ESCAPING ITERATIVE PARAMETER-SPACE NOISE: DIFFERENTIALLY PRIVATE LEARNING WITH A HYPERNETWORK

Anonymous authors
Paper under double-blind review

ABSTRACT

Differentially private (DP) training of neural networks is often hindered by the large amount of noise required by gradient-based methods such as DP-SGD, which repeatedly inject high-dimensional noise in parameter space throughout training. In this paper, we propose a new framework for DP learning that avoids iterative optimization in parameter space. Instead of updating the target model using privatized gradients, we employ a *hypernetwork trained on public datasets* to map a private dataset to the parameters of the target model. Specifically, each example is embedded into a low-dimensional representation, the embeddings are aggregated and perturbed to obtain a DP dataset embedding, and the hypernetwork generates the target model parameters from this noisy embedding. Because privacy noise is injected only once into a low-dimensional dataset representation, our approach can significantly reduce the adverse effect of noise. We theoretically show in a synthetic setting that, under a fixed privacy budget, models produced by our approach achieve higher utility than those trained with DP-SGD. Moreover, we apply our approach to LoRA fine-tuning of diffusion models and show that it achieves lower FID than LoRA models trained with DP-SGD and other public-data-guided methods.

1 INTRODUCTION

Deep neural networks are often trained on data that include private information. Trained models can leak private information through training-data extraction (Carlini et al., 2023) and membership inference (Wu et al., 2022), motivating privacy protection when releasing neural network parameters. One approach is to ensure that the training algorithm for neural networks satisfies *differential privacy (DP)*, a mathematical framework for quantifying privacy leakage from a dataset. The most widely used approach is DP-SGD (Abadi et al., 2016), which protects the privacy of each training example by adding noise at each step of SGD. Several other gradient-based methods have also been proposed, such as DP-Adam (Zhou et al., 2020) and DP-AdamW (Chooi et al., 2025).

Training neural networks with these gradient-based DP learning methods can incur a substantial utility loss compared with non-private optimization methods (Tramer & Boneh, 2021). A major source of this degradation is the amount of noise required for privacy. In these methods, DP noise is repeatedly injected at every optimization step. Moreover, DP noise must be added in the gradient space, whose dimension is equal to the number of model parameters. Under standard composition-based calibration with a fixed privacy budget and clipping threshold, the root-mean-square norm of the noise added at each step grows with the dimension and number of steps. This issue becomes particularly severe in small-data regimes, such as when finetuning a large model on a limited amount of private data. These observations naturally lead to the following question:

Can we obtain high-utility models under differential privacy without repeatedly injecting parameter-dimensional noise?

As a positive answer to this question, we propose a new framework that directly generates differentially private parameters using a hypernetwork. Instead of updating parameters using gradients computed from a private dataset, we feed the private dataset into a hypernetwork and obtain parameters adapted to the dataset through a single forward pass. To this end, we propose *DP-DeepSets*, a hypernetwork

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

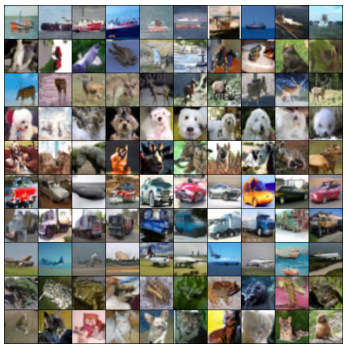

Figure 1: Generated samples on CIFAR-10 with $(\epsilon, \delta) = (2, 10^{-5})$. Each row corresponds to one k-NN set.

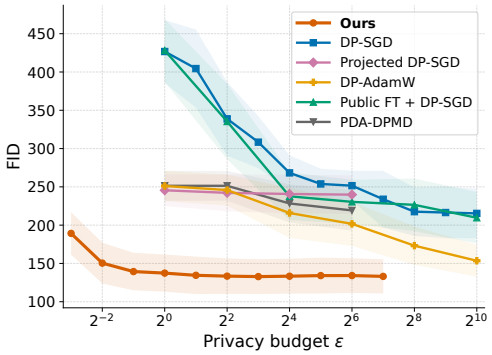

Figure 2: Change in FID with respect to the DP parameter ϵ for each method.

architecture for DP learning that takes a dataset as input and outputs parameters with DP guarantees. This architecture first embeds each data point into a low-dimensional representation, then applies mean pooling and adds DP noise to obtain a differentially private dataset embedding. The parameters are generated from this embedding, and the resulting network inherits the DP guarantee.

Our method has three key advantages by design. First, *DP perturbation occurs only once during parameter generation*. Unlike gradient-based DP optimization, which injects noise at every step, our method adds DP noise only once, to the dataset embedding. Second, *DP noise is injected into a low-dimensional dataset embedding*. In contrast to such optimization, which adds noise to high-dimensional gradients in parameter space, our approach requires a much smaller overall noise magnitude. Consequently, the performance of the learned model is less susceptible to degradation caused by DP noise. Third, *the hypernetwork itself is trained on public data*. Since the hypernetwork is public, the privacy budget is consumed only when constructing the low-dimensional dataset embedding during parameter generation from private data. Moreover, training the hypernetwork incurs no privacy cost, allowing it to learn the relationship between datasets and parameters from a large amount of public data, and thereby acquire a form of "learning to learn."

Our contributions are summarized as follows:

1. We propose a novel framework for DP learning of neural networks using public hypernetworks. Specifically, we develop *DP-DeepSets*, which takes a dataset as input and returns parameters with a DP guarantee.
2. We theoretically compare DP learning using DP-DeepSets with DP-SGD. We consider a synthetic setting involving linear regression, and show that DP-DeepSets achieves high utility since it learns to capture the low-dimensional structure of tasks during training on public data.
3. We apply our framework to LoRA finetuning of diffusion models. We demonstrate that the model trained by our method outperforms DP-SGD and other public-data-guided DP learning methods.

We present results for LoRA finetuning of diffusion models in Figures 1 and 2. The images in Figure 1 are generated by applying LoRA parameters obtained from only 128 private data points via the trained hypernetwork. The generated images capture the characteristics of each CIFAR-10 class. Furthermore, Figure 2 shows that our method outperforms DP-SGD and other public-data-guided methods. See Section 5 for detailed settings and additional observations.

Related work Hypernetworks (Ha et al., 2017) and subsequent generative approaches (Peebles et al., 2022; Wang et al., 2024) generate neural network parameters. Recent methods condition parameter generation on task prompts (Jin et al., 2024; Wang et al., 2025; Khan et al., 2025) or datasets (Ruiz et al., 2024; Soro et al., 2025; Liang et al., 2025). Building on these studies, we develop a dataset-conditioned hypernetwork architecture that replaces gradient-based DP learning by directly generating parameters with a DP guarantee.

AdaDPS (Li et al., 2022) and PDA-DPMD (Amid et al., 2022) use public data to guide gradient-based optimization, including Adam and mirror descent, but still add noise in the full parameter space.

Other approaches reduce gradient dimensionality through subspace projection using public data (Yu et al., 2021; Zhou et al., 2021) or training trajectories (Zheng et al., 2026), or through sparsification with random masks (Zhu & Blaschko, 2023). All of these gradient-based methods still inject noise at every optimization step. PATE (Papernot et al., 2017) and its variants (Papernot et al., 2018) use teachers trained on private data to label unlabeled public data for classification. Our framework instead uses public tasks to train a hypernetwork that generates parameters from private datasets for tasks including generative modeling.

Among public-data-guided methods, the closest to ours is Räisä et al. (2024), which generates parameters by feeding a private dataset into a network called SetConv trained on simulated data. Although the idea is related, their experiments are limited to regression with one-dimensional inputs, whereas we address high-dimensional tasks. This difference motivates a different hypernetwork design: SetConv applies a fixed kernel function to each data point and aggregates the resulting representations, but such fixed-kernel data encoding does not work well for high-dimensional data. Our method instead uses a DeepSets-based architecture with a learnable feature map to embed data points, allowing the hypernetwork to learn how to efficiently encode high-dimensional data.

DP-MERF (Harder et al., 2021) perturbs a mean embedding of fixed random Fourier features and trains a generator to match it. VP-NTK (Hsu et al., 2025) combines privatized neural tangent kernel mean embeddings with visual prompting, keeping its pretrained generator and feature extractor frozen. These methods reuse privatized statistics during optimization without additional privacy cost. Our framework jointly learns a compact dataset encoder and a parameter generator on public tasks using the target model's task loss, accommodating objectives such as diffusion denoising. This allows the representation to focus on task-relevant information, which can improve the privacy–utility trade-off when high-dimensional tasks have low-dimensional structure.

2 Background

2.1 Differential Privacy

Differential Privacy (DP) (Dwork et al., 2006) is a mathematical framework for quantifying and limiting the information about any individual data point that can be inferred from the output of an algorithm. Let $\mathcal{D}^*$ denote the space of datasets. Two datasets $D, D' \in \mathcal{D}^*$ are said to be adjacent if they differ in a single entry. We say that a randomized algorithm $\mathcal{M} : \mathcal{D}^* \to \mathcal{Y}$ satisfies (ϵ, δ)-DP if, for any pair of adjacent datasets $D, D' \in \mathcal{D}^*$ and any measurable set $S \subseteq \mathcal{Y}$, it holds that

$$\mathbb{P}(\mathcal{M}(D) \in S) \leq \mathrm{e}^{\epsilon} \cdot \mathbb{P}(\mathcal{M}(D') \in S) + \delta.$$

In other words, (ϵ, δ)-DP requires that the output distribution does not change substantially between adjacent datasets.

For a query that takes a dataset as input and outputs a real-valued quantity, a common way to guarantee differential privacy is to add Gaussian noise to the output. This is called the Gaussian mechanism, and it provides the following differential privacy guarantee.

Proposition 1 (Gaussian mechanism, Dwork & Roth (2014))**.** *For a given query $q : \mathcal{D}^* \to \mathbb{R}^d$, we define the s^2-*Gaussian mechanism $\mathcal{M}_q : \mathcal{D}^* \to \mathbb{R}^d$ *as $\mathcal{M}_q(D) := z + q(D)$ where $z \sim \mathcal{N}(0, s^2 I_d)$. Let $\Delta := \max_{D \sim D'} \|q(D) - q(D')\|_2$. Then, for any $\epsilon, \delta \in (0, 1)$, the s^2-Gaussian mechanism with $s^2 = 2\log\big(1.25\delta^{-1}\big)(\Delta/\epsilon)^2$ satisfies (ϵ, δ)-differential privacy.*

2.2 Differentially Private SGD (DP-SGD)

For training a machine learning model on a private dataset, one of the most widely used algorithms for ensuring DP of a trained model is Differentially Private SGD (Abadi et al., 2016). In this algorithm, Gaussian noise is added when computing the average of the gradients in a minibatch. More specifically, at each step, DP-SGD includes each record independently in a minibatch $\mathcal{B}$ with probability $q = b/N$, where b is the expected batch size, computes the corresponding per-example gradients, and updates the parameters as follows:

$$\omega \leftarrow \omega - \frac{\eta}{b}\left(\sum_{n \in \mathcal{B}} \mathrm{clip}_C(g_n(\omega)) + \xi\right), \quad \xi \sim \mathcal{N}(0, s^2 I_{d_\omega}) \tag{1}$$

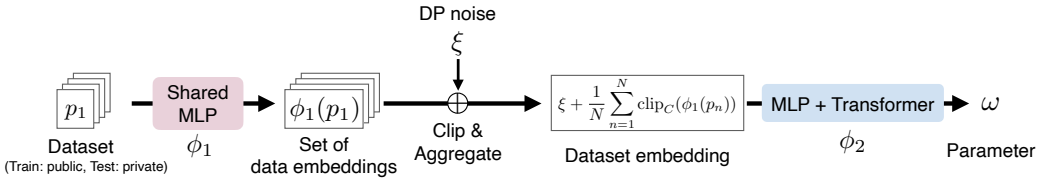

Figure 3: Overview of DP-DeepSets. First, we obtain embeddings of the data points in the input dataset using an MLP ϕ_1, and clip them in L^2-norm. Then, we take the element-wise average and perturb it with Gaussian noise. As a result, we obtain a dataset embedding with a DP guarantee, and feed it into the latter network ϕ_2 to generate the network parameter.

where $\mathrm{clip}_C(x) = \min\left\{1, \frac{C}{\|x\|}\right\}x$, $\eta > 0$ is a learning rate and d_ω is the dimension of the parameter ω. DP-SGD satisfies the following differential privacy guarantee.

Proposition 2 (Theorem 1 of Abadi et al. (2016))**.** *There exist constants c_1, c_2 such that, given the number of steps T, for any $\epsilon < c_1 \frac{b^2 T}{N^2}$, DP-SGD is (ϵ, δ)-differentially private for any $\delta > 0$, if we choose $s^2 \geq c_2 \frac{C^2 b^2 T \log \delta^{-1}}{\epsilon^2 N^2}$.*

3 Framework: Differentially Private Learning with a Public Hypernetwork

We introduce our framework for training hypernetworks on public data to generate differentially private parameters for target networks. We consider a neural network ψ_ω with parameters $\omega \in \mathbb{R}^{d_\omega}$ for solving a given task, and refer to it as the *target network*. Let $p \in \mathbb{R}^{d_p}$ denote a data point and $u \in \mathbb{R}^{d_u}$ auxiliary variables used to define the network input and loss. We denote the per-data-point loss by $\ell : \mathbb{R}^{d_p} \times \mathbb{R}^{d_u} \times \mathbb{R}^{d_\omega} \to \mathbb{R}$.

For regression, $p = (x, y)$, u may be omitted, and $\ell(p, u; \omega) = \frac{1}{2}(\psi_\omega(x) - y)^2$. In generative modeling, such as autoregressive language models and diffusion models, p is the sample itself. In autoregressive language models, u may specify the prediction position or context prefix. In diffusion models, u may include the noise and timestep, and ℓ is typically the mean-squared error for noise prediction.

Our goal is to obtain a hypernetwork ϕ that takes a dataset $p_{1:N}$ and DP noise ξ as input and generates parameters that satisfy an (ϵ, δ)-DP guarantee and yield a small test loss $\mathbb{E}_{p,u,\xi}[\ell(p, u; \phi(p_{1:N}, \xi))]$. To this end, we develop *DP-DeepSets*, a hypernetwork architecture that takes the dataset as input and predicts parameters with a DP guarantee. Moreover, we establish a training procedure for DP-DeepSets using a public dataset.

3.1 DP-DeepSets: A Hypernetwork Architecture for Differentially Private Learning

Here, we describe the mechanism of *DP-DeepSets*. Its overview is also shown in Figure 3.

DP-DeepSets takes a dataset, i.e., a set of data points $p_{1:N} := \{p_n\}_{n=1}^N$ as input. It first obtains a set of data embeddings $\{\phi_1(p_n)\}_{n=1}^N \subset \mathbb{R}^D$ using an MLP $\phi_1 : \mathbb{R}^{d_p} \to \mathbb{R}^D$ shared among the data points. The encoder is applied independently to each example and contains no cross-example operations such as batch normalization. Next, each embedding is clipped to ensure that its norm is at most $C > 0$. This procedure is required to limit privacy leakage from each data point. Then, to aggregate the information in the entire dataset, we take the average of the clipped embeddings. Finally, we add Gaussian noise $\xi \sim \mathcal{N}(0, s_{\mathrm{DP}}^2 I_D)$ for some s_{DP} and guarantee that the resulting embedding is differentially private. As a result, we obtain the following noisy dataset embedding:

$$E(p_{1:N}, \xi) := \xi + \frac{1}{N}\sum_{n=1}^{N} \mathrm{clip}_C(\phi_1(p_n)).$$

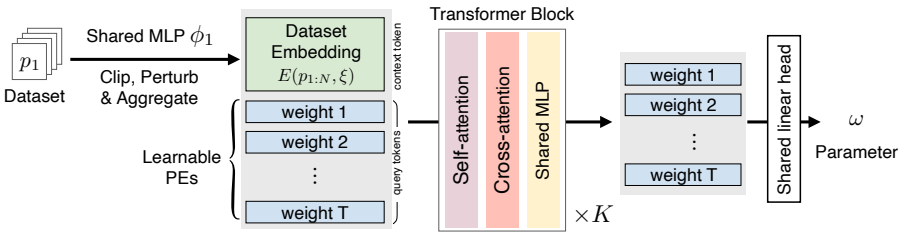

Figure 4: The architecture of the transformer-based decoder ϕ_2 in DP-DeepSets. The dataset embedding is used as a context token. The target parameter vector is partitioned into T segments, each associated with a query token initialized as a learnable positional embedding. They are fed into the K-layer Transformer consisting of self-attention, cross-attention and shared MLPs. The output tokens are fed into the shared linear head, and are reshaped to integrate into the model parameters.

Using a dataset embedding $E(p_{1:N}, \xi)$, we can generate the network parameters as $\omega = \phi_2(E(p_{1:N}, \xi))$ with some neural network $\phi_2 : \mathbb{R}^D \to \mathbb{R}^{d_\omega}$. Inspired by Ruiz et al. (2024), we use a Transformer-based architecture for ϕ_2 as in Figure 4. Specifically, the dataset embedding is fed into a transformer as a context token. The target parameter vector is partitioned into T segments, each associated with a query token initialized as a learnable positional embedding. Then, they are iteratively updated through multiple transformer blocks (namely, self-attention among the query tokens, cross-attention between the query tokens and the context token, and token-wise shared MLPs). This allows the hypernetwork not merely to generate the parameters at each position independently of the dataset embedding, but also to model the interactions among parameters in different layers.

The architecture above can be seen as a *differentially private variant* of DeepSets (Zaheer et al., 2017). DeepSets uses operations such as taking the maximum, summation, and average of the embeddings obtained by a neural network to aggregate the information of sets. Building upon this idea, we adopt the average operation, and we further incorporate the clipping and Gaussian perturbation to guarantee DP. This allows us to apply the theoretical guarantee for the Gaussian mechanism for the mean query and obtain the following rigorous DP guarantee (see Appendix A for the proof):

Corollary 3. *Let $\epsilon, \delta \in (0, 1)$. If $s_{\mathrm{DP}}^2 \geq \frac{8C^2 \log\left(1.25\delta^{-1}\right)}{\epsilon^2 N^2}$, then DP-DeepSets is (ϵ, δ)-differentially private.*

3.2 Training of a Public Hypernetwork

To learn the mapping from datasets to suitable target-network parameters, we train DP-DeepSets on public datasets (Figure 5). Specifically, we prepare a collection of public datasets $\left\{(p_{1:N}^{(1)}, u_{1:N}^{(1)}), \ldots, (p_{1:N}^{(M)}, u_{1:N}^{(M)})\right\}$. We also independently sample noise vectors $\xi^{(1)}, \ldots, \xi^{(M)} \sim \mathcal{N}(0, s_{\mathrm{DP}}^2 I_D)$. Let ϕ_θ denote a DP-DeepSets hypernetwork parameterized by θ. We train it using the following loss function $\mathcal{L}_{\mathrm{HN}}(\theta)$:

$$\mathcal{L}_{\mathrm{HN}}(\theta) := \frac{1}{M} \sum_{m=1}^{M} \mathcal{L}_{\mathrm{dataset}}^{(m)}(\theta), \text{ where } \mathcal{L}_{\mathrm{dataset}}^{(m)}(\theta) := \frac{1}{N} \sum_{n=1}^{N} \ell\Big(p_n^{(m)}, u_n^{(m)}; \phi_\theta(p_{1:N}^{(m)}, \xi^{(m)})\Big). \quad (2)$$

Here, $\mathcal{L}_{\mathrm{HN}}$ is the average loss over the M public datasets. Each $\mathcal{L}_{\mathrm{dataset}}^{(m)}$ is the standard empirical risk evaluated at the generated target-network parameters $\omega^{(m)} = \phi_\theta(p_{1:N}^{(m)}, \xi^{(m)})$. We optimize θ end-to-end through these generated parameters, so the hypernetwork learns to generate suitable parameters for each dataset.

Remark 4. Because DP-DeepSets is trained on public data, it contains no private information. Privacy protection therefore concerns only the parameters generated from private data. We nevertheless perturb dataset embeddings during public training to match the noise distribution used at inference.

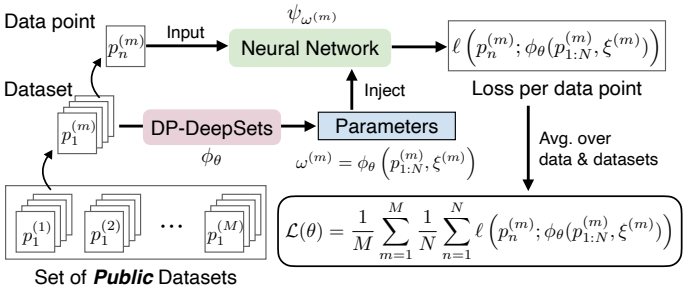

Figure 5: Training DP-DeepSets on public datasets. The hypernetwork generates target-network parameters for each dataset and is trained end-to-end to minimize the resulting target-network losses.

4 Theoretical Analysis: A Case Study in Linear Regression

In this section, we theoretically analyze DP-DeepSets and DP-SGD in a synthetic setting. We show that DP-DeepSets effectively encodes datasets in low-dimensional embeddings and can achieve higher utility than DP-SGD under some settings.

We consider linear regression with a random true parameter $w \sim \mathrm{Unif}(\{v \in \mathbb{R}^d \mid \|A_w^{-1} v\|_\infty \le 1\})$, where $A_w := \mathrm{diag}(\alpha_1, \dots, \alpha_d)$ for $\alpha_1, \dots, \alpha_d > 0$. We observe $y_n := w^\top x_n + z_n$ $(n = 1, \dots, N)$, where the inputs $x_n \sim \mathrm{Unif}(\sqrt{d}\,\mathbb{S}^{d-1})$ and noise variables $z_n \sim \mathcal{N}(0, 1)$ are mutually independent and independent of w. Our goal is to estimate w from the dataset $p_{1:N} = \{(x_1, y_1), \dots, (x_N, y_N)\}$. Public and private tasks are drawn i.i.d. from this model.

We also make the following assumption on the distribution of w.

Assumption 5. *It holds $\underline{\alpha} i^{-\gamma} \le \alpha_i \le \overline{\alpha} i^{-\gamma}$ for some constants $\underline{\alpha}, \overline{\alpha} > 0$ and $\gamma > 1/2$.*

This implies that the coordinates with larger indices are progressively less important, in the sense that the allowable magnitude of w_i decays polynomially with i. Therefore, the true parameter w has a small effective dimension, and accordingly the information needed to recover w from the dataset $p_{1:N}$ is also low-dimensional. If an algorithm can successfully exploit this low-dimensional structure, then it can avoid large DP noise and estimate the parameter efficiently.

4.1 Upper Bound of Estimation Error for DP-DeepSets

First, we provide an upper bound on the error in estimating w using DP-DeepSets. For simplicity, we consider DP-DeepSets consisting only of MLPs without Transformers.

To rigorously define the class of DP-DeepSets, we introduce the class of MLPs. Let $\eta := \max\{0, \cdot\}$ be the ReLU activation function. Then, an MLP $f : \mathbb{R}^{d_{\mathrm{in}}} \to \mathbb{R}^{d_{\mathrm{out}}}$ with depth L and width W is defined as $f(x) = (A_L \eta(\cdot) + b_L) \circ \cdots \circ (A_2 \eta(\cdot) + b_2) \circ (A_1 x + b_1)$, where $A_i \in \mathbb{R}^{d_{i+1} \times d_i}$, $b_i \in \mathbb{R}^{d_{i+1}}$ with $d_1 = d_{\mathrm{in}}$, $d_{L+1} = d_{\mathrm{out}}$ and $\max_i d_i \le W$. Let $\Phi_{\mathrm{MLP}}^{d_{\mathrm{in}}, d_{\mathrm{out}}}(L, W, B)$ denote the class of these MLPs satisfying $\max_i \{\|A_i\|_\infty \vee \|b_i\|_\infty\} \le B$. Next, we define the class of DP-DeepSets. DP-DeepSets ϕ (using MLPs instead of a Transformer) takes a dataset $p_{1:N}$ and DP noise ξ as inputs, and predicts the parameter as follows:

$$\phi(p_{1:N}, \xi) := \mathrm{clip}_F \circ \phi_2 \left(\xi + \frac{1}{N} \sum_{n=1}^{N} \mathrm{clip}_C \circ \phi_1(p_n), \right)$$

where $\phi_1 : \mathbb{R}^{d_p} \to \mathbb{R}^D$ and $\phi_2 : \mathbb{R}^D \to \mathbb{R}^{d_\omega}$ are MLPs, and $F > 0$ is a fixed constant. Here, we add a clipping operation compared to the architecture introduced in Section 3 to constrain expressivity. Such clipping operations are commonly used in prior work analyzing the estimation error of neural networks (e.g., Suzuki (2019), Schmidt-Hieber (2020)). Let $\Phi(L, W, B, C, D)$ denote the class of DP-DeepSets defined above with $\phi_1 \in \Phi_{\mathrm{MLP}}^{d_p, D}(L, W, B)$ and $\phi_2 \in \Phi_{\mathrm{MLP}}^{D, d_\omega}(L, W, B)$. The full definitions are given in Appendix B.1.

As described in Section 3.2, we employ public datasets $p_{1:N}^{(1)}, \dots, p_{1:N}^{(M)}$ and DP noise vectors $\xi^{(1)}, \dots, \xi^{(M)}$ to train the hypernetwork. We analyze the hypernetwork $\widehat{\phi}$ chosen to minimize

the loss $\mathcal{L}_{\mathrm{HN}}$ defined in (2) after setting ℓ to the mean squared error, i.e.,

$$\widehat{\phi} := \underset{\phi \in \Phi(L,W,B,C,D)}{\arg\min} \frac{1}{M} \sum_{m=1}^{M} \frac{1}{N} \sum_{n=1}^{N} \left(\phi(p_{1:N}^{(m)}, \xi^{(m)})^\top x_n^{(m)} - y_n^{(m)} \right)^2 \tag{3}$$

Then, the parameters generated by the trained DP-DeepSets $\widehat{\phi}$ for a test (private) dataset satisfy the following theorem. The proof can be found in Appendix B. Write $\nu := \epsilon N / \sqrt{\log(1/\delta)}$.

Theorem 6. *Suppose $d \geq 3$, $N \geq 2d$, $M \geq 1$, $F \geq \|\alpha\|$, $0 < \epsilon \leq 1$, and $0 < \delta \leq \mathrm{e}^{-1}$, with F fixed. For an integer $1 \leq k \leq d$, let $\widehat{\phi}$ be the ERM in* (3)*, with sufficiently large fixed constants in*

$$L \simeq \log[d(\mathrm{e} + \nu^2)], \quad W \simeq d, \quad B \simeq d(1+\nu),$$
$$D = k, \qquad C = A_C \sqrt{k} \log\big(\mathrm{e} + \nu^2\big), \qquad s_{\mathrm{DP}}^2 = A_{\mathrm{H}} C^2 / \nu^2,$$

where A_C is sufficiently large and $A_{\mathrm{H}} \geq 16$ is fixed. Use independent Gaussian noise with covariance $s_{\mathrm{DP}}^2 I_k$ on each task, independently of its data. The resulting private-task release is (ϵ, δ)-DP and satisfies

$$E_{\mathrm{HN}}(k) := \mathbb{E} \left\| \widehat{\phi}(p_{1:N}, \xi) - w \right\|^2 \leq \widetilde{O} \left(\frac{d}{\sqrt{M}} + \frac{k}{N} + \frac{k^2}{\nu^2} + \int_k^d t^{-2\gamma} \mathrm{d}t \right).$$

The expectation is over all task parameters, data, and DP noise.

The four terms describe generalization across M public tasks, estimation from N samples in a k-dimensional representation, privacy noise, and approximation error, respectively. Public-task learning reduces the dimension in both the statistical and privacy terms to k, while the decoder still generates d-dimensional parameters. Increasing k reduces approximation error at the cost of larger statistical and privacy terms.

4.2 Lower Bound for DP-SGD

Next, we present the lower bound on the error in estimating the parameter w using DP-SGD for the target model $\psi_\omega(x) = \omega^\top x$. We run DP-SGD on the following empirical loss for linear regression:

$$\mathcal{L}(\omega) := \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}_n(\omega), \quad \text{where} \quad \mathcal{L}_n(\omega) := \frac{1}{2} (\omega^\top x_n - y_n)^2.$$

We analyze the full-batch DP-SGD update in (1), with $b = N$, $g_n(\omega) = \nabla \mathcal{L}_n(\omega)$, and a constant learning rate η. After T steps with noise vectors $\xi_1, \ldots, \xi_T$, the resulting parameter ω_T satisfies the following theorem. The proof is provided in Appendix C.

Theorem 7. *Suppose $d \geq 3$, $N \geq 2d$, $0 < \epsilon \leq 1$, and $0 < \delta \leq \mathrm{e}^{-1}$. Run the full-batch update above from $\omega_0 = 0$. For every $C > 0$, $\eta \in (0, 1)$, and $T \in \mathbb{Z}_{>0}$, fixed before observing the private task, use independent noise $\xi_t \sim \mathcal{N}(0, s^2 I_d)$ with $s^2 \simeq C^2 T \log(1/\delta)/\epsilon^2$, using a sufficiently large fixed proportionality constant. This algorithm is (ϵ, δ)-DP, and*

$$E_{\mathrm{SGD}}^\star := \inf_{\substack{C>0,\ \eta \in (0,1) \\ T \in \mathbb{Z}_{>0}}} \mathbb{E} \|\omega_T - w\|^2 \gtrsim \frac{d}{N} + \min \left\{ 1, \frac{d^2}{\nu^2} \right\}.$$

The expectation is over w, $p_{1:N}$, and the DP noise.

The two terms are statistical and privacy penalties for this DP-SGD algorithm. The bound holds uniformly over the clipping threshold, learning rate, and number of updates. Thus tuning these parameters, including running more updates under the same privacy budget, cannot remove the privacy penalty.

Comparison under privacy constraints With sufficiently many public tasks, HN offers two advantages over the full-batch DP-SGD of Theorem 7 under the same privacy budget. *First, HN can*

achieve vanishing error where DP-SGD retains a positive error floor. For $1 \le \nu \le d$, Theorem 7 gives $E^{\star}_{\mathrm{SGD}} \gtrsim 1$. For HN, balancing the upper-bound terms gives, up to integer rounding,

$$k_* \simeq \min\{d, N^{1/(2\gamma)}, \nu^{2/(2\gamma+1)}\},$$
$$E_{\mathrm{HN}}(k_*) \lesssim \Lambda \left(d/\sqrt{M} + N^{-(2\gamma-1)/(2\gamma)} + \nu^{-2(2\gamma-1)/(2\gamma+1)} \right),$$

where Λ is the logarithmic factor in Appendix B.5. Along sequences satisfying the theorem assumptions and $1 \le \nu \le d$, if this upper bound tends to zero, HN achieves vanishing error while the DP-SGD lower bound stays bounded away from zero (Corollary 13).

Second, HN can retain its statistical rate while DP-SGD is limited by privacy noise. Take $k \simeq N^{1/(2\gamma)} < d$ and suppose

$$M \gtrsim (dN/k)^2, \qquad kN \lesssim \nu^2 \ll dN,$$

with $N/d \to \infty$. In this privacy window, the HN privacy term is at most its statistical term up to logarithmic factors, and the public-task and approximation terms are also controlled, giving $E_{\mathrm{HN}}(k) \lesssim \Lambda k/N$. For DP-SGD, the privacy term instead dominates its statistical term d/N. In the non-saturated part of this window, $\nu \ge d$, the error ratio satisfies $E_{\mathrm{HN}}(k)/E^{\star}_{\mathrm{SGD}} \lesssim \Lambda k\nu^2/(Nd^2)$. Thus HN has asymptotically smaller error whenever this ratio bound tends to zero. See Appendix B.6 for integer rounding and matching rates at $k = d$ with sufficiently large M and $\nu \ge d$.

5 Experiments: LoRA Finetuning of Diffusion Models

In this section, we demonstrate the effectiveness of our approach in LoRA finetuning of diffusion models.

Models and Datasets As the pretrained model, we use the checkpoint for the unconditional diffusion model trained on ImageNet64 (`imagenet64_uncond_100M_1500K.pt`), provided in the codebase[1] of Nichol & Dhariwal (2021). We consider *finetuning on small private datasets* constructed from CIFAR-10 (Krizhevsky, 2009). Specifically, we first divide the 60,000 data points (50,000 training and 10,000 test images) into two subsets, treating one subset of 30,000 data points as public data and the remaining half as private data. Next, we embed the data into 2,048-dimensional representations using Inception (Szegedy et al., 2016). We then select 100 representative data points from the private chunk and construct a *set of 100 private datasets, each containing 128 data points*, by taking the k-nearest neighbors of each representative point with $k = 128$. Here, the representative points are selected to minimize overlap between the resulting datasets.

Settings for DP-DeepSets To feed image datasets into DP-DeepSets, we use CLIP (Radford et al., 2021) to embed images into 512-dimensional representations. CLIP is frozen and applied independently to each image. The base diffusion model is kept fixed, and DP-DeepSets generates only the LoRA parameters. The LoRA parameters are partitioned into 318 tokens, so that each token contains only components of the LoRA parameters corresponding to the same location.

When training DP-DeepSets, we employ the public chunk of CIFAR-10 containing 30,000 data points. More concretely, we take the k-nearest neighbors of each data point in the Inception embedding space with $k = 128$ to obtain 30,000 public datasets with 128 data points. Then, we fit the parameters θ of DP-DeepSets using the loss function $\mathcal{L}_{\mathrm{HN}}$ defined in (2) with $N = 128$ and $M = 30{,}000$, where $p^{(m)}_{1:N}$ denotes the CLIP embeddings of the k-NN set for the m-th data point in the public chunk. We train a separate hypernetwork for each target ϵ, using its corresponding noise scale during both public training and private inference.

Comparison baselines As the most basic baseline, we apply DP-SGD to each private dataset separately. We also evaluate DP-AdamW (Chooi et al., 2025) and Projected DP-SGD (Zhou et al., 2021); the latter clips and perturbs gradients in a fixed rank-32 subspace constructed from public gradients. In addition, we consider two types of public-data-guided baselines. First, we consider finetuning the model on public data and then performing DP finetuning on private data. Specifically, we first finetune the model with standard Adam on the public chunk consisting of 30,000 examples, and then fit the model to the private data using DP-SGD. Second, we compare our method with PDA-DPMD (Amid et al., 2022). This method performs mirror descent using the loss on public data

[1] `https://github.com/openai/improved-diffusion`

as the mirror map. In our experiments, we use the noise prediction loss for denoising on public data as the mirror map. The reported baseline configurations are fixed using held-out public tasks before evaluation on the 100 private tasks; Appendix E gives the selection procedure and candidate grids.

Results For each method, we conduct finetuning on 100 datasets, which yields 100 LoRA finetuned models. Then, we generate 1,024 images from each finetuned model, and compute the FID between the generated images and the private dataset corresponding to the model. For the DP cost, we fix $\delta = 10^{-5}$, and conduct experiments with various values of ϵ. We report the average and the standard deviation of FID scores over 100 datasets.

The results are shown in Figure 2. We also display the generated samples from the model finetuned by our method with $\epsilon = 2$ in Figure 1. We describe some observations below:

- **Superiority of DP-DeepSets over DP-SGD:** We observe that the FID scores of our method are much lower than those of DP-SGD. In particular, as ϵ decreases, DP-SGD exhibits a substantial increase in FID for $\epsilon \leq 16$, whereas the proposed method maintains nearly the same performance down to around $\epsilon = 1$. This can be explained in part by adding noise once to a low-dimensional dataset representation, which mitigates its impact compared with DP-SGD's repeated parameter-dimensional noise. This effect is particularly pronounced in our setting, where the private dataset is small, containing only 128 samples.
- **Stronger results than other public-data-guided approaches:** We observe that the proposed method outperforms baselines that use public data (Projected DP-SGD, Public FT + DP-SGD, and PDA-DPMD). These results suggest that the benefit comes not simply from using public data, but from learning a mapping from datasets to model parameters through hypernetwork training. By learning across public tasks, the hypernetwork can exploit shared task structure to adapt to small private datasets.

Generalization under distribution shift To evaluate generalization across public and private data distributions, we train DP-DeepSets on public datasets constructed from ImageNet64 and evaluate it on the same CIFAR-10 private datasets. We use $\epsilon \in \{1, 4\}$ and $\delta = 10^{-5}$; training and architecture details are provided in Appendix E. As shown in Table 1, DP-DeepSets achieves substantially lower FID than both DP-AdamW and DP-SGD at both privacy budgets. These results show that DP-DeepSets can learn useful parameter generation from public data *even when its distribution differs from that of the private data*.

Table 1: FID under distribution shift: DP-DeepSets is trained on ImageNet64 public datasets and evaluated on CIFAR-10 private datasets.

Method	$\epsilon = 1$	$\epsilon = 4$
Ours	**157.4 ± 26.5**	**156.3 ± 26.1**
DP-AdamW	251.1 ± 19.8	245.8 ± 20.3
DP-SGD	426.7 ± 41.1	338.8 ± 48.1

6 Conclusion

In this paper, we propose a new framework for DP learning using a public hypernetwork. Specifically, we develop DP-DeepSets, a hypernetwork architecture for differentially private learning. Since it performs DP-noise perturbation on a low-dimensional representation of private datasets only once, it suppresses the overall magnitude of DP noise, thereby achieving higher utility than gradient-based methods. We also propose an end-to-end pipeline to train DP-DeepSets using public datasets. In a theoretical analysis in a synthetic setting, we show that the proposed method performs better because it learns how to encode low-dimensional structure into dataset embeddings during training with public datasets. Moreover, we conduct experiments on LoRA finetuning of diffusion models and demonstrate that our approach achieves higher utility than DP-SGD and other public-data-guided methods.

Limitations and Future Work While our proposed framework is broadly applicable to various modalities and tasks, in this paper, we focused on LoRA finetuning of diffusion models, particularly in the context of data generation. Extending our experiments to other modalities, including language, as well as to tasks beyond data generation, remains an important direction for future work. In addition, our experiments considered a small-data regime, where gradient-based DP learning is often severely hindered by the limited dataset size. A promising future direction is to investigate how the effectiveness of our method scales with dataset size, and to clarify up to what regime it can serve as a practical alternative to gradient-based approaches.

AI USE STATEMENT

In this work, we used generative AI tools to assist with implementing the experimental methods, revising mathematical statements, developing and checking proof arguments, and drafting and editing proofs. We also used these tools to obtain feedback on the research methodology and experimental design, interpret experimental results, survey related literature, and improve the clarity and readability of the manuscript. The authors reviewed the AI-assisted material, including the mathematical arguments, cited literature, and experimental code, and made the final decisions about the content included in the paper. The authors take full responsibility for the final manuscript, its theoretical claims, and the accompanying code and experimental results.

REFERENCES

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 308–318. ACM, 2016.

Ehsan Amid, Arun Ganesh, Rajiv Mathews, Swaroop Ramaswamy, Shuang Song, Thomas Steinke, Vinith M Suriyakumar, Om Thakkar, and Abhradeep Thakurta. Public data-assisted mirror descent for private model training. In *International Conference on Machine Learning*, pp. 517–535. PMLR, 2022.

Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *32nd USENIX security symposium (USENIX Security 23)*, pp. 5253–5270, 2023.

Jay Chooi, Kevin Cong, Russell Li, and Lillian Sun. DP-AdamW: Investigating decoupled weight decay and bias correction in private deep learning. In *ICML 2025 Workshop on Collaborative and Federated Agentic Workflows*, 2025. URL https://openreview.net/forum?id=guMa1yMr1t.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and trends® in theoretical computer science*, 9(3-4):211–487, 2014.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pp. 265–284. Springer, 2006.

David Ha, Andrew M Dai, and Quoc V Le. Hypernetworks. In *International Conference on Learning Representations*, 2017.

Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53 (2):217–288, 2011. URL https://arxiv.org/abs/0909.4061v2. arXiv:0909.4061v2.

Frederik Harder, Kamil Adamczewski, and Mijung Park. 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. URL https://proceedings.mlr.press/v130/harder21a.html.

Chia-Yi Hsu, Jia-You Chen, Yu-Lin Tsai, Chih-Hsun Lin, Pin-Yu Chen, Chia-Mu Yu, and Chun-Ying Huang. VP-NTK: Exploring the benefits of visual prompting in differentially private data synthesis. In *2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025. URL https://arxiv.org/abs/2503.16195.

Xiaolong Jin, Kai Wang, Dongwen Tang, Wangbo Zhao, Yukun Zhou, Junshu Tang, and Yang You. Conditional lora parameter generation. *arXiv preprint arXiv:2408.01415*, 2024.

Rana Muhammad Shahroz Khan, Dongwen Tang, Pingzhi Li, Kai Wang, and Tianlong Chen. ORAL: Prompting your large-scale LoRAs via conditional recurrent diffusion. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 1357–1370, 2025.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Tian Li, Manzil Zaheer, Sashank Reddi, and Virginia Smith. Private adaptive optimization with side information. In *International Conference on Machine Learning*, pp. 13086–13105. PMLR, 2022.

Zhiyuan Liang, Dongwen Tang, Yuhao Zhou, Xuanlei Zhao, Mingjia Shi, Wangbo Zhao, Zekai Li, Peihao Wang, Konstantin Schürholt, Damian Borth, et al. Drag-and-drop llms: Zero-shot prompt-to-weights. *Advances in Neural Information Processing Systems*, 38, 2025.

Ilya Mironov. Rényi differential privacy. arXiv preprint arXiv:1702.07476, 2017. URL `https://arxiv.org/abs/1702.07476`.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PMLR, 2021.

Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *International Conference on Learning Representations*, 2017.

Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Ulfar Erlingsson. Scalable private learning with PATE. In *International Conference on Learning Representations*, 2018.

William Peebles, Ilija Radosavovic, Tim Brooks, Alexei A Efros, and Jitendra Malik. Learning to learn with generative models of neural network checkpoints. *arXiv preprint arXiv:2209.12892*, 2022.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Ossi Räisä, Stratis Markou, Matthew Ashman, Wessel Bruinsma, Marlon Tobaben, Antti Honkela, and Richard Turner. Noise-aware differentially private regression via meta-learning. *Advances in Neural Information Processing Systems*, 37:63873–63907, 2024.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Wei Wei, Tingbo Hou, Yael Pritch, Neal Wadhwa, Michael Rubinstein, and Kfir Aberman. Hyperdreambooth: Hypernetworks for fast personalization of text-to-image models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6527–6536, 2024.

Anselm Johannes Schmidt-Hieber. Nonparametric regression using deep neural networks with relu activation function. *Annals of statistics*, 48(4):1875–1897, 2020.

Bedionita Soro, Bruno Andreis, Hayeon Lee, Wonyong Jeong, Song Chong, Frank Hutter, and Sung Ju Hwang. Diffusion-based neural network weights generation. In *The Thirteenth International Conference on Learning Representations*, 2025.

Taiji Suzuki. Adaptivity of deep relu network for learning in besov and mixed smooth besov spaces: optimal rate and curse of dimensionality. In *International Conference on Learning Representations*, 2019.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Florian Tramer and Dan Boneh. Differentially private learning needs better features (or much more data). In *International Conference on Learning Representations*, 2021.

Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, 2018.

Martin J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge University Press, 2019.

Kai Wang, Dongwen Tang, Boya Zeng, Yida Yin, Zhaopan Xu, Yukun Zhou, Zelin Zang, Trevor Darrell, Zhuang Liu, and Yang You. Neural network diffusion. *arXiv preprint arXiv:2402.13144*, 2024.

Kai Wang, Dongwen Tang, Wangbo Zhao, Konstantin Schürholt, Zhangyang Wang, and Yang You. Scaling up parameter generation: A recurrent diffusion approach. *Advances in Neural Information Processing Systems*, 38, 2025.

Yixin Wu, Ning Yu, Zheng Li, Michael Backes, and Yang Zhang. Membership inference attacks against text-to-image generation models. *arXiv preprint arXiv:2210.00968*, 2022.

Da Yu, Huishuai Zhang, Wei Chen, and Tie-Yan Liu. Do not let privacy overbill utility: Gradient embedding perturbation for private learning. In *International Conference on Learning Representations*, 2021.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

Lele Zheng, Xiang Wang, Tao Zhang, Yang Cao, Ke Cheng, and Yulong Shen. Differentially private subspace fine-tuning for large language models. *arXiv preprint arXiv:2601.11113*, 2026.

Yingxue Zhou, Xiangyi Chen, Mingyi Hong, Zhiwei Steven Wu, and Arindam Banerjee. Private stochastic non-convex optimization: Adaptive algorithms and tighter generalization bounds. *arXiv preprint arXiv:2006.13501*, 2020.

Yingxue Zhou, Steven Wu, and Arindam Banerjee. Bypassing the ambient dimension: Private sgd with gradient subspace identification. In *International Conference on Learning Representations*, 2021.

Junyi Zhu and Matthew Blaschko. Improving differentially private sgd via randomly sparsified gradients. *Transactions on Machine Learning Research*, 2023.

A Proof of Corollary 3

Here, we prove Corollary 3.

Proof. Let $p_{1:N}$ and $p'_{1:N}$ be adjacent datasets. Without loss of generality, we assume that $p_n = p'_n$ for $n = 2, \dots, N$. Then, we have

$$\|E(p_{1:N}, \xi) - E(p'_{1:N}, \xi)\| \le \frac{1}{N} \|\mathrm{clip}_C(\phi_1(p_1)) - \mathrm{clip}_C(\phi_1(p'_1))\| \le \frac{2C}{N}.$$

Proposition 1 implies that the map E is (ϵ, δ)-DP. Since the network ϕ_2 does not depend on $p_{1:N}$, the post-processing property of DP yields the claim. □

B Proof of Theorem 6

B.1 Definitions of model classes

We give the full definitions of the model classes used in Section 4. Let $\eta := \max\{0, \cdot\}$ be the ReLU activation function. An MLP $f : \mathbb{R}^{d_{\mathrm{in}}} \to \mathbb{R}^{d_{\mathrm{out}}}$ with depth L and width at most W has the form

$$f(x) = (A_L \eta(\cdot) + b_L) \circ \cdots \circ (A_2 \eta(\cdot) + b_2) \circ (A_1 x + b_1),$$

where $A_i \in \mathbb{R}^{d_{i+1} \times d_i}$ and $b_i \in \mathbb{R}^{d_{i+1}}$, with $d_1 = d_{\mathrm{in}}$, $d_{L+1} = d_{\mathrm{out}}$, and $\max_i d_i \le W$. The class of such MLPs with norm constraint B is

$$\Phi_{\mathrm{MLP}}^{d_{\mathrm{in}}, d_{\mathrm{out}}}(L, W, B) := \left\{ f \;\middle|\; \max_i \{\|A_i\|_\infty \vee \|b_i\|_\infty\} \le B \right\}.$$

For a fixed output clipping threshold $F > 0$, the DP-DeepSets map is

$$\phi(p_{1:N}, \xi) := \mathrm{clip}_F \circ \phi_2 \left(\xi + \frac{1}{N} \sum_{n=1}^{N} \mathrm{clip}_C(\phi_1(p_n)) \right),$$

where $\phi_1 : \mathbb{R}^{d_p} \to \mathbb{R}^D$, $\phi_2 : \mathbb{R}^D \to \mathbb{R}^{d_\omega}$, and $\xi \in \mathbb{R}^D$. The corresponding model class is

$$\Phi(L, W, B, C, D) := \left\{ \phi \;\middle|\; \phi_1 \in \Phi_{\mathrm{MLP}}^{d_p, D}(L, W, B), \quad \phi_2 \in \Phi_{\mathrm{MLP}}^{D, d_\omega}(L, W, B) \right\}.$$

Here, C is the embedding clipping threshold and D is the embedding dimension; the fixed constant F is omitted from the notation.

We define the empirical risk as

$$\widehat{R}(\phi) := \frac{1}{M} \sum_{m=1}^{M} \left[\frac{1}{N} \sum_{n=1}^{N} \left\{ \phi(p_{1:N}^{(m)}, \xi^{(m)})^\top x_n^{(m)} - y_n^{(m)} \right\}^2 \right].$$

B.2 Upper Bound of the Empirical Error

We first prove the following lemma, which upper bounds the empirical parameter estimation error by the expected error.

Lemma 8. *Suppose $d \ge 3$, $N \ge 2d$, $M \ge 1$, $D \ge 1$, and $F \ge \|\alpha\|$, with F fixed. DP noise vectors are independent $\mathcal{N}(0, s_{\mathrm{DP}}^2 I_D)$ and independent of the tasks. Let $\widehat{\phi}$ minimize $\widehat{R}$ over $\Phi(L, W, B, C, D)$, and fix $\phi^* \in \Phi(L, W, B, C, D)$ independently of the public tasks. Set $\Delta_N := \mathbb{E}\|\phi^*(p_{1:N}, \xi) - w\|^2$, and define*

$$J = [(W+1)(B \vee 1)]^L, \qquad J_\theta = \sqrt{d}\, L (B \vee 1)^{L-1} (W+1)^L,$$

$$\mathcal{A}_N = 1 + \log\left(1 + 16 B J_\theta N (1 + C + \sqrt{D}\, s_{\mathrm{DP}})\right) + \log(1 + 16 C J N),$$

$$\tau = \min\{1, (4K_{N,d}^2)^{-1}\},$$

where $K_{N,d}$ is from Lemma 14. Then, with a constant depending only on $F, \overline{\alpha}, \gamma$,

$$\mathbb{E}\left[\frac{1}{M} \sum_{m=1}^{M} \left\| \widehat{\phi}(p_{1:N}^{(m)}, \xi^{(m)}) - \phi^*(p_{1:N}^{(m)}, \xi^{(m)}) \right\|^2 \right] \lesssim \tau^{-1} \left[\Delta_N + \left(\frac{D}{N} + \frac{LW^2}{MN} \right) \mathcal{A}_N \right].$$

Proof. First, it holds $\widehat{R}(\widehat{\phi}) \le \widehat{R}(\phi^*)$, which implies

$$\frac{1}{M}\sum_{m=1}^{M}\left[\frac{1}{N}\sum_{n=1}^{N}\left\{\widehat{\phi}(p_{1:N}^{(m)},\xi^{(m)})^\top x_n^{(m)} - y_n^{(m)}\right\}^2\right] \le \frac{1}{M}\sum_{m=1}^{M}\left[\frac{1}{N}\sum_{n=1}^{N}\left\{\phi^*(p_{1:N}^{(m)},\xi^{(m)})^\top x_n^{(m)} - y_n^{(m)}\right\}^2\right]$$

Let $h^{(m)} := \widehat{\phi}(p_{1:N}^{(m)},\xi^{(m)}) - \phi^*(p_{1:N}^{(m)},\xi^{(m)})$ and $\Delta^{(m)} := w^{(m)} - \phi^*(p_{1:N}^{(m)},\xi^{(m)})$. Since it holds

$$y_n^{(m)} = (w^{(m)})^\top x_n^{(m)} + z_n^{(m)} = (\phi^*(p_{1:N}^{(m)},\xi^{(m)}) + \Delta^{(m)})^\top x_n^{(m)} + z_n^{(m)},$$

we have

$$\frac{1}{M}\sum_{m=1}^{M}\left[\frac{1}{N}\sum_{n=1}^{N}\left\{\left(h^{(m)}\right)^\top x_n^{(m)} - \left(\Delta^{(m)}\right)^\top x_n^{(m)} - z_n^{(m)}\right\}^2\right] \le \frac{1}{M}\sum_{m=1}^{M}\left[\frac{1}{N}\sum_{n=1}^{N}\left[(\Delta^{(m)})^\top x_n^{(m)} + z_n^{(m)}\right]^2\right].$$

We have

$$\frac{1}{M}\sum_{m=1}^{M}\left[(h^{(m)})^\top\Sigma^{(m)}h^{(m)} - 2(h^{(m)})^\top\Sigma^{(m)}\Delta^{(m)} - 2(g^{(m)})^\top h^{(m)}\right] \le 0,$$

where $\Sigma^{(m)} := \frac{1}{N}\sum_{n=1}^{N} x_n^{(m)}(x_n^{(m)})^\top$ and $g^{(m)} := \frac{1}{N}\sum_{n=1}^{N} z_n^{(m)}x_n^{(m)}$. This implies that

$$\frac{1}{M}\sum_{m=1}^{M}\left\{(h^{(m)})^\top\Sigma^{(m)}h^{(m)}\right\} \le \frac{2}{M}\sum_{m=1}^{M}\left\{(h^{(m)})^\top\left(\Sigma^{(m)}\Delta^{(m)} + g^{(m)}\right)\right\}.$$

We control the noise inner product uniformly over possible output sequences. Write $\|v\|_M^2 = M^{-1}\sum_m \|v_m\|^2$ and

$$Q(v) = \frac{1}{M}\sum_m v_m^\top\Sigma^{(m)}v_m, \qquad G(v) = \frac{1}{M}\sum_m (g^{(m)})^\top v_m.$$

For any encoder, the average clipped embedding lies in $B_D(0,C)$. Consequently every public-task output sequence lies in the larger set

$$\mathcal{V}_{\boldsymbol{\xi}} = \left\{\left(\mathrm{clip}_F f(\xi^{(m)} + b_m)\right)_{m=1}^{M} : f \in \Phi_{\mathrm{MLP}}^{D,d}(L,W,B),\ b_m \in B_D(0,C)\right\}.$$

This is an enlargement used only for analysis; all tasks share one decoder. After padding its hidden layers, the decoder has at most $P_2 = L(W^2+W) \le 2LW^2$ parameters. Its Euclidean input-Lipschitz constant is at most J, and a parameter perturbation of size q in ℓ_∞ changes its output at input v by at most $J_\theta(1+\|v\|)q$. These follow by propagating input and parameter perturbations through the L affine layers, as in the proof of Lemma 9; Euclidean output clipping is 1-Lipschitz. Let

$$R_{\boldsymbol{\xi}} := \left[\frac{1}{M}\sum_m (1 + C + \|\xi^{(m)}\|)^2\right]^{1/2}.$$

Discretize the decoder parameters at precision $r/(4J_\theta R_{\boldsymbol{\xi}})$ and each b_m at Euclidean precision $r/(4J)$, with centers inside their respective parameter cube and balls. The mean squared output distance is at most r^2, and the resulting cover obeys

$$\log\mathcal{N}(r,\mathcal{V}_{\boldsymbol{\xi}},\|\cdot\|_M) \le P_2\log\left(1 + \frac{8BJ_\theta R_{\boldsymbol{\xi}}}{r}\right) + MD\log\left(1 + \frac{8CJ}{r}\right). \tag{4}$$

Taking differences of two $r/2$-nets gives an r-net of $\mathcal{H}_{\boldsymbol{\xi}} := \mathcal{V}_{\boldsymbol{\xi}} - \mathcal{V}_{\boldsymbol{\xi}}$. Its centers satisfy $\|u_m\| \le 2F$, and $h = (h^{(m)})_m$ belongs to this set. The net depends on $\boldsymbol{\xi}$, but not on observation noise z. Moreover,

$\mathbb{E}R_{\boldsymbol{\xi}} \le 1 + C + \sqrt{D}\, s_{\mathrm{DP}}$ by Minkowski's and Jensen's inequalities. Thus expected log covering numbers have the same bound with this quantity in place of $R_{\boldsymbol{\xi}}$, without a $\log M$ factor.

Condition on all $x_n^{(m)}, w^{(m)}, \boldsymbol{\xi}$. For any fixed net point u, independence of the observation noises gives

$$G(u) \mid x, w, \boldsymbol{\xi} \sim \mathcal{N}\left(0, \frac{Q(u)}{MN}\right), \qquad \mathbb{E}_z \exp\left[\frac{MN}{8}\left(\pm 2G(u) - \frac{1}{4}Q(u)\right)\right] = 1.$$

For a net of cardinality K, a union bound and integration of $\min\{1, 2K\exp(-MNt/8)\}$ over $t \ge 0$ yield

$$\mathbb{E}_z \max_{u \text{ in the net}} \left(|2G(u)| - \tfrac{1}{4}Q(u)\right)_+ \le \frac{8\{\log(2K) + 1\}}{MN}.$$

Choose $r = 1/N$. For $u, v \in \mathcal{H}_{\boldsymbol{\xi}}$ with $\|u - v\|_M \le r$, Cauchy–Schwarz gives

$$|G(u) - G(v)| \le r\|g\|_M,$$
$$|Q(u) - Q(v)| \le 4Fr\left(\frac{1}{M}\sum_m \|\Sigma^{(m)}\|_{\mathrm{op}}^2\right)^{1/2}.$$

Here $\mathbb{E}\|g\|_M \le \sqrt{d/N}$, and integrating the upper tail in Lemma 14 gives $\mathbb{E}\|\Sigma^{(1)}\|_{\mathrm{op}}^2 \lesssim 1$ for $N \ge 2d$. The expected discretization error is therefore $O(1/N)$. Applying (4) to the difference net, then Jensen's inequality to its log cardinality, proves

$$H := \sup_{u \in \mathcal{H}_{\boldsymbol{\xi}}} \left(|2G(u)| - \tfrac{1}{4}Q(u)\right)_+, \qquad \mathbb{E}H \lesssim \left(\frac{D}{N} + \frac{P_2}{MN}\right)\mathcal{A}_N. \tag{5}$$

The learned output and the oracle output evaluated on public data may both depend on z; the uniform bound above applies to their difference.

Return to the basic inequality. Young's inequality gives

$$\frac{2}{M}\sum_m (h^{(m)})^\top \Sigma^{(m)}\Delta^{(m)} \le \tfrac{1}{4}Q(h) + 4Q(\Delta), \qquad 2G(h) \le \tfrac{1}{4}Q(h) + H,$$

and hence $Q(h) \le 8Q(\Delta) + 2H$. To estimate $Q(\Delta)$, take a sufficiently large absolute K_0. Since $\|\Delta^{(m)}\| \le 2F$ and $\|\Sigma^{(m)}\|_{\mathrm{op}} \le d$, the upper tail in Lemma 14 yields

$$\mathbb{E}Q(\Delta) \le K_0\Delta_N + 4F^2 d\,\mathbb{P}(\|\Sigma^{(1)}\|_{\mathrm{op}} > K_0) \lesssim \Delta_N + d\mathrm{e}^{-cN}.$$

This argument allows dependence between $\Delta^{(m)}$ and $\Sigma^{(m)}$. Finally, set $p_N := \mathbb{P}(\lambda_{\min}(\Sigma^{(1)}) < \tau)$. The lower tail in Lemma 14 implies

$$p_N \le 2^{-(N-d)}, \qquad \tau^{-1} \lesssim \left(1 + \sqrt{\frac{\log(2N)}{d}}\right)^2.$$

Split the task indices according to whether their minimum eigenvalue is at least τ. The quadratic form controls the good tasks; on every other task, $\|h^{(m)}\| \le 2F$. Taking averages, rather than requiring a simultaneous event, gives

$$\mathbb{E}\frac{1}{M}\sum_m \|h^{(m)}\|^2 \le \tau^{-1}\mathbb{E}Q(h) + 4F^2 p_N$$
$$\lesssim \tau^{-1}\left[\Delta_N + d\mathrm{e}^{-cN} + \left(\frac{D}{N} + \frac{P_2}{MN}\right)\mathcal{A}_N\right] + 4F^2 p_N.$$

Both $d\mathrm{e}^{-cN}$ and p_N are $O(1/N)$ for $N \ge 2d$. They are absorbed into $D\mathcal{A}_N/N$, proving the claim.

□

B.3 Difference Between the Empirical Error and the Test Error

Here, we prove the following lemma, which upper-bounds the difference between the empirical error and the test error between two MLPs. Eventually, we will use this lemma by setting ϕ as the empirical minimizer, and ϕ^* as the network constructed to achieve small error.

Lemma 9. *For any $\phi, \phi^* \in \Phi(L, W, B, C, D)$, it holds*

$$\mathbb{E}\left[\sup_{\phi\in\Phi}\left|\mathbb{E}[f_\phi(p_{1:N},\xi)] - \frac{1}{M}\sum_{m=1}^{M} f_\phi(p_{1:N}^{(m)},\xi^{(m)})\right|\right] \leq \frac{A_9 LW}{\sqrt{M}}\sqrt{\log\{dDL(B\vee 1)(W+1)(C\vee 1)\cdot(1+s_{\mathrm{DP}}^2)\}}.$$

where A_9 is a constant depending only on F, $\overline{\alpha}$, and γ.

Proof. For $\phi \in \Phi(L, W, B, C, D)$, let $\mathrm{vec}(\phi) \in [-B, B]^{\mathcal{D}}$ denote the vector obtained by concatenating its parameters into a single vector. The input and output dimensions are $d_p = d + 1$ and $d_\omega = d$; by the model-class definition, $W \geq \max\{d_p, d_\omega, D\}$.

For a fixed $\phi^* \in \Phi(L, W, B, C, D)$, we define the function class $\mathcal{F}(L, W, B, C, D)$ as

$$\mathcal{F}(L, W, B, C, D) := \left\{ f_\phi : (p_{1:N}, \xi) \mapsto \|\phi(p_{1:N}, \xi) - \phi^*(p_{1:N}, \xi)\|^2 \;\middle|\; \phi \in \Phi(L, W, B, C, D) \right\}.$$

We first remark that the number of parameters $\mathcal{D}(L, W, D)$ of neural networks in $\Phi(L, W, B, C, D)$ is bounded as

$$\mathcal{D}(L, W, D) \leq 2L(W^2 + W) \leq 4LW^2.$$

Each of the two MLPs has L affine layers, with at most W^2 weights and W biases per layer. Hidden layers can be padded to width W when forming the parameter vector. There exists a δ-cover $\mathcal{C}$ of $\{\mathrm{vec}(\phi) \mid \phi \in \Phi(L, W, B, C, D)\} = [-B, B]^{\mathcal{D}}$ with $|\mathcal{C}| \leq (1 + 2B/\delta)^{\mathcal{D}}$ with respect to the $\|\cdot\|_\infty$ norm. Therefore, for any $\phi \in \Phi(L, W, B, C, D)$, we can choose $\mathrm{vec}(\phi') \in \mathcal{C}$ such that $\|\mathrm{vec}(\phi) - \mathrm{vec}(\phi')\|_\infty \leq \delta$. Then, for any $x_{1:N} \in \mathbb{R}^{d\times N}$, $y_{1:N} \in \mathbb{R}^N$ and $\xi \in \mathbb{R}^D$, we have

$$\|\phi_1(p_n) - \phi_1'(p_n)\|_\infty \leq L(B \vee 1)^{L-1}(W+1)^L \cdot (1 \vee \|p_n\|_\infty)\delta.$$

The proof of this inequality can be found in Suzuki (2019), for example. Then, we have

$$\begin{aligned}
&\left\|\left(\frac{1}{N}\sum_{n=1}^{N}\mathrm{clip}_C(\phi_1(p_n)) + \xi\right) - \left(\frac{1}{N}\sum_{n=1}^{N}\mathrm{clip}_C(\phi_1'(p_n)) + \xi\right)\right\|_\infty \\
&\qquad \leq \frac{1}{N}\sum_{n=1}^{N}\|\mathrm{clip}_C(\phi_1(p_n)) - \mathrm{clip}_C(\phi_1'(p_n))\|_\infty \\
&\qquad \leq \frac{1}{N}\sum_{n=1}^{N}\|\mathrm{clip}_C(\phi_1(p_n)) - \mathrm{clip}_C(\phi_1'(p_n))\|_2 \\
&\qquad \leq \frac{1}{N}\sum_{n=1}^{N}\|\phi_1(p_n) - \phi_1'(p_n)\|_2 \\
&\qquad \leq \sqrt{D}\,L(B\vee 1)^{L-1}(W+1)^L \cdot \left(\frac{1}{N}\sum_{n=1}^{N}(1\vee\|p_n\|_\infty)\right)\delta.
\end{aligned}$$

On the data distribution, $\|x_n\|_\infty \geq 1$. We can bound the term $\frac{1}{N}\sum_{n=1}^N (1 \vee \|p_n\|_\infty)$ as follows:

$$\begin{aligned}\frac{1}{N}\sum_{n=1}^N (1 \vee \|p_n\|_\infty) &\leq \frac{1}{N}\sum_{n=1}^N (\|x_n\|_2 + |y_n|) \leq \frac{1}{N}\sum_{n=1}^N (\|x_n\|_2 + \|x_n\|_2\|w\|_2 + |z_n|) \\ &\leq (1+\|\alpha\|)\frac{1}{N}\sum_{n=1}^N \|x_n\|_2 + \frac{1}{N}\sum_{n=1}^N |z_n| \\ &\leq d(1+\|\alpha\|) + \frac{1}{N}\sum_{n=1}^N |z_n| \\ &\leq d \cdot \mathfrak{C}_1(z_{1:N}),\end{aligned}$$

where we define $\mathfrak{C}_1(z_{1:N}) := 1 + \|\alpha\| + \frac{1}{N}\sum_{n=1}^N |z_n|$. Therefore, we have

$$\begin{aligned}&\left\| \left(\frac{1}{N}\sum_{n=1}^N \mathrm{clip}_C(\phi_1(p_n)) + \xi \right) - \left(\frac{1}{N}\sum_{n=1}^N \mathrm{clip}_C(\phi_1'(p_n)) + \xi \right) \right\|_\infty \\ &\qquad \leq \sqrt{D}\, dL(B \vee 1)^{L-1}(W+1)^L \cdot \mathfrak{C}_1(z_{1:N})\delta.\end{aligned}$$

Since clip_F is 1-Lipschitz in Euclidean norm, ϕ_2 is $(BW)^L$-Lipschitz, and we have

$$\begin{aligned}&\|\phi(p_{1:N},\xi) - \phi'(p_{1:N},\xi)\|_\infty \\ &\leq \sqrt{d}\left\| \phi_2\left(\frac{1}{N}\sum_{n=1}^N \mathrm{clip}_C(\phi_1(p_n)) + \xi \right) - \phi_2'\left(\frac{1}{N}\sum_{n=1}^N \mathrm{clip}_C(\phi_1'(p_n)) + \xi \right) \right\|_\infty \\ &\leq \sqrt{d}\left\| \phi_2\left(\frac{1}{N}\sum_{n=1}^N \mathrm{clip}_C(\phi_1(p_n)) + \xi \right) - \phi_2\left(\frac{1}{N}\sum_{n=1}^N \mathrm{clip}_C(\phi_1'(p_n)) + \xi \right) \right\|_\infty \\ &\qquad + \sqrt{d}\left\| \phi_2\left(\frac{1}{N}\sum_{n=1}^N \mathrm{clip}_C(\phi_1'(p_n)) + \xi \right) - \phi_2'\left(\frac{1}{N}\sum_{n=1}^N \mathrm{clip}_C(\phi_1'(p_n)) + \xi \right) \right\|_\infty \\ &\leq \sqrt{dD}(BW)^L dL(B \vee 1)^{L-1}(W+1)^L \mathfrak{C}_1(z_{1:N})\delta \\ &\qquad + \sqrt{d}\, L(B \vee 1)^{L-1}(W+1)^L((C \vee 1) + \|\xi\|_\infty)\delta \\ &\leq \sqrt{dD}\, dL(B \vee 1)^{2L-1}(W+1)^{2L}(\mathfrak{C}_1(z_{1:N}) + (C \vee 1) + \|\xi\|_\infty)\delta.\end{aligned}$$

By the definition of f_ϕ, we have

$$\begin{aligned}&|f_\phi(p_{1:N},\xi) - f_{\phi'}(p_{1:N},\xi)| \\ &= \left| \|\phi(p_{1:N},\xi) - \phi^*(p_{1:N},\xi)\|^2 - \|\phi'(p_{1:N},\xi) - \phi^*(p_{1:N},\xi)\|^2 \right| \\ &\leq 4F \cdot |\|\phi(p_{1:N},\xi) - \phi^*(p_{1:N},\xi)\| - \|\phi'(p_{1:N},\xi) - \phi^*(p_{1:N},\xi)\|| \\ &\leq 4F\|\phi(p_{1:N},\xi) - \phi'(p_{1:N},\xi)\| \\ &\leq 4F\sqrt{d}(\mathfrak{C}_1(z_{1:N}) + (C \vee 1) + \|\xi\|_\infty) \cdot \sqrt{dD}\, dL(B \vee 1)^{2L-1}(W+1)^{2L} \cdot \delta \\ &\leq 4F\left(\frac{1}{N}\sum_{n=1}^N |z_n| + \|\xi\|_\infty + \|\alpha\| + 2 \right) \cdot d^2\sqrt{D}(C \vee 1)L(B \vee 1)^{2L-1}(W+1)^{2L} \cdot \delta \\ &\leq 4F\mathfrak{C}_2(z_{1:N},\xi) \cdot d^2\sqrt{D}(C \vee 1)L(B \vee 1)^{2L-1}(W+1)^{2L} \cdot \delta,\end{aligned}$$

where we define $\mathfrak{C}_2(z_{1:N},\xi) := \mathfrak{C}_1(z_{1:N}) + \|\xi\|_\infty + 1$. Let P_M be the empirical measure of $(p_{1:N}, w, \xi)$ defined by M data. Then, we have

$$\begin{aligned}&\|f_\phi - f_{\phi'}\|_{L^2(P_M)}^2 \\ &= \int |f_\phi(p_{1:N},\xi) - f_{\phi'}(p_{1:N},\xi)|^2 \mathrm{d}P_M(p_{1:N},\xi) \\ &\leq \left(4F \cdot d^2\sqrt{D}(C \vee 1)L(B \vee 1)^{2L-1}(W+1)^{2L} \cdot \delta \right)^2 \cdot \frac{1}{M}\sum_{m=1}^M \mathfrak{C}_2(z_{1:N}^{(m)}, \xi^{(m)})^2.\end{aligned}$$

Since it holds

$$\mathbb{E}\left[\frac{1}{M}\sum_{m=1}^{M}\mathfrak{C}_2(z_{1:N}^{(m)},\xi^{(m)})^2\right]=\mathbb{E}_{z_{1:N},\xi}\left[\mathfrak{C}_2(z_{1:N},\xi)^2\right]\le A_8^2(s_{\mathrm{DP}}^2+1)\log(2D),$$

for some constant A_8, we have

$$\frac{1}{M}\sum_{m=1}^{M}\mathfrak{C}_2(z_{1:N}^{(m)},\xi^{(m)})^2\le\eta^{-1}\mathbb{E}\left[\frac{1}{M}\sum_{m=1}^{M}\mathfrak{C}_2(z_{1:N}^{(m)},\xi^{(m)})^2\right]\le A_8^2(s_{\mathrm{DP}}^2+1)\log(2D)\eta^{-1},$$

with probability $1-\eta$. Therefore, we have

$$\begin{aligned}\|f_\phi-f_{\phi'}\|_{L^2(P_M)}&\le 4FA_8\cdot d^2\sqrt{D}(C\vee 1)L\\&\qquad\cdot(B\vee 1)^{2L-1}(W+1)^{2L}\cdot\sqrt{(s_{\mathrm{DP}}^2+1)\log(2D)}\eta^{-1/2}\delta.\end{aligned}$$

This implies that the set $\{f_\phi\mid\phi\in\Phi,\mathrm{vec}(\phi)\in\mathcal{C}\}$ is a $(4FA_8\cdot d^2\sqrt{D}(C\vee 1)L(B\vee 1)^{2L-1}(W+1)^{2L}\cdot\sqrt{(s_{\mathrm{DP}}^2+1)\log(2D)}\eta^{-1/2}\delta)$-cover of the function class $\mathcal{F}$ with respect to $\|\cdot\|_{L^2(P_M)}$. By replacing δ with $(4FA_8\cdot d^2\sqrt{D}(C\vee 1)L(B\vee 1)^{2L-1}(W+1)^{2L}\cdot\sqrt{(s_{\mathrm{DP}}^2+1)\log(2D)}\eta^{-1/2})^{-1}\cdot\delta$, we obtain the following bound of the covering number of $\mathcal{F}$ for $0<\delta\le 4F^2$:

$$\begin{aligned}&\mathcal{N}(\delta;\mathcal{F},\|\cdot\|_{L^2(P_M)})\\&\le\left\{1+8FA_8\,d^2\sqrt{D}(C\vee 1)L(B\vee 1)^{2L}(W+1)^{2L}\right.\\&\qquad\left.\cdot\sqrt{(s_{\mathrm{DP}}^2+1)\log(2D)}\,\eta^{-1/2}\delta^{-1}\right\}^{\mathcal{D}}\\&\le\left\{\mathfrak{C}_3(L,W,B,C,d,D,s_{\mathrm{DP}})\eta^{-1/2}\delta^{-1}\right\}^{2L\mathcal{D}},\end{aligned}$$

where $\mathfrak{C}_3(L,W,B,C,d,D,s_{\mathrm{DP}}):=(1+4F^2+8FA_8)\cdot d^2\sqrt{D}L(B\vee 1)(W+1)(C\vee 1)\cdot(s_{\mathrm{DP}}^2+1)\log(2\mathrm{e}D)$. Therefore, we have

$$\begin{aligned}\log\mathcal{N}(\delta;\mathcal{F},\|\cdot\|_{L^2(P_M)})&\le 2L\mathcal{D}\log\left\{\mathfrak{C}_3(L,W,B,C,d,D,s_{\mathrm{DP}})\eta^{-1/2}\delta^{-1}\right\}\\&\le 8L^2W^2\log\left\{\mathfrak{C}_3(L,W,B,C,d,D,s_{\mathrm{DP}})\eta^{-1/2}\delta^{-1}\right\}.\end{aligned}$$

Using Dudley's entropy integral bound (Wainwright, 2019, Theorem 5.22), we have

$$\widehat{\mathfrak{R}}_M(\mathcal{F})\le\frac{A_6}{\sqrt{M}}\int_0^\infty\sqrt{\log\mathcal{N}(\delta;\mathcal{F},L_2(P_M))}\,\mathrm{d}\delta\,.$$

Since it holds

$$\mathcal{N}(\delta;\mathcal{F},\|\cdot\|_{L^2(P_M)})\le\mathcal{N}(\delta;\mathcal{F},\|\cdot\|_\infty)=1$$

for $\delta\ge 4F^2$, we have

$$\widehat{\mathfrak{R}}_M(\mathcal{F})\le\frac{2\sqrt{2}A_6LW}{\sqrt{M}}\int_0^{4F^2}\sqrt{\log\left\{\mathfrak{C}_3(L,W,B,C,d,D,s_{\mathrm{DP}})\eta^{-1/2}\delta^{-1}\right\}}\,\mathrm{d}\delta\,.$$

Since $\int_0^R\sqrt{\log(S/\delta)}\,\mathrm{d}\delta\le R\sqrt{\log(\mathrm{e}S/R)}$ for $0<R\le S$, we have

$$\widehat{\mathfrak{R}}_M(\mathcal{F})\le\frac{8\sqrt{2}F^2A_6LW}{\sqrt{M}}\sqrt{\log\left\{\frac{\mathrm{e}}{4F^2}\,\mathfrak{C}_3(L,W,B,C,d,D,s_{\mathrm{DP}})\eta^{-1/2}\right\}}.$$

Therefore, we have

$$\begin{aligned}\mathfrak{R}(\mathcal{F})=\mathbb{E}[\widehat{\mathfrak{R}}_M(\mathcal{F})]&\le\int_0^1\frac{8\sqrt{2}F^2A_6LW}{\sqrt{M}}\sqrt{\log\left\{\frac{\mathrm{e}}{4F^2}\,\mathfrak{C}_3(L,W,B,C,d,D,s_{\mathrm{DP}})\eta^{-1/2}\right\}}\,\mathrm{d}\eta\\&\le\frac{A_9LW}{\sqrt{M}}\sqrt{\log\left\{\frac{\mathrm{e}}{4F^2}\,\mathfrak{C}_3(L,W,B,C,d,D,s_{\mathrm{DP}})\right\}},\end{aligned}$$

Here we used $\mathfrak{C}_3 \geq 4F^2$ and Jensen's inequality to integrate over η; A_9 absorbs constants depending on F, $\overline{\alpha}$, and γ. Symmetrization inequality implies that

$$\begin{aligned}
&\mathbb{E}\left[\sup_{\phi\in\Phi}\left|\mathbb{E}[f_\phi(p_{1:N},\xi)] - \frac{1}{M}\sum_{m=1}^{M} f_\phi(p_{1:N}^{(m)},\xi^{(m)})\right|\right] \\
&\leq 2\mathfrak{R}(\mathcal{F}) \leq \frac{2A_9 LW}{\sqrt{M}}\sqrt{\log\left\{\frac{\mathrm{e}}{4F^2}\,\mathfrak{C}_3(L,W,B,C,d,D,s_{\mathrm{DP}})\right\}} \\
&\leq \frac{A_9' LW}{\sqrt{M}}\sqrt{\log\{dDL(B\vee 1)(W+1)(C\vee 1)\cdot(s_{\mathrm{DP}}^2+1)\}}.
\end{aligned}$$

where A_9' is a constant. This completes the proof. □

B.4 Constructing a Network with Small Error

In this section, we prove the following theorem, which constructs the network that outputs a vector close to the true parameter given N samples $p_{1:N}$ and DP noise ξ.

Theorem 10. *Suppose $d \geq 3$, $N \geq 2d$, $F \geq \|\alpha\|$, $0 < \epsilon \leq 1$, and $0 < \delta \leq \mathrm{e}^{-1}$, with F fixed. For any integer $1 \leq k \leq d$, set $C = A_C\sqrt{k}\log(\mathrm{e}+\nu^2)$ and $s_{\mathrm{DP}}^2 = A_{\mathrm{H}}C^2/\nu^2$, where A_C is sufficiently large and $A_{\mathrm{H}} \geq 16$ is fixed. There exists $\phi \in \Phi(L,W,B,C,k)$ with*

$$L \leq A_{15}\log[d(\mathrm{e}+\nu^2)], \qquad W \leq 8d, \qquad B \leq A_{15}d(1+\nu),$$

such that, for $\xi \sim \mathcal{N}(0, s_{\mathrm{DP}}^2 I_k)$ independent of the task,

$$\Delta_N := \mathbb{E}\|\phi(p_{1:N},\xi) - w\|^2 \leq A_{14}\left[\frac{k}{N} + \frac{k^2\log^2(\mathrm{e}+\nu^2)}{\nu^2} + \int_k^d t^{-2\gamma}\mathrm{d}t\right]. \tag{6}$$

The constants depend only on $\overline{\alpha}, \gamma$ and the fixed calibrations.

To construct the network, we use the following lemma:

Lemma 11 (Adapted from Schmidt-Hieber (2020, Supplement, Lemma A.2))**.** *Let $i \in \mathbb{Z}_{\geq 0}$. There exists a neural network $\phi_{\mathrm{mult}} \in \Phi_{\mathrm{MLP}}(L,W,B)$ with*

$$L = i+5, \quad W \leq 6, \quad B = 1,$$

satisfying the following three conditions:

- $|\phi_{\mathrm{mult}}(x,y) - xy| \leq 2^{-i}$ *for all* $(x,y) \in [0,1]^2$.
- $|\phi_{\mathrm{mult}}(x,y)| \leq 1$ *for all* $(x,y) \in \mathbb{R}^2$.
- *For $(x,y) \in [0,1]^2$, if $xy = 0$, then $\phi_{\mathrm{mult}}(x,y) = 0$.*

For $i \geq 1$, the cited construction has $i+4$ hidden layers, hence $i+5$ affine layers in our convention, and its final clipping operation bounds the output on all of $\mathbb{R}^2$. For $i = 0$, the zero network with five affine layers satisfies all three conditions. We prove the following corollary using the lemma above.

Corollary 12. *Let $\varepsilon > 0$, $\rho_x, \rho_y \geq 1$. There exists a neural network $\phi_{\mathrm{mult}} \in \Phi_{\mathrm{MLP}}(L,W,B)$ with*

$$L \leq 6 + \left\lceil \log_2\left(1 + \frac{4\rho_x\rho_y}{\varepsilon}\right)\right\rceil, \quad W \leq 8, \quad B \leq 4\rho_x\rho_y,$$

satisfying the following three conditions:

- $|\phi_{\mathrm{mult}}(x,y) - xy| \leq \varepsilon$ *for all* $x \in [-\rho_x,\rho_x]$ *and* $y \in [-\rho_y,\rho_y]$.
- $|\phi_{\mathrm{mult}}(x,y)| \leq \rho_x\rho_y$ *for all* $x,y \in \mathbb{R}$.
- $|\phi_{\mathrm{mult}}(x,y) - xy| \leq \rho_x(\rho_y + |y|)$ *for all* $x \in [-\rho_x,\rho_x]$ *and* $y \in \mathbb{R}$.

Proof. Let $\phi_1, \phi_2, \phi_3 \in \Phi_{\mathrm{MLP}}(L, W, B)$ with

$$L = 1, \quad W \leq 2, \quad B = \rho_x \rho_y,$$

be neural networks defined as

$$\phi_1(x) := \frac{x}{2\rho_x} + \frac{1}{2}, \quad \phi_2(y) := \frac{y}{2\rho_y} + \frac{1}{2}, \quad \phi_3(x, y) := \rho_y x + \rho_x y + \rho_x \rho_y.$$

Then, we have

$$\phi_1(x)\phi_2(y) = \frac{xy}{4\rho_x\rho_y} + \frac{x}{4\rho_x} + \frac{y}{4\rho_y} + \frac{1}{4} = \frac{xy}{4\rho_x\rho_y} + \frac{1}{4\rho_x\rho_y}\phi_3(x, y),$$

which implies

$$xy = 4\rho_x\rho_y\phi_1(x)\phi_2(y) - \phi_3(x, y).$$

Now, we have $\phi_1(x), \phi_2(y) \in [0, 1]$ if $x \in [-\rho_x, \rho_x]$ and $y \in [-\rho_y, \rho_y]$. Therefore, Lemma 11 implies that there exists a neural network $\psi \in \Phi_{\mathrm{MLP}}(L, W, B)$ with

$$L \leq 5 + \left\lceil \log_2 \left(1 + \frac{4\rho_x\rho_y}{\varepsilon} \right) \right\rceil, \quad W \leq 6, \quad B \leq 1,$$

satisfying the following two statements:

- If $x \in [-\rho_x, \rho_x]$ and $y \in [-\rho_y, \rho_y]$, it holds $|\psi(\phi_1(x), \phi_2(y)) - \phi_1(x)\phi_2(y)| \leq \frac{\varepsilon}{4\rho_x\rho_y}$.
- For all $x, y \in \mathbb{R}$, it holds $|\psi(\phi_1(x), \phi_2(y))| \leq 1$.

We define

$$\begin{aligned}
\phi_4(x, y) &:= 4\rho_x\rho_y\psi(\phi_1(x), \phi_2(y)) - \phi_3(x, y), \\
\phi_5(x, y) &:= \max\{-\rho_x\rho_y, \min\{\rho_x\rho_y, \phi_4(x, y)\}\}.
\end{aligned}$$

To realize ϕ_5 as an MLP, absorb ϕ_1, ϕ_2 into the first affine layer of ψ and propagate $\eta(\phi_3/(4\rho_x\rho_y))$ and $\eta(-\phi_3/(4\rho_x\rho_y))$ through its hidden layers using two additional units. Since $\phi_3/(4\rho_x\rho_y) = x/(4\rho_x) + y/(4\rho_y) + 1/4$, these units use weights and biases bounded by 1; the first affine layer of the composed ψ has weights bounded by 1 and biases bounded by 2. For the output clipping, use

$$\phi_5(x, y) = 4\rho_x\rho_y \left[\eta\left(\frac{\phi_4(x, y)}{4\rho_x\rho_y} + \frac{1}{4} \right) - \eta\left(\frac{\phi_4(x, y)}{4\rho_x\rho_y} - \frac{1}{4} \right) \right] - \rho_x\rho_y.$$

This replaces the output affine layer of ψ by a width-two ReLU layer and a final affine layer, increasing the depth by one. The resulting weights and biases are bounded by $4\rho_x\rho_y$.

Then we have $\phi_5 \in \Phi_{\mathrm{MLP}}(L, W, B)$ with

$$L \leq 6 + \left\lceil \log_2 \left(1 + \frac{4\rho_x\rho_y}{\varepsilon} \right) \right\rceil, \quad W \leq 8, \quad B \leq 4\rho_x\rho_y.$$

Moreover, for all $x, y \in \mathbb{R}$, we have

$$|\phi_5(x, y)| \leq \rho_x\rho_y.$$

For $x \in [-\rho_x, \rho_x]$ and $y \in [-\rho_y, \rho_y]$, we have $|xy| \leq \rho_x\rho_y$, which yields

$$\begin{aligned}
|\phi_5(x, y) - xy| &\leq |\phi_4(x, y) - xy| \\
&\leq 4\rho_x\rho_y|\psi(\phi_1(x), \phi_2(y)) - \phi_1(x)\phi_2(y)| \leq \varepsilon.
\end{aligned}$$

Moreover, for any $x \in [-\rho_x, \rho_x],\ y \in \mathbb{R}$, we have

$$|\phi_5(x, y) - xy| \leq |\phi_5(x, y)| + |xy| \leq \rho_x\rho_y + \rho_x|y| \leq \rho_x(\rho_y + |y|).$$

This completes the proof. □

Using the corollary above, we prove Theorem 10.

Proof of Theorem 10. We first define the matrix $U_k \in \mathbb{R}^{d\times k}$ as

$$(U_k)_{i,j} = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

Then, both $U_k U_k^\top$ and $I - U_k U_k^\top$ are projection matrices. Therefore, we have $(I - U_k U_k^\top)^2 = I - U_k U_k^\top$. Moreover, since it holds

$$\mathbb{E}[ww^\top] = \frac{1}{3}\operatorname{diag}(\alpha_1^2, \dots, \alpha_d^2),$$

we have

$$U_k^\top \mathbb{E}[ww^\top] U_k = \frac{1}{3}\operatorname{diag}(\alpha_1^2, \dots, \alpha_k^2).$$

Hence, we have

$$\begin{aligned}
\mathbb{E}[\|U_k U_k^\top w - w\|^2] &= \mathbb{E}[\|(I - U_k U_k^\top)w\|^2] \\
&= \mathbb{E}[w^\top (I - U_k U_k^\top)^2 w] \\
&= \mathbb{E}[w^\top (I - U_k U_k^\top) w] \\
&= \operatorname{tr}\big((I - U_k U_k^\top)\mathbb{E}[ww^\top]\big) \\
&= \operatorname{tr}\big(\mathbb{E}[ww^\top]\big) - \operatorname{tr}\big(U_k U_k^\top \mathbb{E}[ww^\top]\big) \\
&= \operatorname{tr}\big(\mathbb{E}[ww^\top]\big) - \operatorname{tr}\big(U_k^\top \mathbb{E}[ww^\top] U_k\big) \\
&= \frac{1}{3}\sum_{i=1}^{d} \alpha_i^2 - \frac{1}{3}\sum_{i=1}^{k} \alpha_i^2 = \frac{1}{3}\sum_{i=k+1}^{d} \alpha_i^2.
\end{aligned}$$

Since $\alpha_i \le \frac{\overline{\alpha}}{i^\gamma}$ for $i = 1, \dots, d$, we have

$$\sum_{i=k+1}^{d} \alpha_i^2 \le \overline{\alpha}^2 \int_k^d \frac{\mathrm{d}t}{t^{2\gamma}} \left(\le \frac{\overline{\alpha}^2}{2\gamma - 1} \frac{1}{k^{2\gamma-1}} \right),$$

which implies

$$\mathbb{E}[\|U_k U_k^\top w - w\|^2] \le \frac{\overline{\alpha}^2}{3} \int_k^d \frac{\mathrm{d}t}{t^{2\gamma}}.$$

Below, we consider approximating $U_k U_k^\top w$ instead of w. We denote $P_k := U_k U_k^\top \in \mathbb{R}^{d\times d}$.

Approximating $P_k w$ using $x_{1:N}$ and $y_{1:N}$ Since it holds

$$\mathbb{E}[y_n P_k x_n \mid w] = \mathbb{E}[(w^\top x_n + z_n) P_k x_n] = P_k \mathbb{E}[x_n x_n^\top] w + \mathbb{E}[z_n] P_k \mathbb{E}[x_n] = P_k w$$

Moreover, since the pairs $(x_1, y_1), \dots, (x_N, y_N)$ are conditionally independent given w, we have

$$\begin{aligned}
\mathbb{E}\left[\left\| \frac{1}{N}\sum_{n=1}^{N} y_n P_k x_n - P_k w \right\|^2 \,\middle|\, w \right] &= \frac{1}{N^2}\sum_{n=1}^{N} \mathbb{E}\left[\|y_n P_k x_n - P_k w\|^2 \,\middle|\, w \right] \\
&= \frac{1}{N}\mathbb{E}\left[\|y_1 P_k x_1 - P_k w\|^2 \,\middle|\, w \right].
\end{aligned}$$

Since $y_1 = w^\top x_1 + z_1$, we have

$$\begin{aligned}
\mathbb{E}\left[(y_1 P_k x_1)^\top P_k w \mid w \right] &= \mathbb{E}\left[y_1 x_1^\top P_k P_k w \mid w \right] \\
&= \mathbb{E}\left[w^\top P_k P_k w \mid w \right] \\
&= \|P_k w\|^2.
\end{aligned}$$

Therefore, we have

$$\mathbb{E}\left[\|y_1 P_k x_1 - P_k w\|^2 \;\middle|\; w\right] = \mathbb{E}\left[\|y_1 P_k x_1\|^2 \;\middle|\; w\right] - \|P_k w\|^2.$$

As for the first term, since z and x are independent, we have

$$\begin{aligned}\mathbb{E}\left[\|y_1 P_k x_1\|^2 \;\middle|\; w\right] &= \mathbb{E}\left[\left\|(w^\top x_1 + z_1) P_k x_1\right\|^2 \;\middle|\; w\right] \\ &= \mathbb{E}\left[(w^\top x_1)^2 \|P_k x_1\|^2 \;\middle|\; w\right] + \mathbb{E}[z_1^2]\mathbb{E}\left[\|P_k x_1\|^2\right] \\ &= \frac{d}{d+2}\left(k\|w\|^2 + 2\|P_k w\|^2\right) + k.\end{aligned}$$

Here, the last equality follows from the fourth-moment formula for $x_1 \sim \mathrm{Unif}(\sqrt{d}\mathbb{S}^{d-1})$:

$$\mathbb{E}[x_{1,i} x_{1,j} x_{1,a} x_{1,b}] = \frac{d}{d+2}(\mathbf{1}_{i=j}\mathbf{1}_{a=b} + \mathbf{1}_{i=a}\mathbf{1}_{j=b} + \mathbf{1}_{i=b}\mathbf{1}_{j=a}).$$

Indeed, since $P_k^2 = P_k$ and $\mathrm{tr}(P_k) = k$, we have

$$\begin{aligned}\mathbb{E}\left[(w^\top x_1)^2 \|P_k x_1\|^2 \;\middle|\; w\right] &= \mathbb{E}\left[(w^\top x_1)^2 x_1^\top P_k x_1 \mid w\right] \\ &= \sum_{i,j,a,b=1}^{d} w_i w_j (P_k)_{ab} \mathbb{E}[x_{1,i} x_{1,j} x_{1,a} x_{1,b}] \\ &= \frac{d}{d+2}\left(\|w\|^2 \,\mathrm{tr}(P_k) + 2w^\top P_k w\right) \\ &= \frac{d}{d+2}\left(k\|w\|^2 + 2\|P_k w\|^2\right),\end{aligned}$$

and $\mathbb{E}[z_1^2]\mathbb{E}[\|P_k x_1\|^2] = \mathrm{tr}(P_k) = k$. Therefore, we have

$$\begin{aligned}\mathbb{E}\left[\|y_1 P_k x_1 - P_k w\|^2 \;\middle|\; w\right] &= \frac{d}{d+2}\left(k\|w\|^2 + 2\|P_k w\|^2\right) + k - \|P_k w\|^2 \\ &= \frac{dk}{d+2}\|w\|^2 + \frac{d-2}{d+2}\|P_k w\|^2 + k \\ &\le \frac{dk}{d+2}\|\alpha\|^2 + \frac{(d-2)k}{d+2}\|\alpha\|^2 + k \\ &\le k(2\|\alpha\|^2 + 1).\end{aligned}$$

This implies that

$$\mathbb{E}\left[\left\|\frac{1}{N}\sum_{n=1}^{N} y_n P_k x_n - P_k w\right\|^2 \;\middle|\; w\right] \le \frac{k(2\|\alpha\|^2 + 1)}{N}.$$

Evaluating truncation error Next, we upper bound the error between $\frac{1}{N}\sum_{n=1}^{N} y_n P_k x_n$ and its data-wise truncation $U_k\left\{\frac{1}{N}\sum_{n=1}^{N} \mathrm{clip}_C(y_n U_k^\top x_n)\right\}$ for $C > 0$. We have

$$\begin{aligned}&\mathbb{E}\left[\left\|U_k\left\{\frac{1}{N}\sum_{n=1}^{N} \mathrm{clip}_C(y_n U_k^\top x_n)\right\} - \frac{1}{N}\sum_{n=1}^{N} y_n P_k x_n\right\|^2\right] \\ &\quad= \mathbb{E}\left[\left\|\left\{\frac{1}{N}\sum_{n=1}^{N} \mathrm{clip}_C(y_n U_k^\top x_n)\right\} - \frac{1}{N}\sum_{n=1}^{N} y_n U_k^\top x_n\right\|^2\right] \\ &\quad\le \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}\left[\left\|\mathrm{clip}_C(y_n U_k^\top x_n) - y_n U_k^\top x_n\right\|^2\right] \\ &\quad\le \mathbb{E}\left[\left\|\mathrm{clip}_C(y_1 U_k^\top x_1) - y_1 U_k^\top x_1\right\|^2\right].\end{aligned}$$

Therefore, it is sufficient to bound $\mathbb{E}\Big[\big\|\mathrm{clip}_C(y_1 U_k^\top x_1) - y_1 U_k^\top x_1\big\|^2\Big]$. Let $Y := y_1$ and $R := \big\|U_k^\top x_1\big\|$. By the definition $\mathrm{clip}_C(v) = \min\{1, C/\|v\|\}v$, we have $\|\mathrm{clip}_C(v) - v\| = (\|v\| - C)_+$ for any vector v. Applying this with $v = YU_k^\top x_1$, whose norm is $|Y|R$, gives

$$\big\|\mathrm{clip}_C(YU_k^\top x_1) - YU_k^\top x_1\big\|^2 = (|Y|R - C)_+^2 \le Y^2 R^2 \mathbf{1}_{\{|Y|R>C\}}.$$

Since $\|w\| \le \|\alpha\| \le \overline{\alpha}\sqrt{1 + (2\gamma-1)^{-1}}$, spherical linear-functional tails and the independent Gaussian residual give, uniformly in w,

$$\mathbb{P}(|Y| > a_1\sqrt{t} \mid w) \le 2\mathrm{e}^{-t}, \qquad t \ge 1.$$

To bound R, take a $1/2$-net $\mathcal{U}$ of $\mathbb{S}^{k-1}$ with $|\mathcal{U}| \le 5^k$. Then $R \le 2\max_{u\in\mathcal{U}} |\langle U_k u, x_1\rangle|$. Each $U_k u$ is a unit vector, so a union bound using the spherical tail from Lemma 14 gives

$$\mathbb{P}(R > a_2(\sqrt{k} + \sqrt{t})) \le 2\mathrm{e}^{-t}, \qquad t \ge 1,$$

for an absolute a_2. No independence of spherical coordinates is used. Combining the two tail bounds, with a sufficiently large fixed a_3, gives

$$\mathbb{P}(|Y|R > a_3(\sqrt{kt} + t) \mid w) \le 4\mathrm{e}^{-t}.$$

For $k, t \ge 1$, $\sqrt{kt} + t \le 2\sqrt{k}\,t$. Thus, when $C \ge 2a_3\sqrt{k}$, take $t = C/(2a_3\sqrt{k})$ to obtain an exponential tail in $C/\sqrt{k}$. Integrating the preceding individual tails also gives $\mathbb{E}[Y^8 \mid w] \lesssim 1$ and $\mathbb{E}[R^8] \lesssim k^4$, whence

$$\mathbb{E}[Y^4 R^4 \mid w] \le \sqrt{\mathbb{E}[Y^8 \mid w]\mathbb{E}[R^8]} \lesssim k^2.$$

Cauchy–Schwarz now yields

$$\begin{aligned}\mathbb{E}\left[\big\|\mathrm{clip}_C(YU_k^\top x_1) - YU_k^\top x_1\big\|^2 \mid w\right] &\le \mathbb{E}[Y^2R^2\mathbf{1}_{\{|Y|R>C\}} \mid w] \\ &\le A_{13}k\exp\left(-A_{12}C/\sqrt{k}\right). \end{aligned} \tag{7}$$

Here $A_{13}, A_{12} > 0$ depend only on $\overline{\alpha}, \gamma$.

Approximating $U_k\Big\{\frac{1}{N}\sum_{n=1}^N \mathrm{clip}_C(y_n U_k^\top x_n)\Big\}$ via DP-DeepSets Let $n \in [N]$, $i \in [d]$, $0 < \varepsilon \le 1$ and $\rho \ge 1$. Corollary 12 implies that there exists a neural network $\phi_1 \in \Phi_{\mathrm{MLP}}^{2,1}(L, W, B)$ with

$$L \le 6 + \left\lceil \log_2\left(1 + \frac{4\sqrt{d}\rho}{\varepsilon}\right)\right\rceil, \quad W \le 8, \quad B \le 4\sqrt{d}\rho.$$

satisfying $|\phi_1(x_{n,i}, y_n)| \le \sqrt{d}\rho$ and

$$|\phi_1(x_{n,i}, y_n) - x_{n,i}y_n| \le \begin{cases} \varepsilon & (|y_n| \le \rho), \\ \sqrt{d}(\rho + |y_n|) & (\text{otherwise}). \end{cases}$$

Therefore, we have

$$\mathbb{E}\Big[\{\phi_1(x_{n,i}, y_n) - x_{n,i}y_n\}^2\Big] \le \mathbb{E}\Big[\mathbf{1}_{\{|y_n|\le\rho\}}\varepsilon^2 + \mathbf{1}_{\{|y_n|>\rho\}}d(\rho + |y_n|)^2\Big].$$

Since $(|y| + \rho)^2 \le 4y^2 \le 4y^4/\rho^2$ for $|y| \ge \rho$, we have

$$\mathbb{E}\Big[\{\phi_1(x_{n,i}, y_n) - x_{n,i}y_n\}^2\Big] \le \varepsilon^2 + \mathbb{E}\left[\frac{4dy_n^4}{\rho^2}\right] = \varepsilon^2 + \frac{4d}{\rho^2}\mathbb{E}\big[y_n^4\big].$$

We bound the expectation $\mathbb{E}\big[y^4\big]$. It holds

$$\begin{aligned}\mathbb{E}[y_n^4] &= \mathbb{E}[(w^\top x_n + z_n)^4] \\ &= \mathbb{E}[(w^\top x_n)^4] + 6\mathbb{E}[(w^\top x_n)^2]\mathbb{E}[z_n^2] + \mathbb{E}[z_n^4] \\ &= \mathbb{E}[(w^\top x_n)^4] + 6\mathbb{E}[(w^\top x_n)^2] + 3.\end{aligned}$$

Moreover, we have

$$\mathbb{E}[(w^\top x_n)^2 \mid w] = w^\top \mathbb{E}[x_n x_n^\top] w = \|w\|^2 \le A_{10}^2,$$
$$\mathbb{E}[(w^\top x_n)^4 \mid w] = \frac{3d}{d+2}\|w\|^4 \le 3A_{10}^4,$$

where A_{10} is a constant that depends only on α. This implies that

$$\mathbb{E}[y_n^4] \le 3A_{10}^4 + 6A_{10}^2 + 3 \le A_{11},$$

for some constant A_{11} that depends only on α. Taking $\rho := 2\sqrt{A_{11}} \cdot \sqrt{d}\varepsilon^{-1}$, we have

$$\mathbb{E}\Big[\{\phi_1(x_{n,i}, y_n) - x_{n,i}y_n\}^2\Big] \le \varepsilon^2 + \frac{4dA_{11}}{\rho^2} = 2\varepsilon^2.$$

Next, let $\phi_2 \in \Phi_{\mathrm{MLP}}(L, W, B)$ with

$$L \le 6 + \left\lceil \log_2\left(1 + \frac{8\sqrt{A_{11}}d}{\varepsilon^2}\right)\right\rceil, \quad W \le 8d, \quad B \le 8\sqrt{A_{11}} \cdot d\varepsilon^{-1},$$

be a neural network defined as

$$\phi_2(x_n, y_n) = [\phi_1(x_{n,1}, y_n),\ \ldots,\ \phi_1(x_{n,k}, y_n)]^\top \in \mathbb{R}^k$$

Then, we have

$$\begin{aligned}\mathbb{E}\Big[\big\|\phi_2(x_n, y_n) - y_n U_k^\top x_n\big\|^2\Big] &= \sum_{i=1}^k \mathbb{E}\Big[\{\phi_1(x_{n,i}, y_n) - x_{n,i}y_n\}^2\Big] \\ &\le 2k\varepsilon^2.\end{aligned}$$

Finally, let $\phi_3 \in \Phi_{\mathrm{MLP}}(L, W, B)$ with $L = 1$, $W = d$ and $B = 1$ be a neural network defined as

$$\phi_3 : \mathbb{R}^k \to \mathbb{R}^d, \quad v \mapsto U_k v.$$

Then, we have

$$\begin{aligned}&\mathbb{E}\left[\left\|\phi_3\left(\frac{1}{N}\sum_{n=1}^N \mathrm{clip}_C(\phi_2(x_n, y_n)) + \xi\right) - U_k\left\{\frac{1}{N}\sum_{n=1}^N \mathrm{clip}_C(y_n U_k^\top x_n)\right\}\right\|^2\right] \\ &\le \mathbb{E}\left[\left\|U_k\left\{\frac{1}{N}\sum_{n=1}^N \mathrm{clip}_C(\phi_2(x_n, y_n)) - \frac{1}{N}\sum_{n=1}^N \mathrm{clip}_C(y_n U_k^\top x_n)\right\} + U_k\xi\right\|^2\right] \\ &\le \mathbb{E}\left[\left\|\left\{\frac{1}{N}\sum_{n=1}^N \mathrm{clip}_C(\phi_2(x_n, y_n)) - \frac{1}{N}\sum_{n=1}^N \mathrm{clip}_C(y_n U_k^\top x_n)\right\} + \xi\right\|^2\right] \\ &\le 2\mathbb{E}\left[\left\|\frac{1}{N}\sum_{n=1}^N \left\{\mathrm{clip}_C(\phi_2(x_n, y_n)) - \mathrm{clip}_C(y_n U_k^\top x_n)\right\}\right\|^2\right] + 2\mathbb{E}[\|\xi\|^2] \\ &\le \frac{2}{N}\sum_{n=1}^N \mathbb{E}\Big[\big\|\mathrm{clip}_C(\phi_2(x_n, y_n)) - \mathrm{clip}_C(y_n U_k^\top x_n)\big\|^2\Big] + 2ks_{\mathrm{DP}}^2 \\ &\le \frac{2}{N}\sum_{n=1}^N \mathbb{E}\Big[\big\|\phi_2(x_n, y_n) - y_n U_k^\top x_n\big\|^2\Big] + 2ks_{\mathrm{DP}}^2 \\ &\le 4k\varepsilon^2 + 2ks_{\mathrm{DP}}^2.\end{aligned}$$

Setting $\varepsilon := \min\{1, s_{\mathrm{DP}}/\sqrt{2}\}$, we have

$$\mathbb{E}\left[\left\|\phi_3\left(\frac{1}{N}\sum_{n=1}^N \mathrm{clip}_C(\phi_2(x_n, y_n)) + \xi\right) - U_k\left\{\frac{1}{N}\sum_{n=1}^N \mathrm{clip}_C(y_n U_k^\top x_n)\right\}\right\|^2\right] \le 4ks_{\mathrm{DP}}^2.$$

Finishing the proof If we set

$$\widetilde{\phi}(p_{1:N}, \xi) := \phi_3\left(\frac{1}{N}\sum_{n=1}^{N} \mathrm{clip}_C(\phi_2(x_n, y_n)) + \xi\right),$$

then we have

$$\begin{aligned}
&\mathbb{E}\left[\left\|\widetilde{\phi}(p_{1:N}, \xi) - w\right\|^2\right] \\
&\le 4\left\{\mathbb{E}\left[\left\|\widetilde{\phi}(p_{1:N}, \xi) - U_k\left\{\frac{1}{N}\sum_{n=1}^{N} \mathrm{clip}_C(y_n U_k^\top x_n)\right\}\right\|^2\right]\right. \\
&\qquad +\mathbb{E}\left[\left\|U_k\left\{\frac{1}{N}\sum_{n=1}^{N} \mathrm{clip}_C(y_n U_k^\top x_n)\right\} - \frac{1}{N}\sum_{n=1}^{N} y_n P_k x_n\right\|^2\right] \\
&\qquad\qquad \left.+\mathbb{E}\left[\left\|\frac{1}{N}\sum_{n=1}^{N} y_n P_k x_n - P_k w\right\|^2\right] + \mathbb{E}\left[\|P_k w - w\|^2\right]\right\} \\
&\le 16 k s_{\mathrm{DP}}^2 + 4A_{13} k \exp\left(-A_{12} C/\sqrt{k}\right) + \frac{4k(2\|\alpha\|^2 + 1)}{N} + \frac{4\overline{\alpha}^2}{3}\int_k^d \frac{\mathrm{d}t}{t^{2\gamma}}.
\end{aligned}$$

Define $\phi := \mathrm{clip}_F \circ \widetilde{\phi}$. Since $\|w\| \le \|\alpha\| \le F$ and clip_F is the Euclidean projection onto the radius-F ball,

$$\|\phi(p_{1:N}, \xi) - w\| \le \left\|\widetilde{\phi}(p_{1:N}, \xi) - w\right\|.$$

Thus the preceding error bound also holds for ϕ, which has the output clipping required in the definition of $\Phi(L, W, B, C, D)$.

Set $C = A_C\sqrt{k}\log(\mathrm{e} + \nu^2)$, with A_C large enough that $C \ge 2a_3\sqrt{k}$ and $A_{12}A_C \ge 1$. Then

$$k\exp\left(-A_{12}C/\sqrt{k}\right) \le \frac{k}{\mathrm{e} + \nu^2} \le \frac{k}{\nu^2} \lesssim k s_{\mathrm{DP}}^2, \qquad k s_{\mathrm{DP}}^2 = A_{\mathrm{H}} A_C^2 \frac{k^2 \log^2(\mathrm{e} + \nu^2)}{\nu^2}.$$

Substitution into the preceding four-term decomposition proves (6). The multiplication tolerance $\varepsilon = \min\{1, s_{\mathrm{DP}}/\sqrt{2}\}$ satisfies

$$4k\varepsilon^2 + 2k s_{\mathrm{DP}}^2 \le 4k s_{\mathrm{DP}}^2, \qquad \varepsilon^{-1} \lesssim 1 + \nu, \qquad \varepsilon^{-2} \lesssim 1 + \nu^2,$$

since $C \gtrsim 1$. Inserting these bounds in the existing product-network construction gives

$$L \le A_{15}\log[d(\mathrm{e} + \nu^2)], \qquad W \le 8d, \qquad B \le A_{15} d(1 + \nu), \qquad D = k.$$

The linear decoder $v \mapsto U_k v$ can have the same depth as the encoder by carrying $\mathrm{ReLU}(v)$ and $\mathrm{ReLU}(-v)$ through identity hidden layers and subtracting them at the output. This uses width at most $2k$ and weights bounded by 1; encoder hidden layers can likewise be padded. Thus the constructed oracle lies in exactly the same class as the ERM. The matrix U_k is used to establish existence, and is not supplied to the learned network. All choices remain valid for $k = d$ and $\nu < 1$. This completes the proof.

□

B.5 Proof of Theorem 6

We finally prove our target statement. Let $\widehat{\phi} := \underset{\phi \in \Phi(L,W,B,C,D)}{\arg\min} \widehat{R}(\phi)$ with the parameters specified in Theorem 6. Choose the fixed oracle ϕ^* from Theorem 10 in the same class, padding depths and widths as above if necessary.

Let P_M denote the empirical measure of the M public tasks and let P denote expectation over an independent test task $(p_{1:N}, w, \xi)$. For $\phi \in \Phi(L, W, B, C, D)$, define

$$f_\phi(p_{1:N}, \xi) := \|\phi(p_{1:N}, \xi) - \phi^*(p_{1:N}, \xi)\|^2.$$

Taking expectation over the public tasks and the independent test task before applying the empirical-to-population comparison gives

$$\begin{aligned}
&\mathbb{E}\left[\left\|\widehat{\phi}(p_{1:N},\xi)-w\right\|^2\right]\\
&\lesssim \mathbb{E}\left[\|\phi^*(p_{1:N},\xi)-w\|^2\right]+\mathbb{E}\left[Pf_{\widehat{\phi}}\right]\\
&\lesssim \mathbb{E}\left[\|\phi^*(p_{1:N},\xi)-w\|^2\right]+\mathbb{E}\left[P_Mf_{\widehat{\phi}}\right]+\mathbb{E}\left[\sup_{\phi\in\Phi}|Pf_\phi-P_Mf_\phi|\right].
\end{aligned}$$

Here the second inequality holds pointwise in the public tasks because $Pf_{\widehat{\phi}} \le P_Mf_{\widehat{\phi}} + \sup_{\phi\in\Phi}|Pf_\phi - P_Mf_\phi|$.

The three terms above are bounded by Theorem 10, Lemma 8, and Lemma 9, respectively. More explicitly, put

$$\begin{aligned}
\overline{\Delta}_N &= \frac{k}{N}+\frac{k^2\log^2(\mathrm{e}+\nu^2)}{\nu^2}+\int_k^d t^{-2\gamma}\mathrm{d}t,\\
\mathcal{G} &= \log\{dkL(B\vee 1)(W+1)(C\vee 1)(1+s_{\mathrm{DP}}^2)\}.
\end{aligned}$$

The existing empirical-to-test lemma applies to the unchanged class and loss. With $D=k$, the decomposition therefore gives

$$E_{\mathrm{HN}}(k)\lesssim \tau^{-1}\left[\overline{\Delta}_N+\left(\frac{k}{N}+\frac{LW^2}{MN}\right)\mathcal{A}_N\right]+\frac{LW}{\sqrt{M}}\sqrt{\mathcal{G}}. \tag{8}$$

Since $W\simeq d$, $N\ge 2d$, and $M\ge 1$, $d^2/(MN)\le d/\sqrt{M}$. This absorbs the extra decoder complexity term and proves the stated upper bound.

Explicit logarithmic factors For clarity, one common factor suffices for all $1\le k\le d$:

$$q:=\log[\mathrm{e}N(1+\nu^{-1})],\qquad \Lambda:=q^5. \tag{9}$$

Indeed, $\nu\le N$ under the theorem's privacy ranges, and $d,k\le N$. The specified sizes imply $L\lesssim q$, $\log J,\log J_\theta\lesssim q^2$, $\mathcal{A}_N\lesssim q^2$, $\tau^{-1}\lesssim q$, $\mathcal{G}\lesssim q$, and $\log^2(\mathrm{e}+\nu^2)\lesssim q^2$. For the noise terms this follows from $s_{\mathrm{DP}}^2\lesssim k\log^2(\mathrm{e}+\nu^2)/\nu^2$, whose positive logarithm is $O(q)$. Consequently a fixed constant A'_{H} gives

$$E_{\mathrm{HN}}(k)\le A'_{\mathrm{H}}\Lambda\left[\frac{d}{\sqrt{M}}+\frac{k}{N}+\frac{k^2}{\nu^2}+\int_k^d t^{-2\gamma}\mathrm{d}t\right]. \tag{10}$$

The constant A'_{H} depends only on $F,\overline{\alpha},\gamma$ and the fixed calibration constants. There is no logarithmic dependence on M. The one-shot embedding release has replacement sensitivity $2C/N$. Its Gaussian Rényi divergence of order $\lambda>1$ is at most $\lambda\rho_{\mathrm{H}}$, with $\rho_{\mathrm{H}}=2C^2/(N^2s_{\mathrm{DP}}^2)=2\epsilon^2/(A_{\mathrm{H}}\log(1/\delta))$. The conversion detailed in Appendix C proves (ϵ,δ)-DP, including $\epsilon=1$, and post-processing by the public-trained decoder preserves it.

B.6 Optimization and Comparison under Privacy Constraints

Let

$$a=\frac{2\gamma-1}{2\gamma},\qquad b=\frac{2(2\gamma-1)}{2\gamma+1},\qquad k_*:=\max\left\{1,\left\lfloor\min\{d,N^{1/(2\gamma)},\nu^{2/(2\gamma+1)}\}\right\rfloor\right\}.$$

This is an integer in $[1,d]$. First suppose $\nu\ge 1$, and let u be the minimum inside the floor; then $u/2\le k_*\le u$. If $u=N^{1/(2\gamma)}<d$, then $k_*/N\lesssim N^{-a}$ and $k_*^2/\nu^2\le u^2/\nu^2\le u^{1-2\gamma}=N^{-a}$. If $u=\nu^{2/(2\gamma+1)}<d$, then $k_*^2/\nu^2\le\nu^{-b}$ and $k_*/N\le u/N\le u^{1-2\gamma}=\nu^{-b}$. In either case the tail integral is at most $k_*^{1-2\gamma}/(2\gamma-1)$, of the same order. If the minimum is $u=d$, then $k_*=d$, the integral is zero, and $d/N\le N^{-a}$, $d^2/\nu^2\le\nu^{-b}$. Thus, after increasing a fixed constant A_{opt},

$$E_{\mathrm{HN}}(k_*)\le A_{\mathrm{opt}}\Lambda(d/\sqrt{M}+N^{-a}+\nu^{-b}),\qquad \nu\ge 1. \tag{11}$$

For $\nu < 1$ the original upper bound still holds with $k_* = 1$. Also $E_{\mathrm{HN}} \le 4F^2$ by output clipping, so the optimized-rate inequality extends to this range after adjusting A_{opt}, since $\nu^{-b} > 1$. For $\gamma = 1$, the choice is the rounded version of $\min\{d, \sqrt{N}, \nu^{2/3}\}$, and the optimized terms are $N^{-1/2} + \nu^{-2/3}$.

For the privacy window, take $k = \max\{1, \lfloor N^{1/(2\gamma)} \rfloor\} < d$. Then $\int_k^d t^{-2\gamma} \mathrm{d}t \lesssim k/N$. If $M \gtrsim (dN/k)^2$ and $kN \lesssim \nu^2$, all terms in (10) are $O(k/N)$. The additional condition $\nu^2 \ll dN$, with $N/d \to \infty$, makes $\min\{1, d^2/\nu^2\}/(d/N) \to \infty$. Equivalently, the privacy part of the window is

$$\frac{k \log(1/\delta)}{N} \lesssim \epsilon^2 \ll \frac{d \log(1/\delta)}{N}.$$

In its non-saturated portion $\nu \ge d$,

$$\frac{E_{\mathrm{HN}}(k)}{E^{\star}_{\mathrm{SGD}}} \lesssim \Lambda \frac{k\nu^2}{Nd^2}.$$

This ratio tends to zero whenever the displayed right-hand side does. For $k = d$, if $d/\sqrt{M} \lesssim d/N + d^2/\nu^2$ and $\nu \ge d$, the HN upper bound matches the DP-SGD lower-bound rate up to Λ. For $\nu \le d$, the following gives a direct separation.

Corollary 13. *Assume all conditions of Theorems 6 and 7, with the HN parameters chosen for k_* above and with the respective fixed calibration constants. Let Λ be as in* (9)*. For any $\tau > 0$, if $1 \le \nu \le d$ and*

$$A_{\mathrm{opt}} \Lambda (d/\sqrt{M} + N^{-a} + \nu^{-b}) \le \tau,$$

then $E_{\mathrm{HN}}(k_*) \le \tau$ while $E^{\star}_{\mathrm{SGD}} \ge c_{\mathrm{S}} > 0$, where c_{S} is the uniform constant from the lower bound.

Proof. The upper bound is (11); the lower bound follows from $\min\{1, d^2/\nu^2\} = 1$ in Theorem 7. For example, sufficient conditions, evaluated with the same Λ, are

$$N \ge \max\{2d, (3A_{\mathrm{opt}}\Lambda/\tau)^{1/a}\}, \qquad M \ge (3A_{\mathrm{opt}}\Lambda d/\tau)^2,$$
$$\max\{1, (3A_{\mathrm{opt}}\Lambda/\tau)^{1/b}\} \le \nu \le d, \qquad 0 < \epsilon = \nu\sqrt{\log(1/\delta)}/N \le 1.$$

The remaining theorem conditions, including $\delta \le \mathrm{e}^{-1}$, are retained. Along any sequence satisfying them with $1 \le \nu \le d$ and $\Lambda(d/\sqrt{M} + N^{-a} + \nu^{-b}) \to 0$, the separation is $o(1)$ versus $\Omega(1)$. □

C Lower Bound of DP-SGD

We prove Theorem 7 directly for the clipped updates, for every fixed $C > 0$, $\eta \in (0, 1)$, and $T \ge 1$. Write $s^2 = AC^2T \log(1/\delta)/\epsilon^2$, where $A \ge 16$ is a fixed proportionality constant.

Proof. We first establish the divergence bound for the actual Gaussian updates.

Gaussian privacy accounting Recall the full-batch update

$$\omega_t = \omega_{t-1} - \frac{\eta}{N}\left[\sum_{i=1}^{N} \mathrm{clip}_C((\omega_{t-1}^{\top} x_i - y_i)x_i) + \xi_t\right], \qquad \xi_t \sim \mathcal{N}(0, s^2 I_d), \qquad \omega_0 = 0.$$

For replacement-neighbor datasets D, D', condition on the same history $\omega_0, \ldots, \omega_{t-1}$. Only one term in the gradient sum changes, so its sensitivity is at most $2C$. The conditional mean shift of the next iterate is at most $2\eta C/N$, and its noise covariance is $(\eta s/N)^2 I_d$. For $\lambda > 1$, the equal-covariance Gaussian divergence is $D_\lambda(\mathcal{N}(\mu, \sigma^2 I) \| \mathcal{N}(\mu', \sigma^2 I)) = \lambda \|\mu - \mu'\|^2/(2\sigma^2)$. Thus the per-step divergence is at most $2\lambda C^2/s^2$. Adaptive composition and post-processing give, for the final-output kernels,

$$D_\lambda(K_D \| K_{D'}) \le \lambda\rho, \qquad \rho := \frac{2TC^2}{s^2}. \tag{12}$$

Here the conditional bound holds for each common history, so it composes for the full transcript even though the gradients are adaptive; the final iterate is its post-processing. These Gaussian, composition, and conversion identities are from Mironov (2017, Propositions 1, 3, 7). In particular,

$$\chi^2(K_D\|K_{D'}) = \exp(D_2(K_D\|K_{D'})) - 1 \le \mathrm{e}^{2\rho} - 1 =: h. \tag{13}$$

To verify the stated DP guarantee, write $\ell_\delta = \log(1/\delta)$. Conversion from Rényi DP gives privacy parameter $\lambda\rho + \ell_\delta/(\lambda-1)$. Choosing $\lambda = 1 + \sqrt{\ell_\delta/\rho}$ yields $\rho + 2\sqrt{\rho\ell_\delta}$. Under the stated noise calibration, $\rho = 2\epsilon^2/(A\ell_\delta)$, and

$$\rho + 2\sqrt{\rho\ell_\delta} \le \left(\tfrac{1}{8} + \tfrac{1}{\sqrt{2}}\right)\epsilon < \epsilon$$

for $A \ge 16$, $\epsilon \le 1$, $\ell_\delta \ge 1$. The same calculation applies to the one-shot embedding release in Theorem 6, with sensitivity $2C/N$ and variance $s^2_{\mathrm{DP}} = A_{\mathrm{H}}C^2\ell_\delta/(\epsilon^2N^2)$. All divergence bounds are obtained from the Gaussian updates themselves.

A bounded estimator and rotation equivariance Set $B_\gamma := \overline{\alpha}\sqrt{1 + (2\gamma-1)^{-1}}$, so that $\|w\| \le B_\gamma$. For analysis, define $\widetilde{\omega}_T := \mathrm{clip}_{B_\gamma}(\omega_T)$. Projection onto this ball gives $\|\omega_T - w\| \ge \|\widetilde{\omega}_T - w\|$; therefore a lower bound for $\widetilde{\omega}_T$ also lower-bounds the actual algorithm. Let $\widetilde{K}_D$ be its kernel. Post-processing preserves (12) and (13).

For any orthogonal O, transform each input x_i to Ox_i, keeping y_i fixed. At $O\omega$, the corresponding gradient is

$$((O\omega)^\top Ox_i - y_i)Ox_i = O((\omega^\top x_i - y_i)x_i).$$

Also $\mathrm{clip}_C(Ov) = O\mathrm{clip}_C(v)$ and $O\xi_t \overset{d}{=} \xi_t$. Induction from the origin proves equivariance of every iterate and of the radially clipped estimator. Under fixed parameter w, rotating the design produces the data distribution with parameter Ow. Write, for $w \ne 0$,

$$r := \|w\|, \qquad R(w) := \mathbb{E}_w\|\widetilde{\omega}_T - w\|^2, \qquad m(w) := \mathbb{E}_w\widetilde{\omega}_T.$$

The expectation $\mathbb{E}_w$ averages over data and DP noise with w fixed. Invariance under orthogonal transformations fixing w, followed by equivariance between vectors of the same norm, implies $m(w) = a(r)w$. Jensen's inequality then gives

$$|1 - a(r)|r \le \sqrt{R(w)}. \tag{14}$$

This considers the likelihood as a function of fixed $w \in \mathbb{R}^d$; the original prior on w is retained when averaging the bound below.

The tangential score identity Let

$$P_w := I - ww^\top/r^2, \qquad S_w(D) := \sum_{i=1}^N (y_i - w^\top x_i)x_i = \sum_i z_i x_i.$$

Differentiating the Gaussian likelihood, rather than the algorithm, gives

$$Dm(w) = \mathbb{E}_w[\widetilde{\omega}_T S_w(D)^\top].$$

The interchange with expectation is justified by the bounded estimator and $\|x_i\| = \sqrt{d}$: on any compact set of w, the likelihood and its first derivatives are dominated, uniformly in x, by an integrable Gaussian tail times a polynomial in y. The output kernel given D does not depend on w. In particular m is continuously differentiable. For directions perpendicular to w, differentiation of $a(r)w$ gives $a(r)$ times that direction, so $\mathrm{tr}(P_wDm(w)) = (d-1)a(r)$. Define

$$A_i := z_i\langle\widetilde{\omega}_T - w, P_wx_i\rangle.$$

Since $P_ww = 0$, taking the trace of the score identity yields

$$\sum_i \mathbb{E}_wA_i = (d-1)a(r). \tag{15}$$

Independent replacement and a correlation bound Replace record i of D by an independent record with the same distribution given w, obtaining $D^{(i)}$. Denote the corresponding bounded output by $\widetilde{\omega}_T^{(i)}$, and set $A_i' := z_i\langle\widetilde{\omega}_T^{(i)} - w, P_w x_i\rangle$, using the original x_i, z_i. Given w, this replacement output is independent of that original record. Because z_i has mean zero and variance one and $\mathbb{E}[x_i x_i^\top] = I$,

$$\mathbb{E}_w A_i' = 0, \qquad \mathbb{E}_w (A_i')^2 = \mathbb{E}_w \|P_w(\widetilde{\omega}_T^{(i)} - w)\|^2 \le R(w). \tag{16}$$

To apply the divergence bound, let μ be the common distribution of the base variables $(D, \text{replacement record})$ given w. Construct joint laws $\mathbb{P}_i$ and $\mathbb{Q}_i$ by attaching to that same base the kernels $\widetilde{K}_D$ and $\widetilde{K}_{D^{(i)}}$, respectively. As every pair is replacement-adjacent, disintegration gives

$$\chi^2(\mathbb{P}_i \| \mathbb{Q}_i) = \mathbb{E}_\mu \chi^2(\widetilde{K}_D \| \widetilde{K}_{D^{(i)}}) \le h.$$

For the function $f_i(D, v) := z_i\langle v - w, P_w x_i\rangle$ and likelihood ratio $\ell_i = \mathrm{d}\mathbb{P}_i / \mathrm{d}\mathbb{Q}_i$, Cauchy–Schwarz and (16) imply

$$\begin{aligned} |\mathbb{E}_w A_i| &= |\mathbb{E}_{\mathbb{Q}_i}[(\ell_i - 1) f_i]| \\ &\le \sqrt{\mathbb{E}_{\mathbb{Q}_i}(\ell_i - 1)^2 \, \mathbb{E}_{\mathbb{Q}_i} f_i^2} \le \sqrt{h R(w)}. \end{aligned}$$

Combining with (15) gives $(d-1)a(r) \le N\sqrt{hR(w)}$. Independence was used for the replacement output, not for the actual output.

Pointwise privacy and statistical bounds The last inequality and (14) imply

$$1 - \frac{\sqrt{R(w)}}{r} \le \frac{N\sqrt{h}}{d-1}\sqrt{R(w)}.$$

Solving for $R(w)$, without any sign assumption on $a(r)$, gives

$$R(w) \ge \left(\frac{1}{r} + \frac{N\sqrt{h}}{d-1}\right)^{-2} \ge \frac{1}{4}\min\left\{r^2, \frac{(d-1)^2}{N^2 h}\right\}. \tag{17}$$

For a separate statistical bound, use (15) directly:

$$\begin{aligned} (d-1)a(r) &= \mathbb{E}_w\langle\widetilde{\omega}_T - w, P_w S_w(D)\rangle \\ &\le \sqrt{R(w)\,\mathbb{E}_w\|P_w S_w(D)\|^2} = \sqrt{R(w) N (d-1)}. \end{aligned}$$

The last equality follows from independence of the mean-zero residuals and isotropy of x_i. Combining with (14) in the same way yields

$$R(w) \ge \frac{1}{4}\min\left\{r^2, \frac{d-1}{N}\right\}. \tag{18}$$

Averaging over the original prior Under the original uniform-box prior, $w_1 \sim \mathrm{Unif}[-\alpha_1, \alpha_1]$ and $\alpha_1 \ge \underline{\alpha}$. Thus

$$\mathbb{P}(\|w\| \ge \underline{\alpha}/2) \ge \mathbb{P}(|w_1| \ge \underline{\alpha}/2) \ge \tfrac{1}{2}.$$

Average (17) and (18) on this event. Since $d \ge 3$ and $N \ge 2d$, and since $\max\{u, v\} \ge (u+v)/2$,

$$\mathbb{E}\|\omega_T - w\|^2 \ge \mathbb{E}R(w) \gtrsim \frac{d}{N} + \min\left\{1, \frac{d^2}{N^2 h}\right\}.$$

Finally, the calibration gives

$$h = \exp\left(\frac{4\epsilon^2}{A\ell_\delta}\right) - 1 \le \frac{4\mathrm{e}^{1/4}}{A}\frac{\epsilon^2}{\ell_\delta},$$

which proves the required $d/N + \min\{1, d^2/\nu^2\}$ lower bound. All constants depend only on $\underline{\alpha}, \overline{\alpha}, \gamma$ and A, and are independent of $C, \eta, T, d, N, \epsilon, \delta$. This argument includes $T = 1$ and every fixed choice of hyperparameters made before observing the private task, so taking their infimum preserves the bound. □

D AUXILIARY LEMMAS

The following lemma is about the concentration of eigenvalues of the empirical covariance matrix.

Lemma 14. *Let $N \in \mathbb{Z}_{>0}$, and $x_1, \dots, x_N \sim \mathrm{Unif}(\sqrt{d}\mathbb{S}^{d-1})$ be i.i.d. random variables. We define $\widehat{\Sigma} := \frac{1}{N}\sum_{n=1}^{N} x_n x_n^\top$, and let $\lambda_1, \dots, \lambda_d$ be the eigenvalues of $\widehat{\Sigma}$. Suppose that $N > d$. Then, we have the following two facts:*

- *For any $\delta \in (0,1)$, it holds*

$$\sup_{i=1,\dots,d} |\lambda_i - 1| \le A_1 \left\{ \left(\sqrt{\frac{d}{N}} + \sqrt{\frac{\log(2/\delta)}{N}} \right) + \left(\sqrt{\frac{d}{N}} + \sqrt{\frac{\log(2/\delta)}{N}} \right)^2 \right\}.$$

  with probability $1-\delta$, where A_1 is an absolute constant.

- *There exists a constant A_2 such that*

$$\mathbb{P}(\lambda_{\min}(\widehat{\Sigma}) \le t) \le 2\Big(K_{N,d}\sqrt{t}\Big)^{N-d+1},$$

  for all $t \in (0, K_{N,d}^{-2}]$ where $K_{N,d} := A_2 \frac{\sqrt{N}}{\sqrt{d}} \frac{\sqrt{d}+\sqrt{2\log(2N)}}{\sqrt{N}-\sqrt{d}}$.

Proof. We first prove the first argument. Let $A \in \mathbb{R}^{N\times d}$ be the matrix whose n-th row is $x_n^\top$. Then $\widehat{\Sigma} = A^\top A/N$. The random vector $x_n \sim \mathrm{Unif}(\sqrt{d}\mathbb{S}^{d-1})$ is mean zero and isotropic: by rotational invariance, $\mathbb{E}[x_n x_n^\top] = aI_d$ for some $a > 0$, and taking traces gives $ad = \mathbb{E}\|x_n\|^2 = d$, hence $a = 1$. Moreover, x_n is sub-Gaussian with an absolute sub-Gaussian norm. Indeed, for every $u \in \mathbb{S}^{d-1}$, $\langle x_n, u\rangle$ has the same distribution as $\sqrt{d}\,\theta_1$, where $\theta \sim \mathrm{Unif}(\mathbb{S}^{d-1})$; the standard concentration inequality on the sphere gives

$$\mathbb{P}(|\langle x_n, u\rangle| > t) \le 2\exp\big(-ct^2\big), \quad t \ge 0,$$

for an absolute constant $c > 0$. Thus $\sup_{u\in\mathbb{S}^{d-1}} \|\langle x_n, u\rangle\|_{\psi_2}$ is bounded by an absolute constant.

Theorem 4.6.1 of Vershynin (2018), applied to the matrix A with independent, mean-zero, isotropic, sub-Gaussian rows, implies that for every $t \ge 0$, with probability at least $1 - 2\exp\big(-t^2\big)$,

$$\left\|\widehat{\Sigma} - I_d\right\|_{\mathrm{op}} = \left\|\frac{A^\top A}{N} - I_d\right\|_{\mathrm{op}} \le C\left\{ \frac{\sqrt{d}+t}{\sqrt{N}} + \left(\frac{\sqrt{d}+t}{\sqrt{N}}\right)^2 \right\},$$

where $C > 0$ is an absolute constant. Taking $t = \sqrt{\log(2/\delta)}$ yields the bound with probability at least $1-\delta$. Finally, since $\widehat{\Sigma} - I_d$ is symmetric,

$$\sup_{i=1,\dots,d} |\lambda_i - 1| = \left\|\widehat{\Sigma} - I_d\right\|_{\mathrm{op}},$$

which proves the first argument after absorbing constants into A_1.

We prove the second argument. For $d = 1$, $\widehat{\Sigma} = 1$ deterministically, so the claim is immediate since $K_{N,1} > 1$ with the choice $A_2 = \mathrm{e}$ below. Assume $d \ge 2$ henceforth. Let $u_n := x_n/\sqrt{d}$. Then, we have $u_n \sim \mathrm{Unif}(\mathbb{S}^{d-1})$. Let $r_1, \dots, r_N$ be independent random variables, independent of $u_1, \dots, u_N$, whose common distribution is the norm of a standard Gaussian vector in $\mathbb{R}^d$. By the polar decomposition of a Gaussian vector, $g_n := r_n u_n$ are i.i.d. $\mathcal{N}(0, I_d)$. Let $U := [u_1, \dots, u_N]$ and $G := [g_1, \dots, g_N] = UD$ with $D = \mathrm{diag}(r_1, \dots, r_N)$. Moreover, we have $X = \sqrt{d}U$ and $\widehat{\Sigma} = \frac{d}{N}UU^\top$, which yields

$$\lambda_{\min}(\widehat{\Sigma}) = \frac{d}{N} s_{\min}(U)^2,$$

where $s_{\min}$ is the smallest singular value.

Let $M := \max_{n=1,\dots,N} r_n$. Then, we have $\mathbb{P}(r_n \geq \sqrt{d} + t) \leq \mathrm{e}^{-t^2/2}$. Therefore, we have

$$\mathbb{P}(M \leq \sqrt{d} + \sqrt{2\log(2N)}) \geq \frac{1}{2}.$$

On the event $\mathcal{E} := \{M \leq \sqrt{d} + \sqrt{2\log(2N)}\}$, we have

$$s_{\min}(G) = s_{\min}(UD) \leq \|D\|_{\mathrm{op}} s_{\min}(U) \leq (\sqrt{d} + \sqrt{2\log(2N)}) s_{\min}(U).$$

Therefore, we have

$$\{s_{\min}(U) \leq u\} \cap \mathcal{E} \subset \{s_{\min}(G) \leq (\sqrt{d} + \sqrt{2\log(2N)})u\}.$$

Since $\{s_{\min}(U) \leq u\}$ depends only on the directions $u_1, \dots, u_N$, whereas $\mathcal{E}$ depends only on the independent radii $r_1, \dots, r_N$, these two events are independent. This implies

$$\begin{aligned}\mathbb{P}(s_{\min}(U) \leq u) &\leq \frac{\mathbb{P}(s_{\min}(G) \leq (\sqrt{d} + \sqrt{2\log(2N)})u)}{\mathbb{P}(\mathcal{E})} \\ &\leq 2\mathbb{P}(s_{\min}(G) \leq (\sqrt{d} + \sqrt{2\log(2N)})u).\end{aligned}$$

For the $d \times N$ standard Gaussian matrix G, Halko et al. (2011, Proposition A.3 of the arXiv version) gives the following bound for every $\epsilon > 0$:

$$\mathbb{P}(s_{\min}(G) \leq \epsilon(\sqrt{N} - \sqrt{d})) \leq (A_2\epsilon)^{N-d+1}.$$

Indeed, for $q := N - d + 1$, the bound for $\left\|G^\dagger\right\|_{\mathrm{op}} = s_{\min}(G)^{-1}$ gives

$$\mathbb{P}(s_{\min}(G) \leq \epsilon(\sqrt{N} - \sqrt{d})) \leq \frac{1}{\sqrt{2\pi q}} \left[\frac{\mathrm{e}\sqrt{N}(\sqrt{N} - \sqrt{d})}{q}\,\epsilon\right]^q \leq (\mathrm{e}\epsilon)^q,$$

Here $\sqrt{N}(\sqrt{N} - \sqrt{d}) \leq q$, so $A_2 = \mathrm{e}$ suffices.

Therefore, we have

$$\mathbb{P}(s_{\min}(U) \leq u) \leq 2\left(A_2 \frac{(\sqrt{d} + \sqrt{2\log(2N)})u}{\sqrt{N} - \sqrt{d}}\right)^{N-d+1}.$$

Hence, we have

$$\mathbb{P}(\lambda_{\min}(\widehat{\Sigma}) \leq t) = \mathbb{P}\left(s_{\min}(U) \leq \sqrt{\frac{Nt}{d}}\right) \leq 2(K_{N,d}\sqrt{t})^{N-d+1},$$

which completes the proof. □

E More Details on Experimental Setups

Computational resources We conducted the main experiments on a cluster of eight NVIDIA A100 GPUs, each with 40 GB of VRAM.

License We use the pretrained model `imagenet64_uncond_100M_1500K.pt`, which is provided under the MIT License.

Setups for baselines For all baseline methods, LoRA parameters are inserted only into attention layers, and the LoRA rank is set to 1. We used a cosine learning-rate scheduler with a linear warmup over the first 10% of training steps. The batch size was set to 32.

DP-AdamW uses $(\beta_1, \beta_2) = (0.9, 0.999)$, optimizer epsilon 10^{-8}, and weight decay 0.01. Projected DP-SGD constructs an orthonormal basis from per-example LoRA gradients at the initialized model on public examples, using the leading right-singular directions of the uncentered gradient matrix. A public pilot selects rank 32 from $\{8, 32, 128\}$, after which the basis and rank are fixed. Each private

per-example gradient is projected into these 32 coordinates, clipped, aggregated, perturbed with isotropic Gaussian noise, and mapped back to the LoRA parameter space.

For PDA-DPMD, we set the coefficient of the public-data-based mirror map to $\alpha = 0.5$. We use the first-order approximation described in Section 5.1 of Amid et al. (2022), which linearizes the public loss. For DP-SGD with public-data finetuning, we first finetuned the pretrained model on all 30,000 public examples for 100 epochs using Adam, a learning rate of 10^{-2}, a batch size of 32, and a cosine learning-rate scheduler with a linear warmup over the first 10% of training steps. In this public finetuning stage, LoRA parameters were inserted only into attention layers, and the LoRA rank was set to 8. The public rank-8 LoRA is merged into the diffusion checkpoint, after which a fresh rank-1 LoRA is initialized for private DP-SGD.

Public hyperparameter selection For each target ϵ, DP-SGD and PublicFT + DP-SGD select the learning rate and update count by mean FID on public task IDs 0–7, with the target DP noise. The candidates are learning rates $\{10^2, 10^3\}$ and update counts $\{100, 400, 1600\}$, with $C = 0.01$.

DP-AdamW, Projected DP-SGD, and PDA-DPMD use a staged public refinement. Starting from $C = 0.01$, we record unclipped per-example gradient norms on public task IDs 0–7 and form the 50th, 80th, and 95th percentile clipping candidates C_{50}, C_{80}, C_{95}. At C_{80}, we compare the learning-rate grids in Table 2 and update counts $\{25, 100, 400, 1600\}$; the best three learning-rate/update-count pairs are then evaluated at C_{50} and C_{95}. If a grid endpoint wins, the listed external value is evaluated once; the analogous external update counts are 10 and 3200. The best three configurations are evaluated on public task IDs 8–23. If the best two are within one paired standard error in FID, public task IDs 24–31 are also evaluated. All configurations are fixed before evaluation on the 100 private tasks.

Table 2: Public-selection candidate grids. External learning rates are evaluated only when the corresponding grid endpoint wins.

Method	Initial learning rates	External learning rates
DP-SGD / PublicFT + DP-SGD	$10^2, 10^3$	—
DP-AdamW	$0.003, 0.01, 0.03$	$0.001, 0.1$
Projected DP-SGD	$10^2, 10^3, 10^4$	$10, 10^5$
PDA-DPMD	$10, 10^2, 10^3$	$1, 10^4$

Privacy accounting All reported experiments use $\delta = 10^{-5}$, with one image as a record and task size $N = 128$. For DP-DeepSets, numerical RDP accounting calibrates the single Gaussian release. For iterative baselines, Poisson sampling includes each record independently with rate $q = 32/128 = 0.25$, and the gradient sum is divided by the expected batch size 32. Opacus RDP accounting composes the subsampled Gaussian mechanism over the selected number of optimizer steps and calibrates the noise multiplier separately for each candidate.

Setups for DP-DeepSets For DP-DeepSets, we use the same LoRA configuration as the baselines: LoRA parameters are inserted only into attention layers, and the LoRA rank is set to 1. This results in 40,704 LoRA parameters in total. We split these parameters into 128-dimensional tokens so that each token contains only parameters from the same position, resulting in 318 tokens. We set the dimension of the clipped and perturbed representations to $D = 512$. A post-noise MLP maps the noisy dataset embedding to a 128-dimensional context token for the Transformer. The hypernetwork uses a Transformer with hidden dimension 128, depth 4, MLP width 512, and 4 attention heads. To prevent the generated LoRA parameters from having excessively large norms, the linear head is initialized from a normal distribution with standard deviation 10^{-2}, rather than using the PyTorch default initialization. We set the clipping scale for obtaining the noisy dataset embedding to $C = 1$. We selected this value based on the distribution of dataset-embedding norms observed during a preliminary hypernetwork training run on public data without DP noise. We train the hypernetwork for 200 epochs using 30,000 public datasets, each of which corresponds to a k-NN set of a CIFAR-10 data point with $k = 128$. For training, we use a learning rate of 10^{-3}, a batch size of 4, and a cosine learning-rate scheduler with a linear warmup over the first 10% of training steps. A separate

hypernetwork is trained for each target ϵ, using the corresponding noise scale during public training and private inference.

Setups under distribution shift For the experiment in Table 1, the private datasets are the same CIFAR-10 datasets as above, and the public datasets are constructed from ImageNet64. We downsample the 64×64 images to 32×32 images, matching the resolution of CIFAR-10. As with CIFAR-10, for each data point, we construct a public dataset by taking its k-NN set in the Inception embedding space with $k = 128$, yielding $1{,}281{,}167$ public datasets. Using this collection of public datasets, we train DP-DeepSets for 5 epochs. We use the same Transformer configuration, clipping threshold $C = 1$, and head-initialization standard deviation 10^{-2} as in the main experiment.

Table 3: Completed FID results at the main comparison budgets. Values are mean $\pm$ standard deviation over 100 private tasks; a dash denotes an unavailable budget.

Method	$\epsilon = 1$	4	16	64	256	1024
DP-DeepSets	137.35 ± 24.28	133.41 ± 23.15	133.33 ± 22.80	134.14 ± 22.61	—	—
DP-SGD	426.74 ± 41.14	338.82 ± 48.10	268.24 ± 23.01	251.53 ± 19.62	217.60 ± 31.71	215.34 ± 32.29
Projected DP-SGD	245.53 ± 20.64	242.17 ± 24.07	240.70 ± 25.17	239.80 ± 25.22	—	—
DP-AdamW	251.14 ± 19.77	245.80 ± 20.32	215.86 ± 32.56	201.69 ± 28.78	173.21 ± 25.06	153.48 ± 20.80
Public FT + DP-SGD	428.14 ± 41.07	335.35 ± 49.42	237.68 ± 31.20	230.40 ± 28.86	226.51 ± 34.32	209.65 ± 33.61
PDA-DPMD	251.37 ± 19.60	251.37 ± 19.60	228.32 ± 24.82	219.23 ± 26.53	—	—

Table 4: Additional completed FID points in Figure 2.

ϵ	DP-DeepSets	DP-SGD
0.125	189.24 ± 27.79	—
0.25	150.42 ± 26.70	—
0.5	139.39 ± 24.53	—
2	134.48 ± 24.48	404.48 ± 51.10
8	132.80 ± 22.71	308.38 ± 34.08
32	134.10 ± 23.47	253.80 ± 19.91
128	133.08 ± 22.36	233.85 ± 37.08
512	—	216.47 ± 33.59

F Ablation Studies

We study the dimension D of the clipped per-example representations and the hypernetwork size using the CIFAR-10 setup in Appendix E. All variants use $(\epsilon, \delta) = (1, 10^{-5})$, $C = 1$, a peak learning rate of 10^{-3}, seed 1234, and 200 training epochs. We evaluate the final checkpoints on the same 100 private datasets, generating $1{,}024$ images per dataset.

For the dimension ablation, D-Small and D-Large set D to 256 and $1{,}024$, respectively, while retaining the Base encoder widths and decoder. For the network-size ablation, Net-Small keeps $D = 512$ and reduces the per-item and post-noise MLP hidden widths from 512 to 256, the Transformer hidden dimension from 128 to 64, the number of Transformer layers from 4 to 2, and the feed-forward width from 512 to 256. All configurations use four attention heads and a 128-dimensional decoder context.

Table 5: Ablations of the DP representation dimension D and hypernetwork size at $(\epsilon, \delta) = (1, 10^{-5})$. FID values are mean $\pm$ standard deviation across 100 private datasets; lower is better. Base reuses the main experiment's result.

Setting	D	Network profile	FID
Base	512	Base	137.35 ± 24.28
D-Small	256	Base	136.64 ± 23.94
D-Large	1,024	Base	136.09 ± 24.16
Net-Small	512	Small	147.73 ± 21.59

The FID scores for $D = 256$, 512, and $1{,}024$ differ by at most 1.27, indicating that performance is stable across the tested values of D. The smaller hypernetwork increases FID from 137.35 to 147.73 with D fixed at 512, showing greater sensitivity to network capacity than to D in these experiments.