deviation from 3 repeats.

Downstream	m / Generator	1	2	4	8	16	32
Linear Regression	DDPM	8.24 ± 2.392	8.24 ± 2.392	8.31 ± 2.334	8.25 ± 2.388	7.79 ± 1.602	7.79 ± 1.602
	SP-P	5.57 ± 0.646	5.42 ± 0.429	5.35 ± 0.443	5.38 ± 0.470	5.35 ± 0.463	5.33 ± 0.452
Ridge Regression	DDPM	8.24 ± 2.392	8.24 ± 2.392	8.31 ± 2.335	8.25 ± 2.388	7.79 ± 1.599	7.79 ± 1.599
	SP-P	5.57 ± 0.648	5.43 ± 0.432	5.35 ± 0.445	5.39 ± 0.472	5.35 ± 0.465	5.33 ± 0.454
1-NN	DDPM	9.01 ± 0.621	9.01 ± 0.621	8.57 ± 0.621	8.27 ± 0.123	7.72 ± 0.434	7.72 ± 0.434
	SP-P	9.52 ± 0.890	7.35 ± 0.764	6.21 ± 0.482	5.66 ± 0.514	5.30 ± 0.439	5.13 ± 0.479
5-NN	DDPM	5.76 ± 0.457	5.76 ± 0.457	5.71 ± 0.399	5.66 ± 0.493	5.56 ± 0.491	5.56 ± 0.491
	SP-P	6.32 ± 0.932	5.75 ± 0.704	5.42 ± 0.550	5.26 ± 0.539	5.18 ± 0.523	5.14 ± 0.513
Decision Tree	DDPM	9.17 ± 0.199	8.79 ± 0.289	8.19 ± 0.741	7.81 ± 0.509	7.17 ± 0.174	7.17 ± 0.194
	SP-P	9.37 ± 0.523	7.38 ± 0.418	6.05 ± 0.237	5.43 ± 0.165	5.07 ± 0.274	4.95 ± 0.348
Random Forest	DDPM	5.36 ± 0.372	5.34 ± 0.405	5.24 ± 0.361	5.21 ± 0.399	5.14 ± 0.360	5.13 ± 0.361
	SP-P	5.76 ± 0.378	5.40 ± 0.282	5.12 ± 0.268	4.96 ± 0.253	4.87 ± 0.276	4.85 ± 0.318
Gradient Boosting	DDPM	5.47 ± 0.506	5.47 ± 0.506	5.47 ± 0.503	5.45 ± 0.526	5.39 ± 0.509	5.39 ± 0.508
	SP-P	5.45 ± 0.336	5.25 ± 0.301	5.08 ± 0.294	4.99 ± 0.329	4.96 ± 0.324	4.96 ± 0.363
MLP	DDPM	5.06 ± 0.323	5.04 ± 0.335	5.00 ± 0.249	5.00 ± 0.273	4.92 ± 0.267	4.92 ± 0.280
	SP-P	6.02 ± 1.935	5.38 ± 1.075	4.96 ± 0.550	4.97 ± 0.570	4.84 ± 0.399	4.81 ± 0.399
SVM	DDPM	5.55 ± 0.592	5.55 ± 0.592	5.55 ± 0.586	5.54 ± 0.598	5.51 ± 0.610	5.51 ± 0.610
	SP-P	5.38 ± 0.515	5.35 ± 0.546	5.28 ± 0.499	5.28 ± 0.583	5.27 ± 0.604	5.26 ± 0.588

Table S5: Table of results from the ACS 2018 dataset. The numbers are the mean MSE ± standard deviation from 3 repeats. The nans for linear regression are caused by excluding repeats with extremely large MSE ($\geq 10^6$).

Downstream	m / Generator	1	2	4	8	16	32
Linear Regression	DDPM	0.87 ± 0.008	0.87 ± 0.007	0.87 ± 0.005	0.87 ± 0.001	0.87 ± 0.003	0.87 ± 0.003
	SP-P	0.78 ± 0.007	0.78 ± 0.006	0.78 ± 0.007	0.78 ± nan	nan	nan
Ridge Regression	DDPM	0.87 ± 0.008	0.87 ± 0.007	0.87 ± 0.005	0.87 ± 0.003	0.87 ± 0.004	0.87 ± 0.003
	SP-P	0.78 ± 0.007	0.78 ± 0.006	0.78 ± 0.007	0.78 ± 0.007	0.77 ± 0.007	0.77 ± 0.006
1-NN	DDPM	1.55 ± 0.022	1.26 ± 0.026	1.10 ± 0.019	1.03 ± 0.014	1.00 ± 0.013	0.98 ± 0.011
	SP-P	1.38 ± 0.015	1.04 ± 0.010	0.88 ± 0.001	0.79 ± 0.004	0.75 ± 0.006	0.73 ± 0.007
5-NN	DDPM	0.98 ± 0.006	0.93 ± 0.009	0.90 ± 0.008	0.89 ± 0.007	0.89 ± 0.004	0.88 ± 0.004
	SP-P	0.93 ± 0.002	0.82 ± 0.005	0.78 ± 0.004	0.75 ± 0.008	0.74 ± 0.008	0.73 ± 0.009
Decision Tree	DDPM	1.41 ± 0.008	1.09 ± 0.005	0.95 ± 0.006	0.87 ± 0.004	0.84 ± 0.006	0.82 ± 0.011
	SP-P	1.25 ± 0.035	0.96 ± 0.023	0.82 ± 0.011	0.74 ± 0.005	0.71 ± 0.003	0.69 ± 0.004
Random Forest	DDPM	0.88 ± 0.010	0.82 ± 0.013	0.80 ± 0.009	0.78 ± 0.009	0.78 ± 0.009	0.77 ± 0.009
	SP-P	0.84 ± 0.010	0.76 ± 0.010	0.72 ± 0.006	0.69 ± 0.005	0.68 ± 0.005	0.68 ± 0.006
Gradient Boosting	DDPM	0.74 ± 0.008	0.74 ± 0.008	0.74 ± 0.006	0.74 ± 0.005	0.74 ± 0.004	0.74 ± 0.004
	SP-P	0.68 ± 0.005	0.68 ± 0.005	0.67 ± 0.005	0.67 ± 0.005	0.67 ± 0.006	0.67 ± 0.006
MLP	DDPM	0.82 ± 0.019	0.77 ± 0.018	0.76 ± 0.017	0.75 ± 0.007	0.74 ± 0.005	0.74 ± 0.004
	SP-P	0.76 ± 0.006	0.71 ± 0.003	0.68 ± 0.008	0.67 ± 0.007	0.67 ± 0.007	0.66 ± 0.006
SVM	DDPM	0.79 ± 0.004	0.78 ± 0.002	0.77 ± 0.002	0.78 ± 0.002	0.78 ± 0.003	0.78 ± 0.002
	SP-P	0.71 ± 0.005	0.69 ± 0.003	0.69 ± 0.002	0.69 ± 0.004	0.69 ± 0.004	0.68 ± 0.004

Table S6: Table of results from the California Housing dataset. The numbers are the mean MSE ± standard deviation from 3 repeats.

Downstream	m / Generator	1	2	4	8	16	32
Linear Regression	DDPM	0.12 ± 0.004	0.12 ± 0.004	0.12 ± 0.003	0.12 ± 0.003	0.12 ± 0.003	0.12 ± 0.003
	SP-P	0.12 ± 0.001	0.11 ± 0.002	0.11 ± 0.002	0.11 ± 0.001	0.11 ± 0.002	0.11 ± 0.002
Ridge Regression	DDPM	0.12 ± 0.004	0.12 ± 0.004	0.12 ± 0.003	0.12 ± 0.003	0.12 ± 0.003	0.12 ± 0.003
	SP-P	0.12 ± 0.001	0.11 ± 0.002	0.11 ± 0.002	0.11 ± 0.001	0.11 ± 0.002	0.11 ± 0.002
1-NN	DDPM	0.14 ± 0.008	0.14 ± 0.008	0.12 ± 0.006	0.13 ± 0.007	0.12 ± 0.004	0.12 ± 0.004
	SP-P	0.16 ± 0.002	0.12 ± 0.002	0.09 ± 0.002	0.08 ± 0.001	0.08 ± 0.002	0.08 ± 0.001
5-NN	DDPM	0.09 ± 0.002	0.09 ± 0.002	0.09 ± 0.002	0.09 ± 0.002	0.09 ± 0.002	0.09 ± 0.002
	SP-P	0.10 ± 0.003	0.09 ± 0.001	0.08 ± 0.001	0.08 ± 0.001	0.08 ± 0.001	0.08 ± 0.001
Decision Tree	DDPM	0.13 ± 0.003	0.13 ± 0.004	0.10 ± 0.004	0.11 ± 0.006	0.10 ± 0.001	0.10 ± 0.002
	SP-P	0.15 ± 0.004	0.10 ± 0.004	0.08 ± 0.001	0.07 ± 0.001	0.07 ± 0.001	0.06 ± 0.001
Random Forest	DDPM	0.07 ± 0.002	0.07 ± 0.002	0.07 ± 0.001	0.07 ± 0.001	0.07 ± 0.001	0.07 ± 0.001
	SP-P	0.08 ± 0.002	0.07 ± 0.003	0.07 ± 0.002	0.06 ± 0.001	0.06 ± 0.001	0.06 ± 0.001
Gradient Boosting	DDPM	0.07 ± 0.002	0.07 ± 0.002	0.07 ± 0.002	0.07 ± 0.002	0.07 ± 0.002	0.07 ± 0.002
	SP-P	0.08 ± 0.002	0.07 ± 0.002	0.07 ± 0.001	0.07 ± 0.001	0.07 ± 0.001	0.07 ± 0.001
MLP	DDPM	0.07 ± 0.000	0.07 ± 0.001	0.07 ± 0.001	0.07 ± 0.001	0.06 ± 0.001	0.06 ± 0.001
	SP-P	0.08 ± 0.001	0.07 ± 0.002	0.07 ± 0.002	0.07 ± 0.001	0.07 ± 0.001	0.07 ± 0.001
SVM	DDPM	0.07 ± 0.001	0.07 ± 0.001	0.07 ± 0.001	0.07 ± 0.001	0.07 ± 0.001	0.07 ± 0.001
	SP-P	0.08 ± 0.001	0.07 ± 0.001	0.07 ± 0.001	0.07 ± 0.001	0.07 ± 0.001	0.07 ± 0.001

Table S7: Table of results from the Insurance dataset. The numbers are the mean MSE ± standard deviation from 3 repeats.

Downstream	m Generator	1	2	4	8	16	32
Linear Regression	DDPM	0.25 ± 0.037	0.24 ± 0.039	0.23 ± 0.030	0.23 ± 0.023	0.23 ± 0.020	0.23 ± 0.021
	SP-P	0.20 ± 0.009	0.20 ± 0.009	0.20 ± 0.008	0.20 ± 0.006	0.20 ± 0.006	0.20 ± 0.006
Ridge Regression	DDPM	0.25 ± 0.037	0.24 ± 0.039	0.23 ± 0.031	0.23 ± 0.023	0.23 ± 0.021	0.23 ± 0.021
	SP-P	0.20 ± 0.009	0.20 ± 0.010	0.20 ± 0.008	0.20 ± 0.007	0.20 ± 0.006	0.20 ± 0.006
1-NN	DDPM	0.40 ± 0.076	0.37 ± 0.104	0.34 ± 0.056	0.33 ± 0.042	0.34 ± 0.040	0.34 ± 0.040
	SP-P	0.29 ± 0.063	0.20 ± 0.012	0.17 ± 0.015	0.15 ± 0.009	0.15 ± 0.006	0.14 ± 0.002
5-NN	DDPM	0.23 ± 0.031	0.21 ± 0.031	0.21 ± 0.017	0.20 ± 0.009	0.20 ± 0.014	0.21 ± 0.017
	SP-P	0.19 ± 0.014	0.16 ± 0.007	0.15 ± 0.008	0.15 ± 0.005	0.15 ± 0.005	0.14 ± 0.003
Decision Tree	DDPM	0.38 ± 0.074	0.34 ± 0.110	0.32 ± 0.075	0.31 ± 0.071	0.32 ± 0.070	0.33 ± 0.069
	SP-P	0.26 ± 0.011	0.21 ± 0.012	0.17 ± 0.003	0.16 ± 0.002	0.15 ± 0.007	0.14 ± 0.007
Random Forest	DDPM	0.22 ± 0.015	0.20 ± 0.028	0.19 ± 0.015	0.19 ± 0.013	0.19 ± 0.009	0.19 ± 0.010
	SP-P	0.18 ± 0.023	0.16 ± 0.019	0.15 ± 0.007	0.14 ± 0.006	0.14 ± 0.008	0.14 ± 0.007
Gradient Boosting	DDPM	0.18 ± 0.017	0.17 ± 0.011	0.17 ± 0.004	0.17 ± 0.005	0.17 ± 0.013	0.17 ± 0.016
	SP-P	0.16 ± 0.009	0.15 ± 0.009	0.14 ± 0.007	0.14 ± 0.009	0.14 ± 0.010	0.13 ± 0.009
MLP	DDPM	0.23 ± 0.010	0.23 ± 0.008	0.21 ± 0.008	0.20 ± 0.009	0.20 ± 0.013	0.20 ± 0.014
	SP-P	0.16 ± 0.009	0.16 ± 0.011	0.15 ± 0.009	0.15 ± 0.009	0.15 ± 0.011	0.15 ± 0.012
SVM	DDPM	0.18 ± 0.018	0.17 ± 0.023	0.16 ± 0.015	0.16 ± 0.010	0.16 ± 0.004	0.16 ± 0.004
	SP-P	0.15 ± 0.015	0.14 ± 0.014	0.14 ± 0.015	0.14 ± 0.015	0.14 ± 0.015	0.14 ± 0.014

Table S8: Table of results from the Adult dataset. The numbers are the mean Brier score ± standard deviation from 3 repeats.

Downstream	m Generator	1	2	4	8	16	32
1-NN	DDPM - Log Prob. Avg.	0.24 ± 0.005	0.19 ± 0.004	0.19 ± 0.006	0.18 ± 0.002	0.18 ± 0.003	0.17 ± 0.003
	DDPM - Prob. Avg.	0.24 ± 0.005	0.19 ± 0.004	0.16 ± 0.005	0.15 ± 0.002	0.14 ± 0.002	0.14 ± 0.002
	SP-P - Log Prob. Avg.	0.21 ± 0.005	0.16 ± 0.003	0.15 ± 0.003	0.15 ± 0.005	0.14 ± 0.003	0.14 ± 0.003
	SP-P - Prob. Avg.	0.21 ± 0.005	0.16 ± 0.003	0.13 ± 0.002	0.12 ± 0.002	0.11 ± 0.001	0.11 ± 0.001
5-NN	DDPM - Log Prob. Avg.	0.14 ± 0.003	0.14 ± 0.001	0.14 ± 0.002	0.14 ± 0.002	0.13 ± 0.001	0.13 ± 0.000
	DDPM - Prob. Avg.	0.14 ± 0.003	0.13 ± 0.001	0.13 ± 0.001	0.13 ± 0.001	0.13 ± 0.000	0.12 ± 0.000
	SP-P - Log Prob. Avg.	0.14 ± 0.003	0.14 ± 0.004	0.14 ± 0.003	0.13 ± 0.003	0.13 ± 0.001	0.13 ± 0.002
	SP-P - Prob. Avg.	0.14 ± 0.003	0.13 ± 0.003	0.12 ± 0.002	0.11 ± 0.001	0.11 ± 0.001	0.11 ± 0.001
Decision Tree	DDPM - Log Prob. Avg.	0.21 ± 0.000	0.17 ± 0.001	0.16 ± 0.001	0.15 ± 0.002	0.15 ± 0.000	0.14 ± 0.001
	DDPM - Prob. Avg.	0.21 ± 0.000	0.16 ± 0.001	0.14 ± 0.001	0.13 ± 0.001	0.12 ± 0.000	0.12 ± 0.000
	SP-P - Log Prob. Avg.	0.18 ± 0.003	0.15 ± 0.001	0.14 ± 0.002	0.13 ± 0.002	0.12 ± 0.002	0.12 ± 0.001
	SP-P - Prob. Avg.	0.18 ± 0.003	0.14 ± 0.001	0.12 ± 0.002	0.11 ± 0.001	0.10 ± 0.001	0.10 ± 0.001
Random Forest	DDPM - Log Prob. Avg.	0.12 ± 0.001	0.12 ± 0.001	0.11 ± 0.001	0.11 ± 0.001	0.11 ± 0.001	0.11 ± 0.001
	DDPM - Prob. Avg.	0.12 ± 0.001	0.12 ± 0.001	0.11 ± 0.001	0.11 ± 0.001	0.11 ± 0.001	0.11 ± 0.001
	SP-P - Log Prob. Avg.	0.12 ± 0.002	0.11 ± 0.002	0.11 ± 0.002	0.10 ± 0.001	0.10 ± 0.001	0.10 ± 0.001
	SP-P - Prob. Avg.	0.12 ± 0.002	0.11 ± 0.002	0.10 ± 0.002	0.10 ± 0.001	0.10 ± 0.001	0.10 ± 0.001
MLP	DDPM - Log Prob. Avg.	0.14 ± 0.006	0.13 ± 0.002	0.12 ± 0.002	0.12 ± 0.002	0.12 ± 0.001	0.12 ± 0.001
	DDPM - Prob. Avg.	0.14 ± 0.006	0.12 ± 0.002	0.12 ± 0.002	0.12 ± 0.002	0.11 ± 0.001	0.11 ± 0.001
	SP-P - Log Prob. Avg.	0.13 ± 0.003	0.12 ± 0.002	0.11 ± 0.002	0.11 ± 0.001	0.11 ± 0.002	0.11 ± 0.001
	SP-P - Prob. Avg.	0.13 ± 0.003	0.11 ± 0.002	0.11 ± 0.001	0.10 ± 0.001	0.10 ± 0.001	0.10 ± 0.001
Gradient Boosting	DDPM - Log Prob. Avg.	0.11 ± 0.002	0.11 ± 0.000	0.11 ± 0.000	0.11 ± 0.000	0.11 ± 0.001	0.11 ± 0.001
	DDPM - Prob. Avg.	0.11 ± 0.002	0.11 ± 0.000	0.11 ± 0.000	0.11 ± 0.000	0.11 ± 0.001	0.11 ± 0.001
	SP-P - Log Prob. Avg.	0.10 ± 0.001	0.10 ± 0.001	0.10 ± 0.001	0.10 ± 0.001	0.10 ± 0.001	0.10 ± 0.001
	SP-P - Prob. Avg.	0.10 ± 0.001	0.10 ± 0.001	0.10 ± 0.001	0.10 ± 0.001	0.10 ± 0.001	0.10 ± 0.001
SVM	DDPM - Log Prob. Avg.	0.12 ± 0.003	0.12 ± 0.001	0.12 ± 0.001	0.12 ± 0.000	0.12 ± 0.000	0.12 ± 0.001
	DDPM - Prob. Avg.	0.12 ± 0.003	0.12 ± 0.001	0.12 ± 0.001	0.12 ± 0.000	0.12 ± 0.000	0.12 ± 0.001
	SP-P - Log Prob. Avg.	0.12 ± 0.001	0.11 ± 0.001	0.11 ± 0.001	0.11 ± 0.001	0.11 ± 0.001	0.11 ± 0.001
	SP-P - Prob. Avg.	0.12 ± 0.001	0.11 ± 0.001	0.11 ± 0.001	0.11 ± 0.001	0.11 ± 0.001	0.11 ± 0.001
Logistic Regression	DDPM - Log Prob. Avg.	0.12 ± 0.001	0.12 ± 0.002	0.12 ± 0.001	0.11 ± 0.001	0.11 ± 0.000	0.11 ± 0.000
	DDPM - Prob. Avg.	0.12 ± 0.001	0.11 ± 0.002	0.12 ± 0.001	0.11 ± 0.001	0.11 ± 0.000	0.11 ± 0.000
	SP-P - Log Prob. Avg.	0.10 ± 0.001	0.10 ± 0.001	0.10 ± 0.001	0.10 ± 0.001	0.10 ± 0.001	0.10 ± 0.001
	SP-P - Prob. Avg.	0.10 ± 0.001	0.10 ± 0.001	0.10 ± 0.001	0.10 ± 0.001	0.10 ± 0.001	0.10 ± 0.001

Table S9: Table of results from the Breast Cancer dataset. The numbers are the mean Brier score ± standard deviation from 3 repeats.

Downstream	m / Generator	1	2	4	8	16	32
1-NN	DDPM - Log Prob. Avg.	0.12 ± 0.075	0.09 ± 0.054	0.09 ± 0.054	0.10 ± 0.043	0.10 ± 0.043	0.10 ± 0.043
	DDPM - Prob. Avg.	0.12 ± 0.075	0.09 ± 0.054	0.09 ± 0.054	0.09 ± 0.049	0.09 ± 0.048	0.09 ± 0.047
	SP-P - Log Prob. Avg.	0.11 ± 0.045	0.07 ± 0.015	0.05 ± 0.013	0.04 ± 0.002	0.04 ± 0.003	0.03 ± 0.003
	SP-P - Prob. Avg.	0.11 ± 0.045	0.07 ± 0.015	0.05 ± 0.011	0.04 ± 0.004	0.04 ± 0.000	0.03 ± 0.001
5-NN	DDPM - Log Prob. Avg.	0.14 ± 0.086	0.13 ± 0.087	0.13 ± 0.087	0.13 ± 0.081	0.12 ± 0.070	0.12 ± 0.077
	DDPM - Prob. Avg.	0.14 ± 0.086	0.12 ± 0.068	0.12 ± 0.068	0.11 ± 0.061	0.11 ± 0.058	0.11 ± 0.060
	SP-P - Log Prob. Avg.	0.05 ± 0.012	0.04 ± 0.005	0.03 ± 0.007	0.03 ± 0.003	0.03 ± 0.005	0.02 ± 0.004
	SP-P - Prob. Avg.	0.05 ± 0.012	0.04 ± 0.003	0.04 ± 0.002	0.03 ± 0.002	0.03 ± 0.001	0.03 ± 0.000
Decision Tree	DDPM - Log Prob. Avg.	0.14 ± 0.083	0.09 ± 0.026	0.09 ± 0.026	0.10 ± 0.033	0.10 ± 0.033	0.10 ± 0.031
	DDPM - Prob. Avg.	0.14 ± 0.083	0.09 ± 0.026	0.09 ± 0.025	0.08 ± 0.020	0.09 ± 0.025	0.09 ± 0.023
	SP-P - Log Prob. Avg.	0.10 ± 0.028	0.05 ± 0.012	0.04 ± 0.009	0.03 ± 0.004	0.03 ± 0.009	0.03 ± 0.006
	SP-P - Prob. Avg.	0.10 ± 0.028	0.05 ± 0.012	0.04 ± 0.006	0.03 ± 0.003	0.03 ± 0.006	0.03 ± 0.004
Random Forest	DDPM - Log Prob. Avg.	0.08 ± 0.033	0.07 ± 0.018	0.07 ± 0.019	0.07 ± 0.019	0.07 ± 0.022	0.07 ± 0.022
	DDPM - Prob. Avg.	0.08 ± 0.033	0.07 ± 0.018	0.07 ± 0.019	0.07 ± 0.019	0.07 ± 0.022	0.07 ± 0.022
	SP-P - Log Prob. Avg.	0.04 ± 0.004	0.03 ± 0.008	0.03 ± 0.008	0.03 ± 0.007	0.03 ± 0.006	0.03 ± 0.006
	SP-P - Prob. Avg.	0.04 ± 0.004	0.03 ± 0.007	0.03 ± 0.007	0.03 ± 0.006	0.03 ± 0.006	0.03 ± 0.006
MLP	DDPM - Log Prob. Avg.	0.05 ± 0.015	0.05 ± 0.008	0.04 ± 0.007	0.05 ± 0.010	0.05 ± 0.016	0.05 ± 0.017
	DDPM - Prob. Avg.	0.05 ± 0.015	0.05 ± 0.007	0.05 ± 0.005	0.05 ± 0.009	0.05 ± 0.015	0.05 ± 0.016
	SP-P - Log Prob. Avg.	0.05 ± 0.012	0.04 ± 0.010	0.03 ± 0.006	0.03 ± 0.001	0.02 ± 0.007	0.02 ± 0.006
	SP-P - Prob. Avg.	0.05 ± 0.012	0.04 ± 0.009	0.03 ± 0.003	0.03 ± 0.002	0.03 ± 0.004	0.02 ± 0.003
Gradient Boosting	DDPM - Log Prob. Avg.	0.13 ± 0.089	0.08 ± 0.031	0.09 ± 0.034	0.08 ± 0.029	0.08 ± 0.033	0.08 ± 0.032
	DDPM - Prob. Avg.	0.13 ± 0.089	0.09 ± 0.031	0.09 ± 0.033	0.08 ± 0.029	0.08 ± 0.031	0.08 ± 0.030
	SP-P - Log Prob. Avg.	0.04 ± 0.002	0.04 ± 0.004	0.03 ± 0.003	0.03 ± 0.004	0.03 ± 0.006	0.03 ± 0.006
	SP-P - Prob. Avg.	0.04 ± 0.002	0.04 ± 0.004	0.03 ± 0.002	0.03 ± 0.002	0.03 ± 0.004	0.03 ± 0.005
SVM	DDPM - Log Prob. Avg.	0.11 ± 0.051	0.11 ± 0.053	0.11 ± 0.053	0.11 ± 0.053	0.11 ± 0.051	0.11 ± 0.051
	DDPM - Prob. Avg.	0.11 ± 0.051	0.11 ± 0.053	0.11 ± 0.053	0.11 ± 0.053	0.11 ± 0.051	0.11 ± 0.051
	SP-P - Log Prob. Avg.	0.04 ± 0.005	0.04 ± 0.007	0.03 ± 0.005	0.03 ± 0.003	0.03 ± 0.003	0.03 ± 0.003
	SP-P - Prob. Avg.	0.04 ± 0.005	0.04 ± 0.006	0.03 ± 0.004	0.03 ± 0.003	0.03 ± 0.004	0.03 ± 0.003
Logistic Regression	DDPM - Log Prob. Avg.	0.11 ± 0.056	0.10 ± 0.071	0.10 ± 0.071	0.09 ± 0.075	0.09 ± 0.078	0.09 ± 0.077
	DDPM - Prob. Avg.	0.11 ± 0.056	0.10 ± 0.069	0.10 ± 0.069	0.10 ± 0.074	0.10 ± 0.077	0.10 ± 0.076
	SP-P - Log Prob. Avg.	0.04 ± 0.009	0.03 ± 0.003	0.03 ± 0.004	0.02 ± 0.004	0.02 ± 0.003	0.02 ± 0.003
	SP-P - Prob. Avg.	0.04 ± 0.009	0.03 ± 0.003	0.03 ± 0.003	0.03 ± 0.003	0.03 ± 0.004	0.02 ± 0.004

Table S10: Table of results from the German Credit dataset. The numbers are the mean Brier score ± standard deviation from 3 repeats.

Downstream	m / Generator	1	2	4	8	16	32
1-NN	DDPM - Log Prob. Avg.	0.41 ± 0.030	0.33 ± 0.027	0.35 ± 0.017	0.34 ± 0.021	0.32 ± 0.014	0.32 ± 0.018
	DDPM - Prob. Avg.	0.41 ± 0.030	0.33 ± 0.027	0.29 ± 0.007	0.26 ± 0.010	0.25 ± 0.002	0.25 ± 0.005
	SP-P - Log Prob. Avg.	0.37 ± 0.037	0.28 ± 0.033	0.27 ± 0.026	0.26 ± 0.024	0.24 ± 0.031	0.23 ± 0.028
	SP-P - Prob. Avg.	0.37 ± 0.037	0.28 ± 0.033	0.23 ± 0.015	0.20 ± 0.011	0.19 ± 0.013	0.18 ± 0.013
5-NN	DDPM - Log Prob. Avg.	0.24 ± 0.022	0.25 ± 0.013	0.24 ± 0.019	0.25 ± 0.025	0.25 ± 0.028	0.25 ± 0.029
	DDPM - Prob. Avg.	0.24 ± 0.022	0.23 ± 0.011	0.22 ± 0.010	0.22 ± 0.012	0.22 ± 0.013	0.22 ± 0.012
	SP-P - Log Prob. Avg.	0.23 ± 0.011	0.22 ± 0.019	0.21 ± 0.012	0.21 ± 0.010	0.20 ± 0.007	0.20 ± 0.014
	SP-P - Prob. Avg.	0.23 ± 0.011	0.20 ± 0.016	0.19 ± 0.011	0.18 ± 0.010	0.18 ± 0.007	0.18 ± 0.009
Decision Tree	DDPM - Log Prob. Avg.	0.44 ± 0.048	0.32 ± 0.045	0.31 ± 0.038	0.30 ± 0.030	0.29 ± 0.017	0.28 ± 0.032
	DDPM - Prob. Avg.	0.44 ± 0.048	0.32 ± 0.045	0.26 ± 0.025	0.24 ± 0.015	0.23 ± 0.008	0.22 ± 0.012
	SP-P - Log Prob. Avg.	0.33 ± 0.037	0.25 ± 0.018	0.25 ± 0.013	0.23 ± 0.024	0.21 ± 0.018	0.20 ± 0.018
	SP-P - Prob. Avg.	0.33 ± 0.037	0.25 ± 0.018	0.20 ± 0.014	0.19 ± 0.016	0.17 ± 0.013	0.16 ± 0.011
Random Forest	DDPM - Log Prob. Avg.	0.22 ± 0.014	0.21 ± 0.009	0.21 ± 0.009	0.21 ± 0.010	0.21 ± 0.010	0.21 ± 0.010
	DDPM - Prob. Avg.	0.22 ± 0.014	0.21 ± 0.009	0.21 ± 0.009	0.21 ± 0.010	0.21 ± 0.010	0.21 ± 0.009
	SP-P - Log Prob. Avg.	0.19 ± 0.006	0.18 ± 0.011	0.17 ± 0.009	0.17 ± 0.008	0.17 ± 0.007	0.17 ± 0.008
	SP-P - Prob. Avg.	0.19 ± 0.006	0.18 ± 0.011	0.17 ± 0.009	0.17 ± 0.008	0.17 ± 0.007	0.17 ± 0.007
MLP	DDPM - Log Prob. Avg.	0.33 ± 0.024	0.30 ± 0.016	0.30 ± 0.020	0.29 ± 0.024	0.29 ± 0.022	0.28 ± 0.015
	DDPM - Prob. Avg.	0.33 ± 0.024	0.28 ± 0.018	0.26 ± 0.013	0.25 ± 0.020	0.24 ± 0.019	0.24 ± 0.014
	SP-P - Log Prob. Avg.	0.27 ± 0.037	0.24 ± 0.030	0.21 ± 0.019	0.19 ± 0.016	0.19 ± 0.018	0.18 ± 0.016
	SP-P - Prob. Avg.	0.27 ± 0.037	0.21 ± 0.021	0.18 ± 0.009	0.17 ± 0.012	0.16 ± 0.013	0.16 ± 0.013
Gradient Boosting	DDPM - Log Prob. Avg.	0.23 ± 0.021	0.22 ± 0.012	0.22 ± 0.009	0.22 ± 0.009	0.22 ± 0.008	0.21 ± 0.012
	DDPM - Prob. Avg.	0.23 ± 0.021	0.22 ± 0.013	0.22 ± 0.008	0.22 ± 0.008	0.21 ± 0.007	0.21 ± 0.010
	SP-P - Log Prob. Avg.	0.20 ± 0.018	0.18 ± 0.017	0.17 ± 0.012	0.17 ± 0.014	0.17 ± 0.011	0.16 ± 0.012
	SP-P - Prob. Avg.	0.20 ± 0.018	0.18 ± 0.017	0.17 ± 0.012	0.17 ± 0.012	0.17 ± 0.010	0.16 ± 0.010
SVM	DDPM - Log Prob. Avg.	0.21 ± 0.010	0.21 ± 0.009	0.21 ± 0.010	0.21 ± 0.010	0.21 ± 0.010	0.21 ± 0.010
	DDPM - Prob. Avg.	0.21 ± 0.010	0.21 ± 0.009	0.21 ± 0.010	0.21 ± 0.010	0.21 ± 0.010	0.21 ± 0.010
	SP-P - Log Prob. Avg.	0.20 ± 0.008	0.18 ± 0.013	0.17 ± 0.013	0.17 ± 0.011	0.17 ± 0.008	0.17 ± 0.009
	SP-P - Prob. Avg.	0.20 ± 0.008	0.18 ± 0.014	0.17 ± 0.012	0.17 ± 0.010	0.17 ± 0.008	0.17 ± 0.008
Logistic Regression	DDPM - Log Prob. Avg.	0.23 ± 0.017	0.22 ± 0.008	0.21 ± 0.009	0.21 ± 0.014	0.22 ± 0.013	0.21 ± 0.013
	DDPM - Prob. Avg.	0.23 ± 0.017	0.22 ± 0.006	0.21 ± 0.007	0.21 ± 0.011	0.21 ± 0.010	0.21 ± 0.011
	SP-P - Log Prob. Avg.	0.20 ± 0.003	0.18 ± 0.011	0.17 ± 0.013	0.17 ± 0.011	0.17 ± 0.009	0.16 ± 0.010
	SP-P - Prob. Avg.	0.20 ± 0.003	0.18 ± 0.012	0.17 ± 0.013	0.17 ± 0.010	0.17 ± 0.008	0.16 ± 0.008

Table S11: Table of results from predicting MSE on the ACS 2018 dataset. The numbers are the mean measured and predicted MSEs ± standard deviations from 3 repeats. The nans for linear regression are due to excluding some repeats that had an extremely large MSE ($\geq 10^6$).

Downstream	Generator	m	1	2	4	8	16	32
Linear Regression	DDPM	Predicted	0.87 ± 0.008	0.87 ± 0.007	0.87 ± 0.008	0.88 ± 0.006	0.88 ± 0.007	0.88 ± 0.008
		Measured	0.87 ± 0.008	0.87 ± 0.007	0.87 ± 0.005	0.87 ± 0.001	0.87 ± 0.003	0.87 ± 0.003
	SP-P	Predicted	0.78 ± 0.007	0.78 ± 0.006	0.78 ± 0.006	0.78 ± nan	nan	nan
		Measured	0.78 ± 0.007	0.78 ± 0.006	0.78 ± 0.007	0.78 ± nan	nan	nan
Ridge Regression	DDPM	Predicted	0.87 ± 0.008	0.87 ± 0.007	0.87 ± 0.008	0.87 ± 0.009	0.87 ± 0.010	0.87 ± 0.010
		Measured	0.87 ± 0.008	0.87 ± 0.007	0.87 ± 0.005	0.87 ± 0.003	0.87 ± 0.004	0.87 ± 0.003
	SP-P	Predicted	0.78 ± 0.007	0.78 ± 0.006	0.78 ± 0.006	0.77 ± 0.006	0.77 ± 0.006	0.77 ± 0.006
		Measured	0.78 ± 0.007	0.78 ± 0.006	0.78 ± 0.007	0.78 ± 0.007	0.77 ± 0.007	0.77 ± 0.006
1-NN	DDPM	Predicted	1.55 ± 0.022	1.26 ± 0.026	1.11 ± 0.027	1.04 ± 0.028	1.01 ± 0.029	0.99 ± 0.029
		Measured	1.55 ± 0.022	1.26 ± 0.026	1.10 ± 0.019	1.03 ± 0.014	1.00 ± 0.013	0.98 ± 0.011
	SP-P	Predicted	1.38 ± 0.015	1.04 ± 0.010	0.87 ± 0.009	0.79 ± 0.010	0.75 ± 0.010	0.72 ± 0.010
		Measured	1.38 ± 0.015	1.04 ± 0.010	0.88 ± 0.001	0.79 ± 0.004	0.75 ± 0.006	0.73 ± 0.007
5-NN	DDPM	Predicted	0.98 ± 0.006	0.93 ± 0.009	0.91 ± 0.010	0.90 ± 0.011	0.89 ± 0.011	0.89 ± 0.012
		Measured	0.98 ± 0.006	0.93 ± 0.009	0.90 ± 0.008	0.89 ± 0.007	0.89 ± 0.004	0.88 ± 0.004
	SP-P	Predicted	0.93 ± 0.002	0.82 ± 0.005	0.77 ± 0.008	0.74 ± 0.009	0.73 ± 0.009	0.72 ± 0.010
		Measured	0.93 ± 0.002	0.82 ± 0.005	0.78 ± 0.004	0.75 ± 0.008	0.74 ± 0.008	0.73 ± 0.009
Decision Tree	DDPM	Predicted	1.41 ± 0.008	1.09 ± 0.005	0.94 ± 0.008	0.86 ± 0.009	0.82 ± 0.010	0.80 ± 0.011
		Measured	1.41 ± 0.008	1.09 ± 0.005	0.95 ± 0.006	0.87 ± 0.004	0.84 ± 0.006	0.82 ± 0.011
	SP-P	Predicted	1.25 ± 0.035	0.96 ± 0.023	0.82 ± 0.018	0.75 ± 0.017	0.72 ± 0.016	0.70 ± 0.016
		Measured	1.25 ± 0.035	0.96 ± 0.023	0.82 ± 0.011	0.74 ± 0.005	0.71 ± 0.003	0.69 ± 0.004
Random Forest	DDPM	Predicted	0.88 ± 0.010	0.82 ± 0.013	0.80 ± 0.015	0.78 ± 0.016	0.77 ± 0.017	0.77 ± 0.017
		Measured	0.88 ± 0.010	0.82 ± 0.013	0.80 ± 0.009	0.78 ± 0.009	0.78 ± 0.009	0.77 ± 0.009
	SP-P	Predicted	0.84 ± 0.010	0.76 ± 0.010	0.72 ± 0.010	0.70 ± 0.010	0.69 ± 0.011	0.68 ± 0.011
		Measured	0.84 ± 0.010	0.76 ± 0.010	0.72 ± 0.006	0.69 ± 0.005	0.68 ± 0.005	0.68 ± 0.006
Gradient Boosting	DDPM	Predicted	0.74 ± 0.008	0.74 ± 0.008	0.74 ± 0.008	0.74 ± 0.008	0.74 ± 0.008	0.74 ± 0.008
		Measured	0.74 ± 0.008	0.74 ± 0.008	0.74 ± 0.006	0.74 ± 0.005	0.74 ± 0.004	0.74 ± 0.004
	SP-P	Predicted	0.68 ± 0.005	0.68 ± 0.005	0.68 ± 0.005	0.67 ± 0.005	0.67 ± 0.005	0.67 ± 0.005
		Measured	0.68 ± 0.005	0.68 ± 0.005	0.67 ± 0.005	0.67 ± 0.005	0.67 ± 0.006	0.67 ± 0.006
MLP	DDPM	Predicted	0.82 ± 0.019	0.77 ± 0.018	0.75 ± 0.018	0.74 ± 0.019	0.74 ± 0.019	0.73 ± 0.019
		Measured	0.82 ± 0.019	0.77 ± 0.018	0.76 ± 0.017	0.75 ± 0.007	0.74 ± 0.005	0.74 ± 0.004
	SP-P	Predicted	0.76 ± 0.006	0.71 ± 0.003	0.68 ± 0.003	0.67 ± 0.003	0.66 ± 0.004	0.66 ± 0.004
		Measured	0.76 ± 0.006	0.71 ± 0.003	0.68 ± 0.008	0.67 ± 0.007	0.67 ± 0.007	0.66 ± 0.006
SVM	DDPM	Predicted	0.79 ± 0.004	0.78 ± 0.002	0.77 ± 0.004	0.77 ± 0.005	0.77 ± 0.005	0.77 ± 0.006
		Measured	0.79 ± 0.004	0.78 ± 0.002	0.77 ± 0.002	0.78 ± 0.002	0.78 ± 0.003	0.78 ± 0.002
	SP-P	Predicted	0.71 ± 0.005	0.69 ± 0.003	0.69 ± 0.002	0.68 ± 0.001	0.68 ± 0.001	0.68 ± 0.001
		Measured	0.71 ± 0.005	0.69 ± 0.003	0.69 ± 0.002	0.69 ± 0.004	0.69 ± 0.004	0.68 ± 0.004

Table S12: Table of results from predicting Brier score on the German Credit dataset. The numbers are the mean measured and predicted Brier scores ± standard deviations from 3 repeats.

Downstream	Generator	m	1	2	4	8	16	32
1-NN	DDPM	Measured	0.41 ± 0.027	0.33 ± 0.024	0.32 ± 0.032	0.30 ± 0.044	0.29 ± 0.036	0.28 ± 0.038
		Predicted	0.41 ± 0.027	0.33 ± 0.024	0.30 ± 0.022	0.28 ± 0.021	0.27 ± 0.021	0.27 ± 0.021
	SP-P	Measured	0.37 ± 0.033	0.28 ± 0.029	0.25 ± 0.030	0.23 ± 0.035	0.21 ± 0.035	0.20 ± 0.033
		Predicted	0.37 ± 0.033	0.28 ± 0.029	0.24 ± 0.037	0.22 ± 0.043	0.21 ± 0.046	0.20 ± 0.047
5-NN	DDPM	Measured	0.24 ± 0.020	0.24 ± 0.015	0.23 ± 0.019	0.23 ± 0.024	0.23 ± 0.025	0.23 ± 0.025
		Predicted	0.24 ± 0.020	0.24 ± 0.015	0.23 ± 0.017	0.23 ± 0.020	0.23 ± 0.021	0.23 ± 0.022
	SP-P	Measured	0.23 ± 0.009	0.21 ± 0.018	0.20 ± 0.017	0.19 ± 0.018	0.19 ± 0.015	0.19 ± 0.018
		Predicted	0.23 ± 0.009	0.21 ± 0.018	0.20 ± 0.024	0.20 ± 0.026	0.19 ± 0.028	0.19 ± 0.029
Decision Tree	DDPM	Measured	0.44 ± 0.043	0.32 ± 0.040	0.28 ± 0.038	0.27 ± 0.040	0.26 ± 0.037	0.25 ± 0.040
		Predicted	0.44 ± 0.043	0.32 ± 0.040	0.26 ± 0.039	0.23 ± 0.039	0.21 ± 0.038	0.20 ± 0.038
	SP-P	Measured	0.33 ± 0.033	0.25 ± 0.016	0.23 ± 0.027	0.21 ± 0.031	0.19 ± 0.026	0.18 ± 0.022
		Predicted	0.33 ± 0.033	0.25 ± 0.016	0.22 ± 0.007	0.20 ± 0.003	0.19 ± 0.001	0.18 ± 0.001
Random Forest	DDPM	Measured	0.22 ± 0.013	0.21 ± 0.008	0.21 ± 0.008	0.21 ± 0.009	0.21 ± 0.009	0.21 ± 0.009
		Predicted	0.22 ± 0.013	0.21 ± 0.008	0.21 ± 0.006	0.21 ± 0.005	0.21 ± 0.004	0.21 ± 0.004
	SP-P	Measured	0.19 ± 0.005	0.18 ± 0.010	0.17 ± 0.008	0.17 ± 0.007	0.17 ± 0.006	0.17 ± 0.007
		Predicted	0.19 ± 0.005	0.18 ± 0.010	0.17 ± 0.012	0.17 ± 0.014	0.17 ± 0.014	0.17 ± 0.015
MLP	DDPM	Measured	0.33 ± 0.022	0.29 ± 0.019	0.28 ± 0.025	0.27 ± 0.032	0.26 ± 0.030	0.26 ± 0.026
		Predicted	0.33 ± 0.022	0.29 ± 0.019	0.28 ± 0.023	0.27 ± 0.026	0.26 ± 0.027	0.26 ± 0.028
	SP-P	Measured	0.27 ± 0.033	0.23 ± 0.027	0.19 ± 0.022	0.18 ± 0.019	0.18 ± 0.020	0.17 ± 0.018
		Predicted	0.27 ± 0.033	0.23 ± 0.027	0.21 ± 0.044	0.20 ± 0.054	0.19 ± 0.059	0.19 ± 0.061
Gradient Boosting	DDPM	Measured	0.23 ± 0.019	0.22 ± 0.011	0.22 ± 0.008	0.22 ± 0.008	0.22 ± 0.007	0.21 ± 0.010
		Predicted	0.23 ± 0.019	0.22 ± 0.011	0.22 ± 0.008	0.22 ± 0.007	0.22 ± 0.007	0.22 ± 0.007
	SP-P	Measured	0.20 ± 0.016	0.18 ± 0.016	0.17 ± 0.011	0.17 ± 0.012	0.17 ± 0.009	0.16 ± 0.010
		Predicted	0.20 ± 0.016	0.18 ± 0.016	0.17 ± 0.018	0.17 ± 0.020	0.16 ± 0.021	0.16 ± 0.022
SVM	DDPM	Measured	0.21 ± 0.009	0.21 ± 0.008	0.21 ± 0.009	0.21 ± 0.009	0.21 ± 0.009	0.21 ± 0.009
		Predicted	0.21 ± 0.009	0.21 ± 0.008	0.21 ± 0.008	0.21 ± 0.008	0.21 ± 0.008	0.21 ± 0.008
	SP-P	Measured	0.20 ± 0.007	0.18 ± 0.012	0.17 ± 0.011	0.17 ± 0.010	0.17 ± 0.007	0.17 ± 0.008
		Predicted	0.20 ± 0.007	0.18 ± 0.012	0.17 ± 0.016	0.17 ± 0.018	0.17 ± 0.019	0.17 ± 0.020
Logistic Regression	DDPM	Measured	0.23 ± 0.015	0.22 ± 0.006	0.21 ± 0.007	0.21 ± 0.011	0.21 ± 0.011	0.21 ± 0.011
		Predicted	0.23 ± 0.015	0.22 ± 0.006	0.21 ± 0.003	0.21 ± 0.003	0.21 ± 0.004	0.21 ± 0.004
	SP-P	Measured	0.20 ± 0.003	0.18 ± 0.010	0.17 ± 0.012	0.17 ± 0.009	0.17 ± 0.008	0.16 ± 0.008
		Predicted	0.20 ± 0.003	0.18 ± 0.010	0.17 ± 0.017	0.17 ± 0.020	0.16 ± 0.021	0.16 ± 0.022

Table S13: Table of results from the DP experiment. The numbers are the mean Brier score ± the standard deviation from 3 repeats. The privacy parameters are $\epsilon = 1.5$, $\delta = n^{-2} \approx 4.7 \cdot 10^{-7}$.

Downstream	m Generator	1	2	4	8	16	32
1-NN	AIM - Log Prob. Avg.	0.23 ± 0.012	0.17 ± 0.003	0.16 ± 0.002	0.16 ± 0.011	0.15 ± 0.006	0.15 ± 0.001
	AIM - Prob. Avg.	0.23 ± 0.012	0.17 ± 0.003	0.14 ± 0.003	0.13 ± 0.004	0.12 ± 0.002	0.12 ± 0.001
	NAPSU-MQ - Log Prob. Avg.	0.25 ± 0.008	0.20 ± 0.020	0.19 ± 0.005	0.17 ± 0.006	0.16 ± 0.005	0.16 ± 0.004
	NAPSU-MQ - Prob. Avg.	0.25 ± 0.008	0.20 ± 0.020	0.16 ± 0.003	0.14 ± 0.005	0.13 ± 0.003	0.13 ± 0.003
5-NN	AIM - Log Prob. Avg.	0.13 ± 0.004	0.13 ± 0.002	0.14 ± 0.002	0.14 ± 0.002	0.13 ± 0.004	0.13 ± 0.001
	AIM - Prob. Avg.	0.13 ± 0.004	0.13 ± 0.003	0.12 ± 0.002	0.12 ± 0.001	0.12 ± 0.002	0.12 ± 0.001
	NAPSU-MQ - Log Prob. Avg.	0.16 ± 0.007	0.15 ± 0.004	0.15 ± 0.008	0.15 ± 0.008	0.14 ± 0.003	0.14 ± 0.003
	NAPSU-MQ - Prob. Avg.	0.16 ± 0.007	0.14 ± 0.005	0.13 ± 0.005	0.13 ± 0.005	0.13 ± 0.004	0.13 ± 0.004
Decision Tree	AIM - Log Prob. Avg.	0.13 ± 0.003	0.13 ± 0.002	0.13 ± 0.002	0.13 ± 0.002	0.13 ± 0.002	0.12 ± 0.001
	AIM - Prob. Avg.	0.13 ± 0.003	0.12 ± 0.002	0.12 ± 0.001	0.12 ± 0.001	0.12 ± 0.001	0.12 ± 0.001
	NAPSU-MQ - Log Prob. Avg.	0.16 ± 0.005	0.15 ± 0.004	0.15 ± 0.005	0.14 ± 0.004	0.14 ± 0.004	0.14 ± 0.003
	NAPSU-MQ - Prob. Avg.	0.16 ± 0.005	0.14 ± 0.004	0.13 ± 0.004	0.13 ± 0.004	0.13 ± 0.004	0.13 ± 0.004
Random Forest	AIM - Log Prob. Avg.	0.12 ± 0.002	0.12 ± 0.002	0.12 ± 0.001	0.12 ± 0.002	0.12 ± 0.002	0.12 ± 0.001
	AIM - Prob. Avg.	0.12 ± 0.002	0.12 ± 0.002	0.12 ± 0.001	0.12 ± 0.001	0.12 ± 0.001	0.12 ± 0.001
	NAPSU-MQ - Log Prob. Avg.	0.14 ± 0.007	0.14 ± 0.004	0.13 ± 0.004	0.13 ± 0.004	0.13 ± 0.004	0.13 ± 0.004
	NAPSU-MQ - Prob. Avg.	0.14 ± 0.007	0.13 ± 0.004	0.13 ± 0.003	0.13 ± 0.004	0.13 ± 0.003	0.13 ± 0.003
MLP	AIM - Log Prob. Avg.	0.12 ± 0.002	0.12 ± 0.002	0.12 ± 0.002	0.12 ± 0.001	0.12 ± 0.002	0.12 ± 0.001
	AIM - Prob. Avg.	0.12 ± 0.002	0.12 ± 0.002	0.11 ± 0.001	0.11 ± 0.001	0.11 ± 0.001	0.11 ± 0.001
	NAPSU-MQ - Log Prob. Avg.	0.14 ± 0.006	0.13 ± 0.004	0.13 ± 0.005	0.13 ± 0.004	0.13 ± 0.004	0.13 ± 0.004
	NAPSU-MQ - Prob. Avg.	0.14 ± 0.006	0.13 ± 0.004	0.13 ± 0.004	0.13 ± 0.004	0.13 ± 0.004	0.12 ± 0.004
Grad. Boosting	AIM - Log Prob. Avg.	0.11 ± 0.002	0.11 ± 0.001	0.11 ± 0.001	0.12 ± 0.001	0.12 ± 0.001	0.12 ± 0.001
	AIM - Prob. Avg.	0.11 ± 0.002	0.11 ± 0.001	0.11 ± 0.001	0.12 ± 0.001	0.12 ± 0.001	0.12 ± 0.001
	NAPSU-MQ - Log Prob. Avg.	0.13 ± 0.004	0.13 ± 0.003	0.13 ± 0.004	0.13 ± 0.003	0.13 ± 0.003	0.13 ± 0.004
	NAPSU-MQ - Prob. Avg.	0.13 ± 0.004	0.13 ± 0.003	0.13 ± 0.004	0.13 ± 0.003	0.13 ± 0.003	0.13 ± 0.004
SVM	AIM - Log Prob. Avg.	0.13 ± 0.002	0.13 ± 0.002	0.13 ± 0.002	0.13 ± 0.001	0.13 ± 0.002	0.13 ± 0.002
	AIM - Prob. Avg.	0.13 ± 0.002	0.13 ± 0.002	0.13 ± 0.002	0.13 ± 0.001	0.13 ± 0.002	0.13 ± 0.002
	NAPSU-MQ - Log Prob. Avg.	0.14 ± 0.005	0.14 ± 0.006	0.14 ± 0.005	0.14 ± 0.004	0.13 ± 0.005	0.13 ± 0.005
	NAPSU-MQ - Prob. Avg.	0.14 ± 0.005	0.14 ± 0.006	0.14 ± 0.005	0.14 ± 0.004	0.13 ± 0.004	0.13 ± 0.005
Log. Regression	AIM - Log Prob. Avg.	0.12 ± 0.002	0.12 ± 0.001	0.12 ± 0.002	0.12 ± 0.001	0.12 ± 0.002	0.12 ± 0.001
	AIM - Prob. Avg.	0.12 ± 0.002	0.12 ± 0.001	0.12 ± 0.002	0.12 ± 0.001	0.12 ± 0.001	0.12 ± 0.001
	NAPSU-MQ - Log Prob. Avg.	0.13 ± 0.005	0.13 ± 0.003	0.13 ± 0.004	0.13 ± 0.004	0.13 ± 0.004	0.13 ± 0.004
	NAPSU-MQ - Prob. Avg.	0.13 ± 0.005	0.13 ± 0.003	0.13 ± 0.004	0.13 ± 0.004	0.13 ± 0.004	0.13 ± 0.004

E.4 Estimating Model and Synthetic Data Variances

In this section, we estimate the MV and SDV terms from the decomposition in Theorem 2.1. We first generate 32 synthetic datasets that are 5 times larger than the real dataset, and split each synthetic datasets into 5 equally-sized subsets. This is equivalent to training 32 generators, with parameters θ_i, and for each generator, generating 5 synthetic datasets i.i.d. We then train the downstream predictor on each synthetic dataset, and store the predictions for all test points.

To estimate MV, we compute the sample variance over the 5 synthetic datasets generated from the same θ_i, and then compute the mean over the 32 different θ_i values. To estimate SDV, we compute the sample mean over the 5 synthetic datasets from the same θ_i, and compute the sample variance over the 32 different θ_i values.

The result is an estimate of MV and SDV for each test point. We plot the mean over the test points. The whole experiment is repeated 3 times, with different train-test splits. The datasets, train-test splits, and downstream predictors are the same as in the other experiments, described in Appendix D.

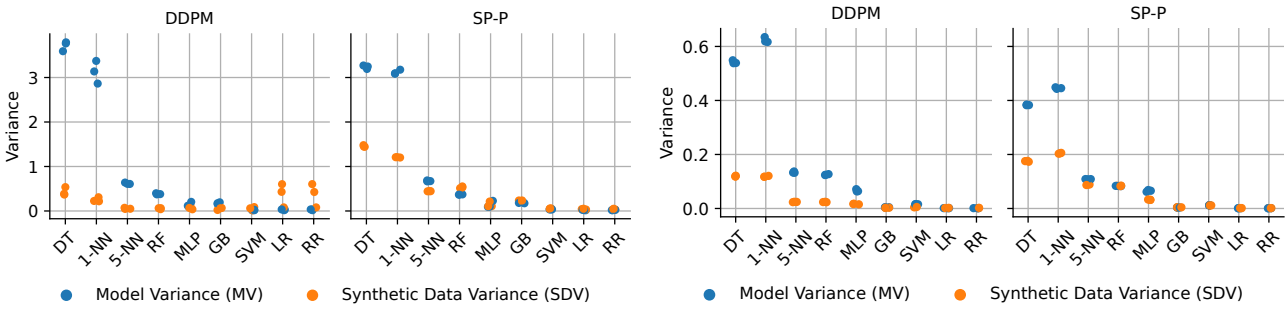
The results are in Figure S11. MV depends mostly on the downstream predictor, while SDV also depends on the synthetic data generator. We also confirm that decision trees and 1-NN have much higher variance than the other models, and linear, ridge and logistic regression have a very low variance.

E.5 Comparison with Generating A Single Large Synthetic Dataset

In this section, we examine an alternative to multiple synthetic datasets: generating a single, large synthetic dataset. van Breugel et al. (2023a) found this lead to poor model evaluation and selection, with a single synthetic dataset leading to overestimating the performance of complex models. They did not directly compare the effect on accuracy, which is what we will do in this section.

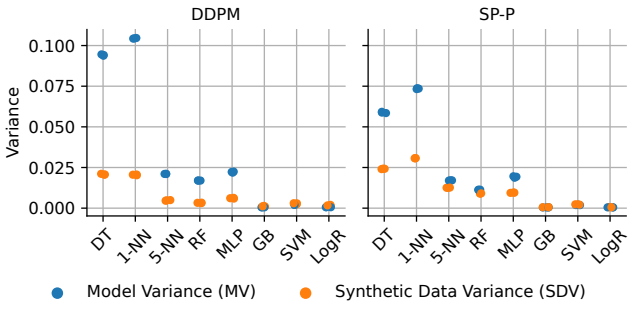
We use the synthetic datasets from the variance estimation experiment[14] of Section E.4 that are 5 times larger than the real dataset. We also take smaller subsets of the whole synthetic dataset to examine other synthetic dataset sizes. We train the same models as in the other experiments, but only on regression datasets generated by synthpop (proper).

[14]We use only one of the 32 synthetic datasets per repeat.

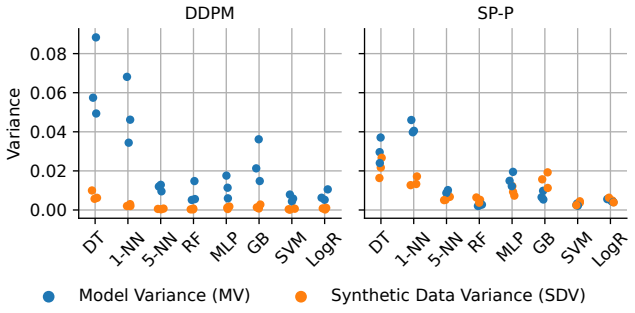


(a) Abalone

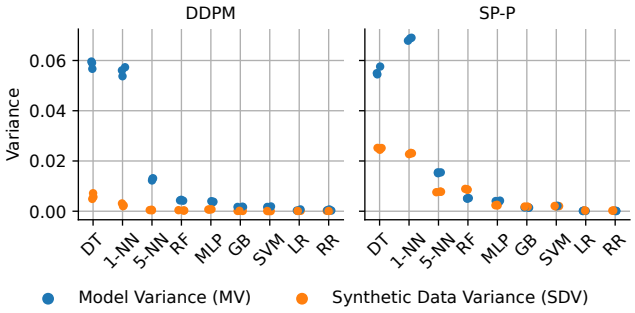
(b) ACS 2018



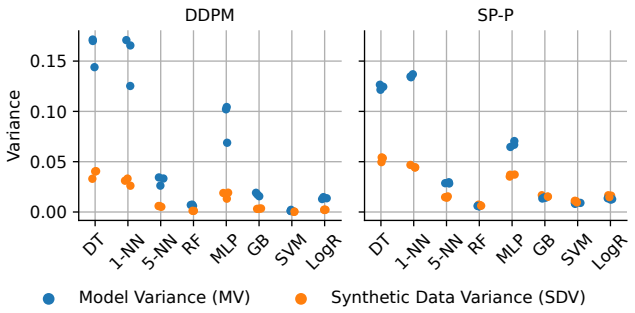
(c) Adult



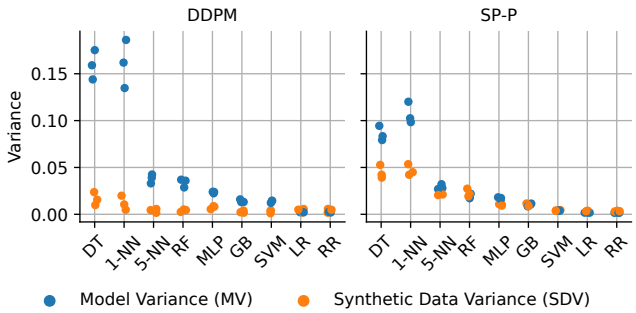
(d) Breast Cancer



(e) California Housing



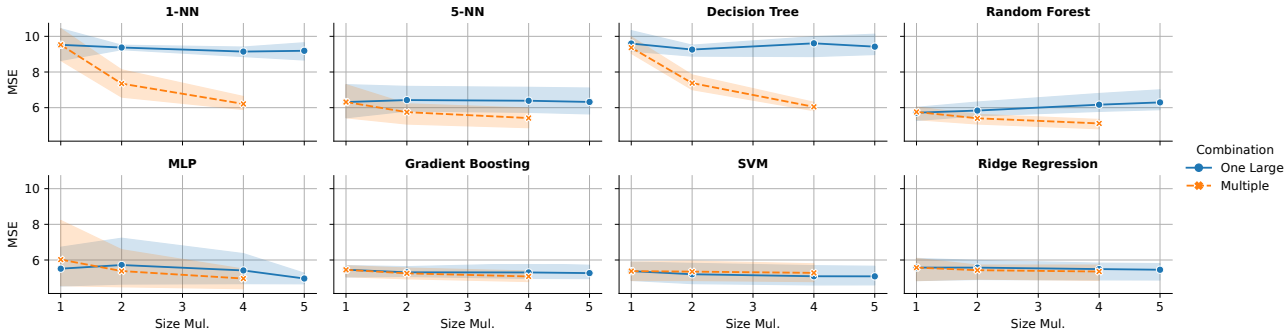
(f) German Credit



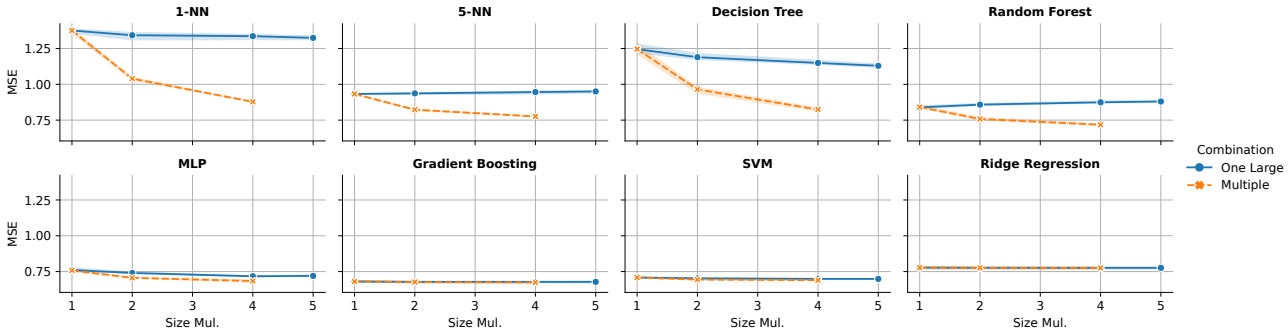
(g) Insurance

Figure S11: Estimating the MV and SDV terms from the decomposition. Decision trees have high variances on all datasets, while linear, ridge and logistic regression have low variances. MV depends mostly on the predictor, while SDV depends on both the predictor and synthetic data generation algorithm. The points are the averages of estimated MV and SDV, averaged over the test data, from 3 repeats with different train-test splits. We excluded some test data points that had extremely large variance estimates ($\geq 10^6$) for linear regression on the ACS 2018 dataset. The predictors are decision tree (DT), nearest neighbours with 1 or 5 neighbours (1-NN and 5-NN), random forest (RF), a multilayer perceptron (MLP), gradient boosted trees (GB), a support vector machine (SVM), linear regression (LR), ridge regression (RR) and logistic regression (LogR). The synthetic data generators are DDPM and synthpop (SP-P).

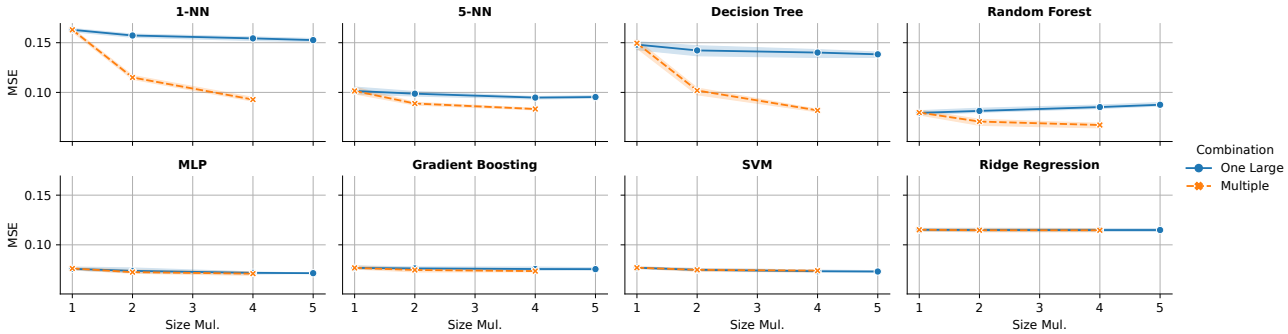
We compare the results from the single large synthetic dataset with the ensemble of multiple synthetic dataset with an equal number of total datapoints in Figure S12. We see that the ensemble of multiple synthetic datasets is equal or better in all cases in terms of MSE.



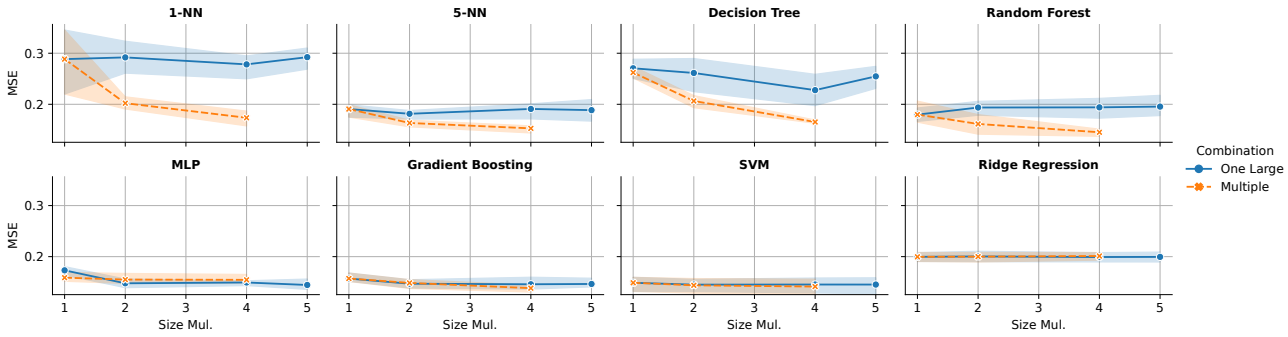
(a) Abalone



(b) ACS 2018



(c) California Housing



(d) Insurance

Figure S12: Comparison of the ensemble of multiple synthetic datasets with one large synthetic dataset with an equal number of synthetic datapoints on the regression datasets with synthpop. We see that multiple synthetic datasets are always equal or better. Size Mul. is the relative total number of synthetic data points to the real data: for multiple synthetic datasets, it is m, and for a single large synthetic dataset it is n_{Syn}/n_{Real}.