On the Complexity of Learning Sparse Functions with Statistical and Gradient Queries

Nirmit Joshi
Toyota Technological Institute at Chicago
nirmit@ttic.edu

Theodor Misiakiewicz
Toyota Technological Institute at Chicago
theodor.misiakiewicz@ttic.edu

Nathan Srebro
Toyota Technological Institute at Chicago
nati@ttic.edu

Abstract

The goal of this paper is to investigate the complexity of gradient algorithms when learning sparse functions (juntas). We introduce a type of Statistical Queries (SQ), which we call Differentiable Learning Queries (DLQ), to model gradient queries on a specified loss with respect to an arbitrary model. We provide a tight characterization of the query complexity of DLQ for learning the support of a sparse function over generic product distributions. This complexity crucially depends on the loss function. For the squared loss, DLQ matches the complexity of Correlation Statistical Queries (CSQ)—potentially much worse than SQ. But for other simple loss functions, including the ℓ_1 loss, DLQ always achieves the same complexity as SQ. We also provide evidence that DLQ can indeed capture learning with (stochastic) gradient descent by showing it correctly describes the complexity of learning with a two-layer neural network in the mean field regime and linear scaling.

1 Introduction

In recent years, major efforts have been devoted to understanding which distributions can be learned efficiently using gradient-type algorithms on generic models [Abbe and Sandon, 2020, Allen-Zhu and Li, 2020, Malach et al., 2021, Damian et al., 2022, Abbe et al., 2021b,a, 2022, 2023, Bietti et al., 2023, Glasgow, 2023, Dandi et al., 2023, 2024, Edelman et al., 2024, Kou et al., 2024]. In this paper, we focus on learning sparse functions (i.e. "juntas" [Blum and Langley, 1997]), that is functions that depend only on a small number P out of a much larger set $d \gg P$ of input coordinates. The challenge in this setting is to identify the few relevant coordinates. For some sparse functions, such as noisy parities, learning is believed to require $O(d^P)$ runtime [Kearns, 1998], while others, such as linear functions, are easy to learn in $\tilde{O}(d)$ time. Which functions are easy to learn and which are hard? What is the complexity of learning a specific sparse function? Recent works [Abbe et al., 2022, 2023] unveiled a rich "leap" hierarchical structure and saddle-to-saddle dynamics that drives learning in this setting. The goal of the present paper is to provide a general characterization for the complexity of learning sparse functions that go beyond *(i)* hypercube data and Fourier analysis, and *(ii)* CSQ (see below) and focusing only on the squared loss.

The notion of complexity we consider is the *Statistical Query* (SQ) complexity, which studies learning by measuring expectations up to some worst-case tolerance (see [Kearns, 1998, Bshouty and Feldman, 2002, Reyzin, 2020] and Section 3). Although based on worst-case error (or almost equivalently, additive independent noise) rather than the sampling error encountered in practice, statistical query complexity has been proven to be a useful guideline for studying the complexity of learning. In

particular, gradient computations are a special case of statistical queries. In specific cases which include binary functions or gradients on the squared or cross-entropy loss, gradient queries are equivalent[1] to the restricted class of *Correlation Statistical Queries* (CSQ) which are strictly less powerful than general statistical queries. Lower bounds on the CSQ complexity have thus been seen as corresponding to the complexity of gradient-based learning[2] in these restricted cases. Part of the motivation for this paper is to emphasize that this relationship is limited to very specific loss functions and does not hold more generally. In order to study the complexity of gradient algorithms for general output and loss functions, we introduce a type of statistical query which we call *Differentiable Learning Query* (DLQ). These queries are defined with respect to a specific loss ℓ—denoted by DLQ_ℓ—and are given by gradients on a loss ℓ with respect to an arbitrary model. Specifically, DLQ_ℓ algorithms correspond to SQ algorithms with queries of the type

$$\phi(y, \boldsymbol{x}) = \frac{\partial}{\partial \omega} \ell(f(\boldsymbol{x}, \omega), y)\Big|_{\omega=0}, \quad \text{where} \quad f : \mathcal{X}^d \times \mathbb{R} \to \mathbb{R}. \tag{1}$$

Depending on the loss ℓ and target distributions, learning with DLQ_ℓ can be less, equal, or more powerful than CSQ.

For inputs on the hypercube $\boldsymbol{x} \sim \mathrm{Unif}(\{\pm 1\}^d)$ and CSQ, Abbe et al. [2022, 2023] showed that the complexity of learning sparse functions is sharply captured by a *leap exponent* defined in terms of the non-zero Fourier coefficients of the sparse function. Informally, it states that $\Theta(d^{k_*})$ queries are necessary and sufficient to learn with CSQ, where k_* is the minimum number of coordinates one need to add at once to "leap" between non-zero Fourier coefficients. E.g., consider the following two sparse functions:

$$\begin{aligned} y_1 &= x_{s_*(1)} + x_{s_*(1)}x_{s_*(2)} + x_{s_*(1)}x_{s_*(2)}x_{s_*(3)} + x_{s_*(1)}x_{s_*(2)}x_{s_*(3)}x_{s_*(4)}, \\ y_2 &= x_{s_*(1)}x_{s_*(2)}x_{s_*(3)} + x_{s_*(1)}x_{s_*(2)}x_{s_*(4)} + x_{s_*(1)}x_{s_*(3)}x_{s_*(4)} + x_{s_*(2)}x_{s_*(3)}x_{s_*(4)}. \end{aligned} \tag{2}$$

Both functions depend on an (unknown) subset of $P = 4$ coordinates $\{s_*(1), s_*(2), s_*(3), s_*(4)\}$. For y_1, the monomials are ordered such that we add only one new coordinate to the support at a time: the leap exponent is 1 and $\Theta(d)$ queries are sufficient to learn y_1. For y_2, we need to add three new coordinates at once: the leap exponent is 3 and $\Theta(d^3)$ queries are required to learn y_2.

We extend and generalize this charactarization in several significant ways:

(a) We go beyond binary input space and allow for arbitrary measurable space with product distribution. In particular, the leap complexity arises from the structure of the permutation group rather than a specific functional basis.

(b) We go beyond CSQ and squared loss, and tightly characterize the query complexity for learning sparse functions with SQ and DLQ_ℓ algorithms with any loss ℓ. This complexity is in terms of a leap exponent defined analogously as above, but now over a set system $\mathcal{C}_A \subseteq 2^{[P]}$ of *"detectable"* subsets $S \subseteq [P]$, where "detectability" depends on the type of queries $A \in \{SQ, CSQ, DLQ_\ell\}$. Learning with different loss functions are not necessarily comparable to CSQ or SQ and depend on the sparse function. However, we show that some losses, such as ℓ_1 or exponential loss, are "generic", i.e. $\mathcal{C}_{DLQ_\ell} = \mathcal{C}_{SQ}$ for any sparse function, and always match the SQ-leap exponent. This shows that differentiable learning with these losses is as powerful as learning with SQ for juntas.

(c) Finally, we introduce a *cover exponent*—defined on the same system of detectable sets $\mathcal{C}_A$—which captures learning with $A \in \{SQ, CSQ, DLQ_\ell\}$ when the queries are chosen *non-adaptively*, i.e., without adapting to the responses of previous queries. This can be roughly thought of as learning with a single gradient step, versus many consecutive steps in the adaptive case. The contrast between the two helps understand to what extent learning relies on adaptivity and can help frame other related results—see discussion in Section 8.

We summarize the complexity of learning a sparse function with SQ, CSQ, DLQ_ℓ and adaptive/non-adaptive queries in Table 1. To see some of these notions play out, consider again the two examples in Eq. (2):

$$\begin{aligned} y_1: \quad &\mathsf{Leap}(\mathcal{C}_{CSQ}) = 1, \quad \mathsf{Cover}(\mathcal{C}_{CSQ}) = 4, \quad &&\mathsf{Leap}(\mathcal{C}_{SQ}) = \mathsf{Cover}(\mathcal{C}_{SQ}) = 1, \\ y_2: \quad &\mathsf{Leap}(\mathcal{C}_{CSQ}) = \mathsf{Cover}(\mathcal{C}_{CSQ}) = 3, \quad &&\mathsf{Leap}(\mathcal{C}_{SQ}) = \mathsf{Cover}(\mathcal{C}_{SQ}) = 1. \end{aligned}$$

---

[1]This equivalence holds if the input distribution is known, which is the case we consider here.

[2]In Section 8, we discuss recent work Dandi et al. [2024] that showed that even with the squared loss, multiple gradient evaluations on the same batch can be strictly more powerful than CSQ.

$q/\tau^2 = \Theta(d^{k_*})$	Adaptive	Non-adaptive
SQ	$k_* = \mathsf{Leap}\,(\mathcal{C}_{\mathsf{SQ}})$	$k_* = \mathsf{Cover}(\mathcal{C}_{\mathsf{SQ}})$
CSQ	$k_* = \mathsf{Leap}(\mathcal{C}_{\mathsf{CSQ}})$	$k_* = \mathsf{Cover}(\mathcal{C}_{\mathsf{CSQ}})$
DLQ_ℓ	$k_* = \mathsf{Leap}(\mathcal{C}_{\mathsf{DLQ}_\ell})$	$k_* = \mathsf{Cover}(\mathcal{C}_{\mathsf{DLQ}_\ell})$

Table 1: Complexity of learning a sparse function over d input coordinates with different query types based on Theorem 5.1. DLQ_ℓ is defined in Section 2, Leap and Cover of a set system in Definition 1, and systems $\mathcal{C}_{\mathsf{A}}$ of detectable sets in Definition 2 based on test functions depending on query type A, as specified in Eq. (14).

For y_1, DLQ_ℓ will require $\Theta(d)$ adaptive or $\Theta(d^4)$ non-adaptive queries to learn with squared loss—equivalent to CSQ, while $\Theta(d)$ adaptive/non-adaptive queries suffice with ℓ_1-loss—equivalent to SQ. For y_2, DLQ_ℓ will only require $\Theta(d)$ queries, either adaptive or non-adaptive, with ℓ_1-loss, compared to $\Theta(d^3)$ queries with the squared loss. Proposition 6.2.(c) provides an example where $\mathsf{Leap}(\mathcal{C}_{\mathsf{SQ}}) < \mathsf{Leap}(\mathcal{C}_{\mathsf{DLQ}_\ell}) < \mathsf{Leap}(\mathcal{C}_{\mathsf{CSQ}})$.

DLQ_ℓ algorithms are quite different than (stochastic) gradient descent algorithms. On one hand, DLQ_ℓ allows evaluating arbitrary gradients rather than following a trajectory, and is thus potentially more powerful than (S)GD on generic models[3]. On the other hand, the lower bound on DLQ_ℓ algorithms in Table 1 is against *worst-case* noise—much more pessimistic than sampling noise encountered in SGD or GD-on-the-training-set [Abbe et al., 2021b]. Nevertheless, DLQ_ℓ does provide useful guidance and we expect that it *does* capture important aspects of the computational complexity of GD in a number of settings. To demonstrate this, Section 7 considers learning sparse functions on the hypercube using online SGD on a two-layer neural network. We show that in this setting, the DLQ_ℓ-leap exponent correctly captures learnability in the mean-field regime and linear scaling. Namely, for DLQ_ℓ-leap 1 functions, SGD on loss ℓ learns in $O(d)$-steps, while for leap greater than 1, the dynamics remains stuck in a suboptimal saddle subspace and require $\omega(d)$-steps to escape.

2 Setting

Junta Recovery Problem. A sparse recovery (junta learning) problem with P relevant coordinates is defined in terms of a coordinate input space $\mathcal{X}$, an output space $\mathcal{Y}$[4], a marginal distribution μ_x over $\mathcal{X}$ and a link distribution $\mu_{y|\boldsymbol{z}}$ over $\mathcal{Y}$ given elements of $\mathcal{X}^P$. This specifies a joint distribution $\mu_{y,\boldsymbol{z}}$ over a measurable space $\mathcal{Y} \times \mathcal{X}^P$, where the marginal distribution $\boldsymbol{z} \sim \mu_x^P$ is the product distribution, and $y \mid \boldsymbol{z} \sim \mu_{y|\boldsymbol{z}}$. We further denote μ_y to be the marginal distribution over $\mathcal{Y}$. We further denote the junta problem by a tuple $\mu := (\mu_x, \mu_{y|\boldsymbol{z}})$, where the support size P and the spaces $\mathcal{X}$ and $\mathcal{Y}$ are implicit. Consider some examples,

$$\boldsymbol{z} \sim \mathrm{Unif}(\{\pm 1\}^P), \quad y = h_*(\boldsymbol{z}) + \varepsilon \quad \text{for some} \quad h_* : \{\pm 1\}^P \to \mathbb{R} \quad \text{and noise } \varepsilon; \tag{3}$$

$$\boldsymbol{z} \sim \mathsf{N}(\boldsymbol{0}, \mathbf{I}_P), \quad y \sim \mathrm{Bernoulli}(\mathrm{sigmoid}(h_*(\boldsymbol{z}))) \quad \text{for some target } h_* : \mathbb{R}^P \to \mathbb{R}. \tag{4}$$

For $d \geq P$, we define the junta recovery problem as learning the family of distributions

$$\mathcal{H}_\mu^d := \left\{ \mathcal{D}_{\mu,s}^d : s \in \mathsf{P}([d], P) \right\} \tag{5}$$

where $\mathsf{P}([d], P)$ is the set of non-repeating sequences from $[d]$ of length P, and the distribution $\mathcal{D}_{\mu,s}^d$ is a distribution over $\mathcal{Y} \times \mathcal{X}^d$ such that

$$\boldsymbol{x} \sim \mu_x^d \quad \text{and} \quad y \mid (x_{s(1)}, \ldots, x_{s(P)}) \sim \mu_{y|\boldsymbol{z}}.$$

In words, $\mathcal{H}_\mu^d$ is the set of distributions on $(y, x_1, \ldots, x_d)$ where $\boldsymbol{x}$ follows the product distribution μ_x^d and the output y only depends on an (unknown) sequence of P coordinates. For ease of notation, we will denote $\mathcal{D}_s^d = \mathcal{D}_{\mu,s}^d$ when the junta problem $\mu = (\mu_x, \mu_{y|\boldsymbol{z}})$ is clear from context.

---

[3]Following Abbe et al. [2021b], we can construct an "emulation" differentiable model, i.e., a model $f(\boldsymbol{x}; \boldsymbol{w})$ on which GD emulates DLQ_ℓ. However, this offers limited insights as the model $f(\boldsymbol{x}; \boldsymbol{w})$ is highly non-standard.

[4]Formally, $\mathcal{X}$ and $\mathcal{Y}$ are measurable Polish spaces endowed with sigma algebras $\Sigma_{\mathcal{X}}$ and $\Sigma_{\mathcal{Y}}$ respectively.

Success as Support Recovery. For any sequence $s \in \mathrm{P}([d], P)$, we additionally denote the associated unordered set of its elements by $S := \{s(i) : i \in [P]\}$. We consider learning as "succeeding" if it outputs the correct index set. That is, we say a junta learning algorithm "succeeds" in learning $\mathcal{H}_\mu^d$ if for every $s_* \in \mathrm{P}([d], P)$, it outputs $\hat{S} \subseteq [d]$ such that $\hat{S} = S_* = \{s_*(i) : i \in [P]\}$.

For our purpose, just the recovery of relevant coordinates is the objective as we only care about the complexity as a scaling of d. Once we recover the relevant coordinates S_*, learning the ordered sequence s_* corresponds to a problem of size P, and thus its complexity is independent of d. Note that the precise ordering s_* may not be identifiable if for some other s_*' with $S_* = S_*'$ we have $\mathcal{D}_{s_*}^d = \mathcal{D}_{s_*'}^d$; this is possible when the measure $\mu_{y|z}$ has symmetries with respect to coordinates of z. We further emphasize that even the support S_* may not be identifiable, when y is independent of some coordinates of z or when only using a restricted query access to the distribution—we will return to this issue in Section 5.

Does the learner know the link $\mu_{y|z}$? We always assume that the learner knows μ_x. In our formulation, we also assume that the learner additionally knows the link $\mu_{y|z}$, and the only unknown is s_*. Indeed, our lower bounds hold even for this easier setting. But under mild assumptions, our upper bounds can be obtained with an algorithm that does not require knowing $\mu_{y|z}$ (the complexity still depends on the problem μ)—see discussion in Section 5.

Well Behaved Distributions. For our lower bounds, we will consider junta problems $\mu = (\mu_x, \mu_{y|z})$, which are "well-behaved" in a way that is standard in the hypothesis testing literature. Let $\mu_{y,z}$ be the induced joint distribution on (y, z) where $z \sim \mu_x^P$ and $y \mid z \sim \mu_{y|z}$. For any subset $U \subseteq [P]$, let $\mu_{y,z,U}$ be the marginal distribution of $(y, (z_i)_{i \in U})$ and

$$\mu_{y,z,U}^0 = \mu_{y,z,U} \otimes \mu_{\mathcal{X}}^{P-|U|}, \tag{6}$$

meaning that $(y, z_1, \ldots, z_P) \sim \mu_{y,z,U}^0$ has $(y, (z_i)_{i \in U}) \sim \mu_{y,z,U}$ and $(z_i)_{i \in [P] \setminus U} \overset{\text{i.i.d.}}{\sim} \mu_x$ independently of $(y, (z_i)_{i \in U})$ (we replace z_i for $i \notin U$ with independent draws from the marginal). The marginal distribution of z under $\mu_{y,z,U}^0$ is still μ_x^P.

Assumption 2.1. *For any $U \subseteq [P]$, we have $\mu_{y,z} \ll \mu_{y,z,U}^0$, and the Radon-Nikodym derivative $\mathrm{d}\mu_{y,z}/\mathrm{d}\mu_{y,z,U}^0$ is square integrable w.r.t. $\mu_{y,z,U}^0$, i.e.*

$$\frac{\mathrm{d}\mu_{y,z}}{\mathrm{d}\mu_{y,z,U}^0} \in L^2(\mu_{y,z,U}^0) \quad \textit{for all} \quad U \subseteq [P]. \tag{7}$$

This is a standard and implicit assumption in the hypothesis testing literature whenever a corresponding null distribution is considered (here all $\mu_{y,z,U}^0$, i.e., with label y depending only on a strict subset $U \subsetneq [P]$); more specifically for statistical query lower bounds [Feldman et al., 2017, Damian et al., 2024b], low-degree likelihood ratio [Hopkins, 2018, Kunisky et al., 2019], or contiguity lower bounds [Perry et al., 2018]. It always holds when $\mathcal{X}$ is finitely supported. We further comment on the necessity of this assumption in Appendix A.1.

3 Statistical and Differentiable Learning Queries

We will consider three classes of learning algorithms, all based on the statistical query paradigm, but differing in the type of queries allowed, as captured by a set $\mathcal{Q}$ of allowed queries.

For a number of queries q, tolerance $\tau > 0$ and a set $\mathcal{Q} \subseteq \mathbb{R}^{\mathcal{Y} \times \mathcal{X}^d}$ of measurable functions $\mathcal{Y} \times \mathcal{X}^d \to \mathbb{R}$, a $\mathcal{Q}$-**restricted statistical query algorithm** $\mathcal{A} \in \mathcal{Q}\text{-SQ}(q, \tau)$ for junta learning takes an input distribution $\mathcal{D}$ over $\mathcal{Y} \times \mathcal{X}^d$ and operates in q rounds where at each round $t \in \{1, \ldots, q\}$, it issues a query $\phi_t \in \mathcal{Q}$, and receives a response v_t such that

$$|v_t - \mathbb{E}_\mathcal{D}[\phi_t(y, \boldsymbol{x})]| \leq \tau \sqrt{\mathbb{E}_{\mathcal{D}_0}[\phi_t(y, \boldsymbol{x})^2]}, \tag{8}$$

where $\mathcal{D}_0 = \mathcal{D}_y \otimes \mathcal{D}_{\boldsymbol{x}}$ is the associated decoupled "null" distribution where $\boldsymbol{x}$ and y are independent, but follow their marginals[5]. The query ϕ_t can depend on the past responses $v_1, \ldots, v_{t-1}$. After

---

[5]Equivalently, we could restrict the $L_2(\mathcal{D}_0)$ norm of the query. See also Remark 3.1.

issuing q queries, the learner $\mathcal{A}$ outputs $\hat{S} \subseteq [d]$. We say that $\mathcal{A}$ **succeeds in learning** $\mathcal{H}_\mu^d$ if for any $\mathcal{D}_{s_*} \in \mathcal{H}_\mu^d$ and any responses v_t satisfying (8) for $\mathcal{D} = \mathcal{D}_{s_*}$, $\mathcal{A}$ outputs $\hat{S} = S_*$.

Above we allow the queries to be chosen adaptively, i.e., depending on past responses. We also consider $\mathcal{Q}$-**restricted non-adaptive statistical query algorithms**, which we denote by $\mathcal{Q}$-naSQ(q, τ), where the query functions $\{\phi_t\}_{t \in [q]}$ are fixed in advance and do not depend on the past responses. I.e. a non-adaptive algorithm is specified by a list of queries and a mapping from the responses to an output $\hat{S} \subseteq [d]$.

Statistical Queries (SQ): In regular, unrestricted *Statistical Query* learning, the allowed query set, denoted by $\mathcal{Q}_{\mathsf{SQ}}$, is the set of all measurable functions. With slight overloading of notation, we refer to the class of these algorithms simply as SQ(q, τ) and naSQ(q, τ).

Correlation Statistical Queries (CSQ): It is a special subclass of statistical queries, which require $\mathcal{Y} \subseteq \mathbb{R}$ (usually we allow the input and output spaces to be abstract), and are restricted to:

$$\mathcal{Q}_{\mathsf{CSQ}} = \left\{ \phi(y, \boldsymbol{x}) = y \cdot \tilde{\phi}(\boldsymbol{x}) \mid \tilde{\phi} : \boldsymbol{x} \to \mathbb{R} \text{ measurable} \right\}. \tag{9}$$

We denote the class of adaptive and non-adaptive algorithms making such queries as CSQ(q, τ) and naCSQ(q, τ) respectively.

Differentiable Learning Queries (DLQ) with Loss ℓ: Let $\mathcal{F} \subseteq V$, which is an open subset of some normed vector space V, be the output space of our models, e.g. usually $V = \mathcal{F} = \mathbb{R}$ for models with a single output unit, but we may have $V = \mathcal{F} = \mathbb{R}^r$ with multiple output units. We consider a loss function $\ell : \mathcal{F} \times \mathcal{Y} \to \mathbb{R}$ that is locally Lipschitz continuous in its first argument for every $y \in \mathcal{Y}$. The loss is additionally equipped with a derivative operator $\nabla \ell : \mathcal{F} \times \mathcal{Y} \to V$ as a part of the loss definition such that for any $\boldsymbol{u} \in \mathcal{F}$ and $y \in \mathcal{Y}$, we have $\nabla \ell(\boldsymbol{u}, y) \in \partial_1 \ell(\boldsymbol{u}, y)$, the set of generalized Clarke subderivatives of $\ell(\cdot, y)$ at $\boldsymbol{u} \in \mathcal{F}$. This is a standard generalization of derivatives to non-differentiable and non-convex losses; in particular, note that (i) for differentiable losses, $\partial_1 \ell(\boldsymbol{u}, y)$ is a singleton with the true gradient of $\ell(\cdot, y)$ at $\boldsymbol{u} \in \mathcal{F}$, and (ii) for convex losses (in the first argument), $\partial_1 \ell(\boldsymbol{u}, y)$ is the set of subderivatives of $\ell(\cdot, y)$ at $\boldsymbol{u} \in \mathcal{F}$. Finally, let $\mathcal{M} := \{ f : \mathcal{X}^d \times \mathbb{R} \to \mathcal{F} \mid f(\boldsymbol{x}, \omega) \text{ is differentiable at } \omega = 0 \text{ for all } \boldsymbol{x} \in \mathcal{X}^d \}$ be the set of allowed models[6]. Then the allowed differentiable learning query set is:

$$\mathcal{Q}_{\mathsf{DLQ}_\ell} = \left\{ \phi(y, \boldsymbol{x}) = \left[\frac{\mathrm{d}}{\mathrm{d}\omega} f(\boldsymbol{x}, \omega) \right]_{\omega=0}^{\mathsf{T}} \nabla \ell(f(\boldsymbol{x}, 0), y) \,\middle|\, f \in \mathcal{M} \right\}. \tag{10}$$

That is, at each round the algorithm chooses a parametric model $f(\boldsymbol{x}, \omega)$, parameterized by a single scalar ω, and the query corresponds to the derivative (with respect to the single parameter ω) of the loss applied to the model, at $\omega = 0$. This captures the first gradient calculation for a single-parameter model initialized at zero. But the derivative at any other point can also be obtained by querying at a shifted model $f_{(\nu)}(\boldsymbol{x}, \omega) = f(\boldsymbol{x}, \nu + \omega)$, and the gradient with respect to a model $f(\boldsymbol{x}, \boldsymbol{w})$ with r parameters $\boldsymbol{w} \in \mathbb{R}^r$ can be obtained by issuing r queries, one for each coordinate, of the form $f_{(\boldsymbol{w}, i)}(\boldsymbol{x}, \omega) = f(\boldsymbol{x}, \boldsymbol{w} + \omega e_i)$, where e_i is the standard basis vector. Queries of the form $\mathcal{Q}_{\mathsf{DLQ}_\ell}$ can thus be used to implement gradient calculations for any differentiable model, noting that the number of queries q is the number of gradient calculations *times the number of parameters*. Finally, observe that, for differentiable losses, the queries of the form (1) are equivalent to the form mentioned in (10) due to the chain rule.

We denote the class of adaptive and non-adaptive algorithms making such queries as DLQ$_\ell(q, \tau)$:=$\mathcal{Q}_{\mathsf{DLQ}_\ell}$-SQ$(q, \tau)$ and naDLQ$_\ell(q, \tau)$:=$\mathcal{Q}_{\mathsf{DLQ}_\ell}$-naSQ$(q, \tau)$.

Remark 3.1. *More common in the SQ literature is to restrict the query in L^∞ (or equivalently, require precision relative to L^∞). Precision relative to $L^2(\mathcal{D}_0)$ is more similar to VSTAT [Feldman et al., 2017], and is more powerful than relative to L^∞, and our lower bounds hold against this stronger notion. In our algorithms and upper bounds we only need this additional power when*

---

[6] We restrict the model class $\mathcal{M}$ to only contain models that are always differentiable at $\omega = 0$ without loss of generality, as for any non-differentiable model $f_1 : \mathcal{X}^d \times \mathbb{R} \to \mathcal{F}$ under any "reasonable" generalized notion of derivative with $\frac{\mathrm{d}}{\mathrm{d}\omega} f_1(\boldsymbol{x}, \omega) \mid_{\omega=0} = g(\boldsymbol{x}) \in V$, one can define a differentiable model $f_2(\boldsymbol{x}, \omega) = f_1(\boldsymbol{x}, 0) + \omega \cdot g(\boldsymbol{x})$ such that $f_2(\boldsymbol{x}, 0) = f_1(\boldsymbol{x}, 0)$ and $\frac{\mathrm{d}}{\mathrm{d}\omega} f_2(\boldsymbol{x}, \omega) \mid_{\omega=0} = g(\boldsymbol{x})$.

$y \mapsto \nabla \ell(\boldsymbol{u}, y)$ *is unbounded. If we further assume that these functions are bounded, e.g., the labels y are bounded and $\nabla \ell$ continuous, our queries have bounded L^∞ and thus operate in the more familiar SQ setting.*

4 Leap and Cover Complexities

Definition 1. *We define the leap and cover complexities for any system of subsets $\mathcal{C} \subseteq 2^{[P]}$.*

(i) For any system of subsets $\mathcal{C} \subseteq 2^{[P]}$, its leap complexity *is defined as*

$$\mathsf{Leap}(\mathcal{C}) := \min_{\substack{U_1,\ldots,U_r \in \mathcal{C} \\ \bigcup_{i=1}^{r} U_i = [P]}} \max_{i \in [r]} |U_i \setminus \cup_{j=0}^{i-1} U_j|. \tag{11}$$

(ii) We define the cover complexity *of $\mathcal{C}$ to be*

$$\mathsf{Cover}(\mathcal{C}) := \max_{i \in [P]} \min_{U \in \mathcal{C}, i \in U} |U|. \tag{12}$$

Remark 4.1. *We always have $\mathsf{Leap}(\mathcal{C}) \leq \mathsf{Cover}(\mathcal{C})$. Both Leap and Cover complexities are closed under taking the union of subsets in $\mathcal{C}$. Also, when $\mathrm{supp}(\mathcal{C}) = \cup_{U \in \mathcal{C}} U \neq [P]$, then we use the convention $\mathsf{Leap}(\mathcal{C}) = \mathsf{Cover}(\mathcal{C}) = \infty$. See a discussion about this convention in Appendix A.3 and in particular, the definition of the relative leap and cover complexities in Definition 3.*

Here $\mathcal{C}$ is the system of subsets which are "detectable", and will depend on the the query access model. Intuitively, the leap and cover complexities of $\mathcal{C}$ capture the exponent of d in the query complexity when recovering the support of an unknown $s_* \in \mathsf{P}([d], P)$, for adaptive and non-adaptive algorithms respectively. To discover the relevant coordinates of s_*, that correspond to $U \in \mathcal{C}$, one needs to enumerate over $\Theta(d^{|U|})$. Hence, a non-adaptive algorithm, which fixes the queries in advance, requires $\Theta(d^{k_i})$ queries to discover i^{th} relevant coordinate i.e. $s_*(i)$, where $k_i = \min_{i \in U \in \mathcal{C}} |U|$. Therefore, non-adaptive algorithms need a total number of queries that scales as $\Theta(d^{\mathsf{Cover}(\mathcal{C})})$ to learn $\mathrm{supp}(s_*)$. On the other hand, adapting queries using previous answers can greatly reduce this complexity as seen in the example in (2) in Section 1. This is captured by the leap complexity, which measures the maximum number of coordinates we need to discover at once. Finally, the set system of detectable subsets will depend on the type of allowed queries.

Definition 2 (Detectable Subsets). *Let μ be a junta problem. Denote*

$$L_0^2(\mu_x) = \left\{ T \in L^2(\mu_x) : \mathbb{E}_{z \sim \mu_x}[T(z)] = 0 \right\}$$

the set of zero-mean functions. For a set of test function *$\Psi \subseteq L_2(\mu_y)$ we say that $U \subseteq [P]$ is Ψ-detectable iff*

$$\exists T \in \Psi, \ \exists T_i \in L_0^2(\mu_x) \text{ for each } i \in U \quad \text{such that} \quad \mathbb{E}_{\mu_{y,\boldsymbol{z}}}\left[T(y) \prod_{i \in U} T_i(z_i) \right] \neq 0. \tag{13}$$

We denote $\mathcal{C}_\Psi(\mu)$ the set of Ψ-detectable sets, i.e. those sets satisfying (13).

The set of relevant test functions depend on the query types allowed and we define:

$$\begin{array}{lll} \text{For SQ,} & \Psi_{\mathsf{SQ}} = L^2(\mu_y) \text{ (i.e. all } L^2 \text{ functions).} \\ \text{For CSQ (recall } \mathcal{Y} \subseteq \mathbb{R}), & \Psi_{\mathsf{CSQ}} = \{y \mapsto y\} \text{ (just the identity).} \\ \text{For } \mathsf{DLQ}_\ell, & \Psi_{\mathsf{DLQ}_\ell} = \{y \mapsto \boldsymbol{v}^\mathsf{T} \nabla \ell(\boldsymbol{u}, y) : \boldsymbol{u} \in \mathcal{F}, \boldsymbol{v} \in V\}. \end{array} \tag{14}$$

While the queries of the form (13), where Ψ_A for $\mathsf{A} \in \{\mathsf{SQ}, \mathsf{CSQ}, \mathsf{DLQ}_\ell\}$ is given by (14), are less general than $\phi(y, \boldsymbol{x})$ with $\phi \in \mathcal{Q}_\mathsf{A}$, they can be implemented by the corresponding query types $\mathcal{Q}_\mathsf{A}$ and are sufficient for deciding between "$S \subseteq [d]$ maps to the corresponding $U \subseteq [P]$" or "$S \not\subseteq S_*$". The sets $\Psi_{\mathsf{SQ}}, \Psi_{\mathsf{CSQ}}, \Psi_{\mathsf{DLQ}_\ell}$ from (14) used for detectibility arise naturally in the proof of the lower bounds of Theorem 5.1.

To ease notation, for query type A, we use the shorthand $\mathcal{C}_\mathsf{A} := \mathcal{C}_{\Psi_\mathsf{A}}$, and

$$\mathsf{Leap}_\mathsf{A}(\mu) := \mathsf{Leap}(\mathcal{C}_\mathsf{A}(\mu)) = \mathsf{Leap}(\mathcal{C}_{\Psi_\mathsf{A}}(\mu)), \qquad \mathsf{Cover}_\mathsf{A}(\mu) := \mathsf{Cover}(\mathcal{C}_\mathsf{A}(\mu)) = \mathsf{Cover}(\mathcal{C}_{\Psi_\mathsf{A}}(\mu)).$$

We refer to these as the A-*leap exponent* and A-*cover exponent* of the problem μ.

5 Main Result: Characterizing the Complexity of Learning Juntas

Theorem 5.1. *For any junta problem μ and any loss ℓ, there exists $C > c > 0$ (that depend on P,μ and the loss, but not on d), such that for query types $\mathsf{A} \in \{\mathsf{SQ}, \mathsf{CSQ}, \mathsf{DLQ}_\ell\}$ with corresponding test function sets Ψ_A as defined in (14):*

Adaptive. *Let $k_* = \mathsf{Leap}_\mathsf{A}(\mu)$. There exists an algorithm $\mathcal{A} \in \mathsf{A}(q, \tau)$ that succeeds in learning $\mathcal{H}_\mu^d$ with $\tau = c$ and $q = Cd^{k_*}$. And if μ satisfies Assumption 2.1, then for any (q, τ) such that $q/\tau^2 \leq cd^{k_*}$, no algorithm $\mathcal{A} \in \mathsf{A}(q, \tau)$ succeeds at learning $\mathcal{H}_\mu^d$.*

Non-adaptive. *Let $k_* = \mathsf{Cover}_\mathsf{A}(\mu)$. There exists an algorithm $\mathcal{A} \in \mathsf{naA}(q, \tau)$ that succeeds in learning $\mathcal{H}_\mu^d$ with $\tau = c$ and $q = Cd^{k_*}$. And if μ satisfies Assumption 2.1, then for any (q, τ) such that $q/\tau^2 \leq cd^{k_*}$, no algorithm $\mathcal{A} \in \mathsf{naA}(q, \tau)$ succeeds at learning $\mathcal{H}_\mu^d$.*

Remark 5.2. *In the positive results in Theorem 5.1, we used all the allowed complexity to have many $(q = \Theta(d^{k_*}))$ queries, and kept the tolerance constant. More generally, it is possibly to trade off between the number of queries q and tolerance τ, at a cost of a log-factor: For $k_* = \mathsf{Leap}_\mathsf{A}(\mu)$ and $k_* = \mathsf{Cover}_\mathsf{A}(\mu)$, respectively, there exists algorithms $\mathcal{A} \in \mathsf{A}(q, \tau)$ and $\mathcal{A} \in \mathsf{naA}(q, \tau)$ that learn $\mathcal{H}_\mu^d$ for any $q \geq C \log(d)$ and $\tau \leq c$ with $q/\tau^2 \geq Cd^{k_*} \log(d)$.*

The proof of this theorem is deferred to Appendix B. Theorem 5.1 shows that the leap and cover complexities sharply capture the scaling in d of statistical query algorithms when learning $\mathcal{H}_\mu^d$.

Remark 5.3. *The above upper bound uses that μ and therefore the T, T_i in Definition 2 are known. In the case when μ is unknown, one can follow a similar strategy as in Damian et al. [2024b] and randomize the transformations T and T_i over a sufficiently large (but finite independent of d) linear combination of functions in Ψ_SQ and $L_0^2(\mu_x)$. Under some regularity assumption on μ and ℓ, one can show by anti-concentration that with constant probability, the expectation in condition (13) is bounded away from 0 by a constant independent of the dimension.*

6 Relationship Between $\mathsf{SQ}, \mathsf{CSQ}$ and DLQ_ℓ

Obviously, $\mathsf{CSQ} \subseteq \mathsf{SQ}$, and indeed we see that $\mathcal{C}_\mathsf{CSQ} \subseteq \mathcal{C}_\mathsf{SQ}$ in Definition 2 because of which $\mathsf{Leap}_\mathsf{CSQ} \geq \mathsf{Leap}_\mathsf{SQ}$ and $\mathsf{Cover}_\mathsf{CSQ} \geq \mathsf{Cover}_\mathsf{SQ}$. For binary $\mathcal{Y}$, these query models collapse, but otherwise there can be an arbitrary gap.

Proposition 6.1 (SQ versus CSQ)**.** *For any μ, let $\mathcal{C}_\mathsf{SQ} := \mathcal{C}_\mathsf{SQ}(\mu)$, and $\mathcal{C}_\mathsf{CSQ} := \mathcal{C}_\mathsf{CSQ}(\mu)$. If $|\mathcal{Y}| = 2$ (binary output), then we always have $\mathcal{C}_\mathsf{CSQ} = \mathcal{C}_\mathsf{SQ}$ and $\mathsf{Leap}_\mathsf{SQ} = \mathsf{Leap}_\mathsf{CSQ}$ and $\mathsf{Cover}_\mathsf{SQ} = \mathsf{Cover}_\mathsf{CSQ}$. On the other hand if $|\mathcal{Y}| > 2$, the SQ-exponents can be much smaller than the CSQ-exponents: e.g., there exist a setting with $\mathsf{Leap}_\mathsf{SQ} = \mathsf{Cover}_\mathsf{SQ} = 1$ and $\mathsf{Leap}_\mathsf{CSQ} = \mathsf{Cover}_\mathsf{CSQ} = P$.*

Similarly, $\mathsf{DLQ}_\ell \subseteq \mathsf{SQ}$ by definition, and thus, $\mathsf{Leap}_{\mathsf{DLQ}_\ell} \geq \mathsf{Leap}_\mathsf{SQ}$ and $\mathsf{Cover}_{\mathsf{DLQ}_\ell} \geq \mathsf{Cover}_\mathsf{SQ}$.

Proposition 6.2 (DLQ_ℓ versus SQ and CSQ)**.** *Consider any $\mu = (\mu_y, \mu_{y|\boldsymbol{z}})$.*

(a) Let $\mathcal{Y}, \mathcal{F} \subseteq \mathbb{R}$. For the squared loss $\ell : (u, y) \mapsto (u - y)^2$, we always have $\mathcal{C}_{\mathsf{DLQ}_\ell} = \mathcal{C}_\mathsf{CSQ}$, and thus, $\mathsf{Leap}_\mathsf{CSQ} = \mathsf{Leap}_{\mathsf{DLQ}_\ell}$ and $\mathsf{Cover}_\mathsf{CSQ} = \mathsf{Cover}_{\mathsf{DLQ}_\ell}$.

(b) A sufficient condition for $\mathcal{C}_\mathsf{SQ} = \mathcal{C}_{\mathsf{DLQ}_\ell}$ is to have $\mathrm{span}(\Psi_{\mathsf{DLQ}_\ell})$ dense in $L_0^2(\mu_y)$. Conversely, if there exists nonzero $T \in L_0^2(\mu_y)$ bounded with $T(y)$ orthogonal in $L^2(\mu_y)$ to any functions in Ψ_{DLQ_ℓ}, then there exists a problem μ such that $\mathsf{Cover}_{\mathsf{DLQ}_\ell} = \mathsf{Leap}_{\mathsf{DLQ}_\ell} > \mathsf{Cover}_\mathsf{SQ} = \mathsf{Leap}_\mathsf{SQ}$.

(c) There exists a loss ℓ and a junta problem μ such that $\mathsf{Leap}_\mathsf{SQ}(\mu) < \mathsf{Leap}_{\mathsf{DLQ}_\ell}(\mu) < \mathsf{Leap}_\mathsf{CSQ}(\mu)$. Similarly, we can have $\mathsf{Leap}_{\mathsf{DLQ}_\ell} > \mathsf{Leap}_\mathsf{CSQ}$.

The condition in Proposition 6.2.(b) can be seen as a universal approximation property of neural networks with activation $y \mapsto \nabla\ell(\boldsymbol{u}, y)$ [Cybenko, 1989, Hornik, 1991, Sonoda and Murata, 2017]. The next lemma gives a few examples of losses with $\mathsf{DLQ}_\ell = \mathsf{SQ}$. For concreteness, we consider $\mathcal{F} = \mathbb{R}$ and $\mathcal{Y} \subseteq \mathbb{R}$.

Theorem 6.3. *For $\ell \in \{\ell_1 : (u, y) \mapsto |u - y|, \ell_{hinge} : (u, y) \mapsto \max(1 - uy, 0)\}$, then $\mathcal{C}_{\mathsf{DLQ}_\ell}(\mu) = \mathcal{C}_{\mathsf{SQ}}(\mu)$ for all[7] problems μ. If we further assume $\mathcal{Y} \subseteq [-M, M]$ for $M \geq 0$, then $\ell(u, y) = e^{-uy}$ (exponential loss) has $\mathcal{C}_{\mathsf{DLQ}_\ell}(\mu) = \mathcal{C}_{\mathsf{SQ}}(\mu)$ for all problems μ.*

The cases of ℓ_1 and Hinge loss follow directly from universal approximation of neural networks with linear threshold activation. The proofs of the above propositions and lemma can be found in Appendix C. Propositions 6.1 and 6.2 combined with Theorems 6.3 and 5.1 directly imply a number of separation results between adaptive and non-adaptive algorithms and between different loss functions. See examples (2) in the introduction, and further examples in Appendix C.

7 Gradient Descent on Neural Networks

The goal of this section is to connect the complexity of DLQ to gradient descent on standard neural networks. We focus on the simple case of $\boldsymbol{x} \sim \mathrm{Unif}(\{+1, -1\}^d)$ uniformly distributed on the hypercube, and $\mathcal{Y} \subseteq \mathbb{R}$ and $\mathcal{F} = \mathbb{R}$. In this setting, condition (13) in Definition 2 simplifies to: there exists $T \in \Psi_\mathsf{A}$ such that $\mathbb{E}_{\mu_{y,z}}\left[T(y)\prod_{i \in U} z_i\right] = \mathbb{E}_{\mu_{y,z}}\left[T(y)\chi_U(\boldsymbol{z})\right] \neq 0$, where $\chi_U(\boldsymbol{z}) := \prod_{i \in U} z_i$ denote the standard Fourier-Walsh basis. In particular, the set $\mathcal{C}_{\mathsf{CSQ}}$ contains exactly all non-zero Fourier coefficients of $h_*(\boldsymbol{z}) := \mathbb{E}[y|\boldsymbol{z}]$, and we recover the leap exponent of Abbe et al. [2023] as discussed in the introduction.

We train a standard two layer neural network (see (NN1) in Appendix D) of width M, using online SGD with a loss function $\ell : \mathbb{R} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$, i.e. SGD on $\mathbb{E}_{(\boldsymbol{x},y) \sim \mathcal{D}_{s_*}^d}[\ell(f_{\mathsf{NN}}(\boldsymbol{x}; \boldsymbol{\Theta}), y)]$ for $\mathcal{D}_{s_*}^d \in \mathcal{H}_\mu^d$. More specifically, we train the parameters $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_j : j \in [M]\}$ using batch-SGD with loss ℓ and batch size b from initialization $(\boldsymbol{\theta}_j)_{j \in [M]} \overset{\text{i.i.d.}}{\sim} \rho_0$ specified in (69). At each step, given samples $(\{(\boldsymbol{x}_{ki}, y_{ki}) : i \in [b]\})_{k \geq 0}$, the weights are updated using

$$\boldsymbol{\theta}_j^{k+1} = \boldsymbol{\theta}_j^k - \frac{\eta}{b}\left(\sum_{i \in [b]} \ell'(f_{\mathsf{NN}}(\boldsymbol{x}_{ki}; \boldsymbol{\Theta}^t), y_{ki})\nabla_{\boldsymbol{\theta}}\sigma_*(\boldsymbol{x}_{ki}; \boldsymbol{\theta}_j^k) + \lambda\,\boldsymbol{\theta}_j^k\right), \qquad (\ell\text{-bSGD})$$

where η is the step-size and we allow for a ℓ_2 regularization with parameter $\lambda \in \mathbb{R}_+$. Recall that $\ell'(u, y) \in \partial_1 \ell(u, y)$ is the defined derivative of $\ell(\cdot, y)$ at $u \in \mathbb{R}$. We define the test error

$$\mathcal{R}(f) = \mathbb{E}_{\mathcal{D}_{s_*}^d}\left[\ell(f(\boldsymbol{x}), y)\right], \qquad (15)$$

and further introduce the excess test error $\overline{\mathcal{R}}(f) = \mathcal{R}(f) - \inf_{\bar{f}:\{\pm 1\}^d \to \mathbb{R}} \mathcal{R}(\bar{f})$.

Dimension-free dynamics. In the junta learning setting, when y only depends on $P \lll d$ coordinates, following Abbe et al. [2022, Secion 3], the SGD dynamics (ℓ-bSGD) concentrates on an effective *dimension-free* (DF) dynamics as $M, d \to \infty$ and $\eta \to 0$. This equivalence holds under a certain assumption on the loss function, and other assumptions on the initialization and activation that are similar to the setup of Abbe et al. [2022] (see Appendix D for details, especially DF-PDE).

Dimension-free dynamics' alignment with the support. In the above limiting regime, $\mathsf{Leap}_{\mathsf{DLQ}_\ell}$ crisply characterizes (DF-PDE) dynamics' alignment with the support.

Theorem 7.1 (Informal version of Theorem D.6). *If $\mathsf{Leap}_{\mathsf{DLQ}_\ell} = 1$ then for some time t, the output of the model at time t of (DF-PDE) dynamics depends on all coordinates z_i. On the other hand, if $\mathsf{Leap}_{\mathsf{DLQ}_\ell} > 1$, then there exists a coordinate i such that for any time $t \geq 0$, the output of the model at time t of (DF-PDE) dynamics does no depend on z_i.*

This establishes that the ability of (DF-PDE) dynamics (comparable to (ℓ-bSGD) in the linear scaling) to learn all relevant coordinates depends on $\mathsf{Leap}_{\mathsf{DLQ}_\ell}(\mu) = 1$. That is if $\mathsf{Leap}_{\mathsf{DLQ}_\ell} > 1$, then (DF-PDE) remains stuck in a suboptimal saddle subspace. On the other hand, if $\mathsf{Leap}_{\mathsf{DLQ}_\ell} = 1$, then (DF-PDE) dynamics escapes this subspace and the weights align with the entire support.

---

[7] For Hinge loss, we further restrict μ_y to measures with finite second moment so that $\Psi_\mathsf{A} \subseteq L^2(\mu_y)$.

Learning of $\mathsf{Leap}_{\mathsf{DLQ}_\ell} = 1$ **with finite SGD.** Showing directly that (DF-PDE) dynamics indeed reach a near global minimizers of the test error remains challenging. Alternatively, we show that a specific layer-wise training dynamics similar to Abbe et al. [2022] achieves a vanishing excess error for $\mathsf{Leap}_{\mathsf{DLQ}_\ell}(\mu) = 1$ settings in the linear scaling of samples.

Roughly speaking, we train the first layer for $k_1 = P$ steps and then the second layer weights for $k_2 = O_d(1)$ steps using batch-SGD with batch size $b = O_d(d)$, both for a loss ℓ. We consider a polynomial activation $\sigma(x) = (1+x)^L$ of degree $L \geq 2^{8P}$. The most notable difference from Abbe et al. [2022] is that we further slightly perturb step-sizes for each coordinate $\eta^{w_i} = \eta\kappa_i$ with $\kappa_i \in [1/2, 3/2]$, and denote $\boldsymbol{\kappa} = (\kappa_1, \ldots, \kappa_d) \in \mathbb{R}^d$ for the first layer training. This perturbation is necessary to break possible coordinate symmetries; see Remark 7.3.

Theorem 7.2 (Informal version of Theorem E.1). *For a convex and analytic loss ℓ, almost surely over the perturbation $\boldsymbol{\kappa}$ and the initial bias $\bar{c} \in \mathbb{R}$ the following holds. If $\mathsf{Leap}_{\mathsf{DLQ}_\ell}(\mu) = 1$, then the above layer-wise SGD training dynamics with total sample size $n = \Theta_d(d)$ and $M = \Theta_d(1)$ achieves excess test error $\overline{\mathcal{R}}(f_{\mathsf{NN}}(\cdot; \Theta^{k_1+k_2})) = o(1)$ with high probability.*

The formal statement and the precise training specifications can be found in Appendix D. This result generalizes Abbe et al. [2022, Theorem 9] beyond squared loss.

Remark 7.3. *A slight coordinate-wise perturbation in the step-sizes for the first layer training is necessary to break the potential coordinate symmetries in the output y—see discussion in Abbe et al. [2022, Appendix A]. This can be removed by either stating the theorem for all but some measure zero set of Leap-1 functions as in Abbe et al. [2022], or by studying the dynamics for $O(\log d)$ steps.*

The query complexity of b-batch SGD on M neurons for Υ SGD-steps is $T_{\mathsf{C}} = \Theta(bM\Upsilon d)$. The above theorems show that for $\mathsf{Leap}_{\mathsf{DLQ}_\ell}(\mu) = 1$, $\Upsilon = \Theta(d/b)$ steps with $M = \Theta(1)$ neurons—and therefore $T_{\mathsf{C}} = \Theta(d^2)$—suffices to learn the support and minimize the excess test error. Furthermore, for $\mathsf{Leap}_{\mathsf{DLQ}_\ell}(\mu) > 1$ and neural networks trained in the mean-field regime, $\Upsilon = \Theta(d)$ (and therefore $T_{\mathsf{C}} = \Theta(d^2)$) is not enough. We further comment on the general conjectural picture in Appendix A.2.

Numerical Simulation. We consider a function similar to y_2 in (2) with $\mathsf{Leap}_{\mathsf{CSQ}} = 3$ but $\mathsf{Leap}_{\mathsf{SQ}} = 1$. Specifically, we set $P = 4$ and $\mathcal{C} = \{\{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\}\}$, and define $y = h_*(\boldsymbol{z})$ where

$$h_*(\boldsymbol{z}) = \sum_{U \in \mathcal{C}} \hat{h}_*(U)\chi_U(\boldsymbol{z}), \text{ where } \hat{h}_*(U) \sim \mathrm{Unif}([-2, 2]) \text{ for all } U \in \mathcal{C}. \tag{16}$$

We train with online 1-batch SGD (ℓ-bSGD) on a two-layer net with different loss functions (without any regularization) and stepsize $\eta \propto 1/d$, where we consider ambient dimensions $d \in \{100, 300, 500\}$. In Figure 1, we plot the test mean squared error versus $\eta \times$ SGD-iterations (thus also scaled with $1/d$), over 10 trials. Additionally, we also plot the continuous time (DF-PDE) in (dashed black line) that corresponds to the limit $d \to \infty$.

(a) $\ell(u, y) = (u - y)^2$ (b) $\ell(u, y) = |u - y|$ (c) $\ell(u, y) = (u - y)^2 + |u - y|^3$
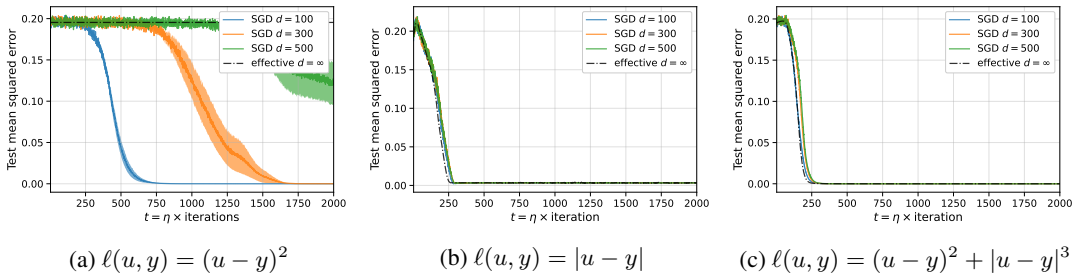
Figure 1: The function $h_*(\boldsymbol{z})$ in (16) has $\mathsf{Leap}_{\mathsf{CSQ}} = 3$ but $\mathsf{Leap}_{\mathsf{SQ}} = 1$. For the squared loss (left plot), (DF-PDE) remains stuck at initialization (no learning), and to escape the saddle, SGD requires a number of iterations that increases faster than $O(d)$. For the absolute loss (center plot) or the other loss (right plot), we have $\mathsf{Leap}_{\mathsf{DLQ}_\ell} = \mathsf{Leap}_{\mathsf{SQ}} = 1$, and the SGD dynamics learns in $\Theta(d)$ steps and (DF-PDE) learns in $O(1)$ continuous time.

8 Conclusion and Outlook

In this paper, we considered learning juntas over general product distributions with statistical query algorithms. To capture learning with gradient evaluations over a general loss and arbitrary model, we introduced *Differentiable Learning Queries* (DLQ_ℓ), which can be seen as a generalization of correlation statistical queries beyond squared loss. We then showed that the complexity of learning juntas with either SQ, CSQ, or DLQ_ℓ algorithms is sharply characterized in terms of a *leap exponent* (adaptive queries) or a *cover exponent* (non-adaptive queries). These exponents are defined in terms of a minimal combination of detectable sets to cover the support, where the system of detectable sets depends on the allowed queries. In general, the leap and cover exponents for different losses are not comparable. However, we identify "generic" losses, including ℓ_1, where DLQ_ℓ algorithms are as powerful as SQ algorithms for learning juntas. We further showed that DLQ_ℓ can indeed capture the complexity of learning with SGD in the case of data on the hypercube.

Worst-case v.s. One-pass v.s. Multi-pass SGD. DLQ_ℓ (like SQ) is defined in terms of worst-case noise. It is well understood that worst-case noise is theoretically very different from estimating population gradients based on samples (either independent samples as in one-pass (S)GD, or with repeated use of samples as in full-batch or multi-pass) when highly specialized models are allowed [Abbe et al., 2021b]. However, we expect DLQ_ℓ to still capture the complexity of one-pass SGD in many settings of interest—with "regular" models—such as in Section 7. With sample reuse (e.g. multi-pass) the situation is more complex: Dandi et al. [2024] showed that two steps of full-batch gradient descent with square loss goes beyond CSQ. Heuristically, and in light of our work, two steps on the same batch can be seen as a single gradient evaluation on a modified loss, thus going beyond $\mathsf{CSQ} = \mathsf{DLQ}_{\ell_{sq}}$, but remaining inside $\mathsf{DLQ}_{\tilde{\ell}} \subseteq \mathsf{SQ}$ for some perturbed non-quadratic loss $\tilde{\ell}$. Indeed, we expect generally that multi-pass SGD on a "regular" model will remain in SQ.

Multi-index and beyond. In this paper, we focused on learning sparse functions. We hope the modular nature of our analysis framework (defining detectable sets in terms of test functions, and leap and cover complexities of set systems), our definition of DLQ, and the distinctions we emphasize between CSQ, SQ and DLQ and between adaptive and non-adaptive complexities, will be helpful in guiding and contextualizing analysis in other settings such as learning single-index or multi-index functions [e.g. Refinetti et al., 2021, Mousavi-Hosseini et al., 2022, Abbe and Boix-Adsera, 2022, Damian et al., 2022, Bietti et al., 2022, 2023, Damian et al., 2024a]. For example, the information exponent for single-index [Arous et al., 2021] can be seen as analogous to our CSQ-cover exponent, the generative exponent for single-index [Damian et al., 2024b] as analogous to our SQ-cover exponent, and the isoLeap exponent for multi-index [Abbe et al., 2023, Dandi et al., 2023, Bietti et al., 2023] as analogous to our CSQ-leap exponent. It would be interesting to obtain a unified understanding of these specialized treatments and extend our general framework to multi-index models, and also to learning under other invariances beyond permutations and rotations.

In our setup, we emphasize generic input and output spaces, without a field structure. This emphasizes that when learning juntas, polynomials or degree of input or output coordinates is irrelevant. Defining multi-index models and introducing rotational invariance necessitates a field structure and gives rise to the relevance of polynomial degrees and decomposition.

An important point is that when considering permutation (as in juntas) vs. rotational (as in multi-index models) invariance, one must consider not only the invariance structure of the target function class, but also the input distribution (i.i.d. coordinates as in our case, or more generally exchangeable vs. spherical) and learning rule. E.g., learning a parity over input coordinates requires only $\Theta(\log d)$ samples, but a rotationally equivariant algorithm effectively learns parities also over rotated axis, which requires $\Omega(d)$ samples [Glasgow, 2023] (and thus $\Omega(d^2)$ runtime). This also explains the need for $\Theta(d)$ steps and thus $\Theta(d^2)$ runtime to learn leap-1 functions using SGD on a rotationally invariant neural net in Section 7. In order to break the $\Omega(d)$-sample lower bound, we need to break the rotation-equivariance, e.g. using sparse initialization and ℓ_1 regularization, which indeed can achieve $\Theta(\log(d))$ sample complexity.

References

Emmanuel Abbe and Enric Boix-Adsera. On the non-universality of deep learning: quantifying the cost of symmetry. *Advances in Neural Information Processing Systems*, 35:17188–17201, 2022.

Emmanuel Abbe and Colin Sandon. On the universality of deep learning. *Advances in Neural Information Processing Systems*, 33:20061–20072, 2020.

Emmanuel Abbe, Enric Boix-Adsera, Matthew S Brennan, Guy Bresler, and Dheeraj Nagaraj. The staircase property: How hierarchical structure can guide deep learning. *Advances in Neural Information Processing Systems*, 34:26989–27002, 2021a.

Emmanuel Abbe, Pritish Kamath, Eran Malach, Colin Sandon, and Nathan Srebro. On the power of differentiable learning versus pac and sq learning. *Advances in Neural Information Processing Systems*, 34:24340–24351, 2021b.

Emmanuel Abbe, Enric Boix Adsera, and Theodor Misiakiewicz. The merged-staircase property: a necessary and nearly sufficient condition for sgd learning of sparse functions on two-layer neural networks. In *Conference on Learning Theory*, pages 4782–4887. PMLR, 2022.

Emmanuel Abbe, Enric Boix Adsera, and Theodor Misiakiewicz. Sgd learning on neural networks: leap complexity and saddle-to-saddle dynamics. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 2552–2623. PMLR, 2023.

Zeyuan Allen-Zhu and Yuanzhi Li. Backward feature correction: How deep learning performs deep learning. *arXiv e-prints*, pages arXiv–2001, 2020.

Gerard Ben Arous, Reza Gheissari, and Aukosh Jagannath. Online stochastic gradient descent on non-convex losses from high-dimensional inference. *Journal of Machine Learning Research*, 22 (106):1–51, 2021.

Alberto Bietti, Joan Bruna, Clayton Sanford, and Min Jae Song. Learning single-index models with shallow neural networks. *Advances in Neural Information Processing Systems*, 35:9768–9783, 2022.

Alberto Bietti, Joan Bruna, and Loucas Pillaud-Vivien. On learning gaussian multi-index models with gradient flow. *arXiv preprint arXiv:2310.19793*, 2023.

Avrim L Blum and Pat Langley. Selection of relevant features and examples in machine learning. *Artificial intelligence*, 97(1-2):245–271, 1997.

Nader H Bshouty and Vitaly Feldman. On using extended statistical queries to avoid membership queries. *Journal of Machine Learning Research*, 2(Feb):359–395, 2002.

Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

Michael Celentano, Theodor Misiakiewicz, and Andrea Montanari. Minimum complexity interpolation in random features models. *arXiv preprint arXiv:2103.15996*, 2021.

Lenaic Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. *Advances in neural information processing systems*, 31, 2018.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

Alex Damian, Eshaan Nichani, Rong Ge, and Jason D Lee. Smoothing the landscape boosts the signal for sgd: Optimal sample complexity for learning single index models. *Advances in Neural Information Processing Systems*, 36, 2024a.

Alex Damian, Loucas Pillaud-Vivien, Jason D Lee, and Joan Bruna. The computational complexity of learning gaussian single-index models. *arXiv preprint arXiv:2403.05529*, 2024b.

Alexandru Damian, Jason Lee, and Mahdi Soltanolkotabi. Neural networks can learn representations with gradient descent. In *Conference on Learning Theory*, pages 5413–5452. PMLR, 2022.

Yatin Dandi, Florent Krzakala, Bruno Loureiro, Luca Pesce, and Ludovic Stephan. Learning two-layer neural networks, one (giant) step at a time. *arXiv preprint arXiv:2305.18270*, 2023.

Yatin Dandi, Emanuele Troiani, Luca Arnaboldi, Luca Pesce, Lenka Zdeborová, and Florent Krzakala. The benefits of reusing batches for gradient descent in two-layer networks: Breaking the curse of information and leap exponents. *arXiv preprint arXiv:2402.03220*, 2024.

Richard M Dudley. *Real analysis and probability*. Chapman and Hall/CRC, 2018.

Benjamin Edelman, Surbhi Goel, Sham Kakade, Eran Malach, and Cyril Zhang. Pareto frontiers in deep feature learning: Data, compute, width, and luck. *Advances in Neural Information Processing Systems*, 36, 2024.

Vitaly Feldman, Elena Grigorescu, Lev Reyzin, Santosh S Vempala, and Ying Xiao. Statistical algorithms and a lower bound for detecting planted cliques. *Journal of the ACM (JACM)*, 64(2): 1–37, 2017.

Margalit Glasgow. Sgd finds then tunes features in two-layer neural networks with near-optimal sample complexity: A case study in the xor problem. In *The Twelfth International Conference on Learning Representations*, 2023.

Samuel Hopkins. *Statistical inference and the sum of squares method*. PhD thesis, Cornell University, 2018.

Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2): 251–257, 1991.

Michael Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)*, 45(6):983–1006, 1998.

Yiwen Kou, Zixiang Chen, Quanquan Gu, and Sham M Kakade. Matching the statistical query lower bound for k-sparse parity problems with stochastic gradient descent. *arXiv preprint arXiv:2404.12376*, 2024.

Dmitriy Kunisky, Alexander S Wein, and Afonso S Bandeira. Notes on computational hardness of hypothesis testing: Predictions using the low-degree likelihood ratio. In *ISAAC Congress (International Society for Analysis, its Applications and Computation)*, pages 1–50. Springer, 2019.

Eran Malach, Pritish Kamath, Emmanuel Abbe, and Nathan Srebro. Quantifying the benefit of using differentiable learning over tangent kernels. In *International Conference on Machine Learning*, pages 7379–7389. PMLR, 2021.

Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018.

Alireza Mousavi-Hosseini, Sejun Park, Manuela Girotti, Ioannis Mitliagkas, and Murat A Erdogdu. Neural networks efficiently learn low-dimensional representations with sgd. *arXiv preprint arXiv:2209.14863*, 2022.

DJ Newman and Morton Slater. Waring's problem for the ring of polynomials. *Journal of Number Theory*, 11(4):477–487, 1979.

Amelia Perry, Alexander S Wein, Afonso S Bandeira, and Ankur Moitra. Optimality and sub-optimality of pca i: Spiked random matrix models. *The Annals of Statistics*, 46(5):2416–2451, 2018.

Maria Refinetti, Sebastian Goldt, Florent Krzakala, and Lenka Zdeborová. Classifying high-dimensional gaussian mixtures: Where kernel methods fail and neural networks succeed. In *International Conference on Machine Learning*, pages 8936–8947. PMLR, 2021.

Lev Reyzin. Statistical queries and statistical algorithms: Foundations and applications. *arXiv preprint arXiv:2004.00557*, 2020.

Grant M Rotskoff and Eric Vanden-Eijnden. Neural networks as interacting particle systems: Asymptotic convexity of the loss landscape and universal scaling of the approximation error. *stat*, 1050:22, 2018.

Justin Sirignano and Konstantinos Spiliopoulos. Mean field analysis of neural networks: A central limit theorem. *Stochastic Processes and their Applications*, 130(3):1820–1852, 2020.

Le Song, Santosh Vempala, John Wilmes, and Bo Xie. On the complexity of learning neural networks. *Advances in neural information processing systems*, 30, 2017.

Sho Sonoda and Noboru Murata. Neural network with unbounded activation functions is universal approximator. *Applied and Computational Harmonic Analysis*, 43(2):233–268, 2017.

Gregory Valiant. Finding correlations in subquadratic time, with applications to learning parities and juntas. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 11–20. IEEE, 2012.

Santosh Vempala and John Wilmes. Gradient descent for one-hidden-layer neural networks: Polynomial convergence and sq lower bounds. In *Conference on Learning Theory*, pages 3115–3117. PMLR, 2019.

A   Additional discussions from the main text

A.1   Necessity of Assumption 2.1

We further elaborate on the necessity of Assumption 2.1 in our setting. Informally, we require the label to have "enough noise". This is necessary to obtain meaningful lower-bounds for general SQ algorithms, as noted in Valiant [2012], Song et al. [2017], Vempala and Wilmes [2019]. When learning real-valued target function classes, allowing for general measurable queries is too weak (note that arbitrary measurable functions are not practical anyway). Indeed, Vempala and Wilmes [2019] showed that any finite set $\mathcal{H}$ of (noiseless) functions can be learned with $\log|\mathcal{H}|$ (bounded) queries and constant tolerance[8]. They identify three possible approaches to address this challenge: (i) Require $y \mapsto \phi(y, \boldsymbol{x})$ to be Lipschitz for every fixed $\boldsymbol{x}$; (ii) Insist on noisy concepts, e.g., $y = f(\boldsymbol{x}) + \zeta$, with $\zeta \sim \mathsf{N}(0, \sigma^2)$, which is equivalent to Lipschitz queries; (iii) Restrict the form of the queries, such as in CSQ. Our Assumption 2.1 can be viewed as a quantitative version of approach (ii). For approach (iii), if we restrict ourselves to CSQ or DLQ_ℓ with ℓ piecewise analytic, we can indeed remove Assumption 2.1 by modifying the proof of Theorem 5.1.

A.2   The conjectural picture of complexity of learning (ℓ-bSGD)

For $k_* := \mathsf{Leap}_{\mathsf{DLQ}_\ell}(\mu)$, learning the junta problem with online SGD on loss ℓ requires

$$
\begin{aligned}
k_* = 1: & & n = \Theta(d), & & T_{\mathsf{C}} = \Theta(d^2), \\
k_* = 2: & & n = \Theta(d\log d), & & T_{\mathsf{C}} = \widetilde{\Theta}(d^2), \\
k_* > 2: & & n = \widetilde{\Theta}(d^{k_*-1}), & & T_{\mathsf{C}} = \widetilde{\Theta}(d^{k_*}).
\end{aligned}
$$

Partial evidence was provided towards this conjecture for the square loss in Abbe et al. [2023], Bietti et al. [2023]: they showed that learning in this setting happens through a saddle-to-saddle dynamics, where each time the neural network needs to align to $k > 1$ new coordinates, a saddle appears and SGD requires $\widetilde{\Theta}(d^{k-1})$ steps to escape.

(a) $\ell(u,y) = (u-y)^2$ (b) $\ell(u,y) = |u-y|$ (c) $\ell(u,y) = (u-y)^2 + |u-y|^3$
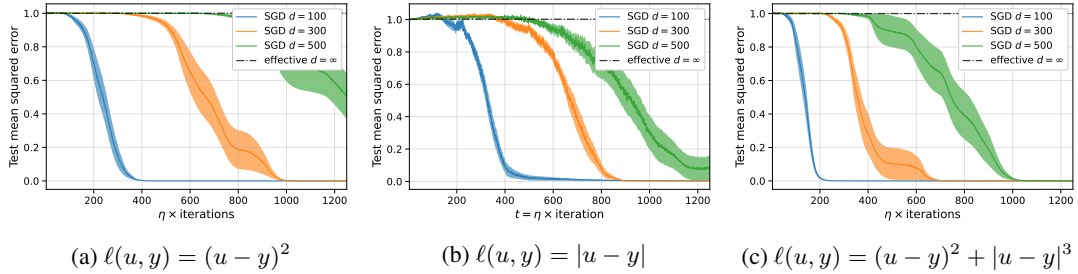
Figure 2: We repeat the same experiment, with the step size $\eta \propto 1/d$ as in Figure 1 but now considering the order 3 parity function $h_*(\boldsymbol{z}) = z_1 z_2 z_3$. Since $\mathcal{Y} = \{-1, +1\}$, by Proposition 6.1, we have $\mathsf{Leap}_{\mathsf{SQ}} = \mathsf{Leap}_{\mathsf{CSQ}} = \mathsf{Leap}_{\mathsf{DLQ}_\ell} = 3$ for all the three loss functions considered. We observe that online SGD learns in $O(d^2)$ iterations for all the losses.

A.3   Identifiability and Relative Leap and Cover Complexities

The Leap and Cover complexity may be infinite when $\mathrm{supp}(\mathcal{C}_{\mathsf{A}}) = \cup_{U \in \mathcal{C}_{\mathsf{A}}} U \subsetneq [P]$, in which case Theorem 5.1 (correctly) implies that we cannot recover the relevant coordinates that correspond to $[P] \setminus \mathrm{supp}(\mathcal{C}_{\mathsf{A}})$ using queries of type A (even with perfect precision). In this case, we can characterize the complexity of recovering $\mathrm{supp}(\mathcal{C}_{\mathsf{A}})$ instead, and this is captured by the following "relative" complexities:

---

[8]The result holds under a "non-degeneracy" condition, namely, for any $f, g \in \mathcal{H}$, $f \neq g$ almost surely. Consider for example $y = x_{s_*(1)}$ with $\boldsymbol{x} \sim \mathsf{N}(\boldsymbol{0}, \mathbf{I}_d)$: using the construction in Vempala and Wilmes [2019], we can learn $\mathcal{H}_\mu^d$ with $q/\tau^2 = \Theta(\log(d))$ queries instead of $q/\tau^2 = \Theta(d)$ as in Theorem 5.1.

Definition 3. *For any system of subsets $\mathcal{C} \subseteq 2^{[P]}$ the relative leap and cover complexities are:*

$$\mathsf{relLeap}(\mathcal{C}) := \min_{\substack{U_1,\ldots,U_r \in \mathcal{C} \\ \bigcup_{i=1}^{r} U_i = \mathrm{supp}(\mathcal{C})}} \max_{i \in [r]} |U_i \setminus \cup_{j=0}^{i-1} U_j|, \tag{17}$$

$$\mathsf{relCover}(\mathcal{C}) := \max_{i \in \mathrm{supp}(\mathcal{C})} \min_{U \in \mathcal{C}, i \in U} |U|. \tag{18}$$

A slight variant of Theorem 5.1 can then be shown, where $\mathsf{relLeap}_\mathsf{A}(\mu)$ and $\mathsf{relCover}_\mathsf{A}(\mu)$ for $\mathsf{A} \in \{\mathsf{SQ}, \mathsf{CSQ}, \mathsf{DLQ}_\ell\}$ characterizes the complexity of recovering $\mathrm{supp}(\mathcal{C}_\mathsf{A})$.

For SQ, we may have $\mathrm{supp}(\mathcal{C}_\mathsf{SQ}) \subsetneq [P]$ only when y doesn't actually depend on some of the coordinates in $[P]$ (i.e. $y \mid \boldsymbol{z} = y|(\boldsymbol{z}_i)_{i \in \mathrm{supp}(\mathcal{C})}$). In this case, once we recover $\mathrm{supp}(\mathcal{C})$ (and possibly enumerating over permutations of its coordinates) we can also recover the conditional $y|\boldsymbol{x}$. That is, $\mathsf{relLeap}_\mathsf{SQ}$ and $\mathsf{relCover}_\mathsf{SQ}$ characterize the complexity of learning the distribution $\mathcal{D}_{s_*}$, even if not the (unidentifiable) set S_*.

For DLQ_ℓ, including CSQ, we may not be able to identify some coordinates even if y does depend on them. Consider for example the junta problem of size $P = 2$ where $y|\boldsymbol{z} = z_1 + \mathcal{N}(0, z_2^2)$. Although y does depend on the second coordinate z_2, it is not identifiable using CSQ queries, $\mathrm{supp}(\mathcal{C}_\mathsf{CSQ}) = \{1\}$, and using CSQ queries we cannot recover the conditional distribution $\mathcal{D}_{s_*}$ (even if we know the link $\mu_{y|\boldsymbol{z}}$). What we *can* say for DLQ_ℓ after recovering $\hat{S} = \mathrm{supp}(\mathcal{C}_{\mathsf{DLQ}_\ell})$, is that although we might not know $\mathcal{D}_{s_*}$, we can find the ℓ-risk minimizer $f_* = \arg\min_{f:\mathcal{X}\to\mathcal{F}} \mathbb{E}_{(y,\boldsymbol{x})\sim\mathcal{D}_{s_*}} [\ell(f(x), y)]$. That is, $\mathsf{relLeap}_{\mathsf{DLQ}_\ell}$ and $\mathsf{relCover}_{\mathsf{DLQ}_\ell}$ characterize the complexity of ℓ-risk minimization.

A.4 Property of SQ Leap and Cover Complexities

We show a useful property of the SQ-exponent. This characterization is similar to the definition of the generative exponent for single-index models on Gaussian data in [Damian et al., 2024b, Definition 2.4], while the detectable set Definition 2 is similar to their variational representation [Damian et al., 2024b, Proposition 2.6].

Proposition A.1 (Property of SQ-detectable sets). *$U \in \mathcal{C}_\mathsf{SQ}$ if and only if there exists $T_i \in L_0^2(\mu_x)$ for all $i \in U$ such that $\|\xi_U\|_{L^2(\mu_y)} > 0$ where*

$$\xi_U(y) = \mathbb{E}_{\mu_{y,\boldsymbol{z}}}\Big[\prod_{i \in U} T_i(z_i) \Big| y \Big].$$

Proof of Proposition A.1. In the forward direction, consider any $U \in \mathcal{C}_\mathsf{SQ}$, then by Definition 2, there exists $T_i \in L_0^2(\mu_x)$ and $T \in L^2(\mu_y)$ such that

$$0 \neq \mathbb{E}_{\mu_{y,\boldsymbol{z}}}\left[T(y) \prod_{i \in U} T_i(z_i) \right] = \mathbb{E}_{\mu_y}\left[T(y)\xi_U(y) \right],$$

where we defined $\xi_U(y) := \mathbb{E}_{\mu_{y,\boldsymbol{z}}}\left[\prod_{i \in U} T_i(z_i) \mid y \right]$. We then have

$$0 < |\mathbb{E}_{\mu_y}\left[T(y)\xi_U(y) \right]| \leq \|T\|_{L^2(\mu_y)} \|\xi_U\|_{L^2(\mu_y)}, \text{ by Cauchy-Schwarz inequality.}$$

Therefore, we conclude that $\|\xi_U\|_{L^2(\mu_y)} > 0$ as desired. In the opposite direction, let us consider $U \subseteq [P]$ such that there exists $T_i \in L_0^2(\mu_x)$ for every $i \in U$ such that $\|\xi_U\|_{L^2(\mu_y)} > 0$, where

$$\xi_U(y) = \mathbb{E}_{\mu_{y,\boldsymbol{z}}}\left[\prod_{i \in U} T_i(z_i) \mid y \right].$$

Then let $T(y) := \xi_U(y)$. It is straightforward to verify that

$$\|T\|_{\mu_y}^2 = \mathbb{E}_{\mu_y}\left[T(y)^2 \right] = \mathbb{E}_{\mu_y}\left[\mathbb{E}_{\mu_{y,\boldsymbol{z}}}\left[\prod_{i \in U} T_i(z_i) \mid y \right]^2 \right] \leq \mathbb{E}_{\mu_{y,\boldsymbol{z}}}\left[\prod_{i \in U} T_i(z_i)^2 \right] < \infty.$$

15

We verified $T(y) \in L^2(\mu_y)$. Moreover,

$$\mathbb{E}_{\mu_y}\left[T(y)\prod_{i\in U}T_i(z_i)\right] = \mathbb{E}_{\mu_y}\left[T(y)\mathbb{E}_{\mu_{y,z}}\left[\prod_{i\in U}T_i(z_i)\mid y\right]\right]$$

$$= \mathbb{E}_{\mu_y}\left[\xi_U(y)\xi_U(y)\right] = \|\xi_U\|^2_{L^2(\mu_y)} \neq 0.$$

Therefore, by Definition 2, we obtain $U \in \mathcal{C}_{\mathsf{SQ}}$. $\qquad\square$

B   Proof of Theorem 5.1

B.1   Preliminaries

Tensor basis of $L^2(\mathcal{X}^k, \mu_x^k)$**:**   Since we assumed that $(\mathcal{X}, \mu_x)$ is a Polish probability space, $L^2(\mathcal{X}, \mu_x)$ is separable [Dudley, 2018]. Consider $\{\psi_i\}_{i\in\mathcal{I}}$, $\mathcal{I} \subseteq \mathbb{N}$, an orthonormal basis of $L^2(\mathcal{X}, \mu_x)$ such that $\psi_0 = 1$ without loss of generality. In particular, the space $L^2(\mathcal{X}^k, \mu_x^k)$ admits the following tensor basis

$$\{\psi_{\boldsymbol{i}} := \psi_{i_1} \otimes \psi_{i_2} \otimes \ldots \otimes \psi_{i_k} \ : \ \boldsymbol{i} = (i_1, \ldots, i_k) \in \mathcal{I}^k\}. \tag{19}$$

We will denote $\operatorname{supp}(\boldsymbol{i}) = \{j \in [d] : i_j > 0\}$.

Distributions ν_S^σ**:**   For any non-repeating sequence $\sigma \in \mathsf{P}([d], P)$ of length P, a subset $U \subseteq [P]$, and a system of subsets $\mathcal{C} = \{U_1, \ldots, U_m\} \subseteq 2^{[P]}$, we denote $\sigma(U) := \{\sigma(i) : i \in U\}$ and $\sigma(\mathcal{C}) := \{\sigma(U_1), \ldots, \sigma(U_m)\}$. The following analysis holds for fix μ and hence we will often omit the subscript μ by denoting $\mathcal{D}_{\mu,\sigma}^d = \mathcal{D}_\sigma^d$ for any $\sigma \in \mathsf{P}([d], P)$. For $\sigma = \mathrm{Id}$, i.e. $\sigma(1) = 1, \ldots, \sigma(P) = P$, we will further omit writing Id and denote $\mathcal{D}^d = \mathcal{D}_{\mathrm{Id}}^d$. For $U \subseteq [P]$, $\mu_{y,z,U}^0 \in \mathcal{P}(\mathcal{Y} \times \mathcal{X}^P)$ is the distribution of (y_U, z) corresponding to decoupling y from $(z_i)_{i\notin U}$ in $\mu_{y,z}$ following Eq. (6), i.e., $(y_U, (z_i)_{i\in U}) \sim \mu_{y,z,U}$ and $(z_i)_{i\notin U} \sim \mu_x^{P-|U|}$ independently.

For clarity, for every $U \subseteq [P]$ and $\sigma \in \mathsf{P}([d], P)$,

$$\nu_U^\sigma := [\mathrm{Id} \otimes \sigma]_\# [\mu_{y,z,U}^0 \otimes \mu_x^{d-P}], \tag{20}$$

where we abuse the notation and view σ as a permutation in Π_d, by extending the sequence $\sigma \in \mathsf{P}([d], P)$ to an ordering of irrelevant coordinates, as notice that it yields the same ν_U^σ. In words, ν_U^σ is the distribution of $(y, \boldsymbol{x})$ where we decoupled y from $(x_{\sigma(i)})_{i\notin U}$ in $\mathcal{D}_\sigma^d$. For $\sigma = \mathrm{Id}$, we denote $\nu_U = \nu_U^{\mathrm{Id}}$. For $U = [P]$, we simply denote $\nu^\sigma := \mathcal{D}_\sigma^d$. Note that $\nu_0 := \nu_\emptyset = \mu_y \otimes \mu_x^d$ and $\nu_U^\sigma = \nu_U$ if $\sigma(i) = i$ for all $i \in U$.

Basic properties under Assumption 2.1.   Recall that our lower bounds in Theorem 5.1 are under Assumption 2.1. It states that the junta problem $\mu = (\mu_x, \mu_{y|z})$ is such that Eq. (7) holds. An immediate consequence of this assumption, using Jensen's inequality is that

$$\frac{\mathrm{d}\mu_{y,z,U}^0}{\mathrm{d}\mu_y \otimes \mu_x^P} \in L^2(\mu_y \otimes \mu_x^P). \tag{21}$$

Indeed, this follows from

$$\left\|\frac{\mathrm{d}\mu_{y,z,U}^0}{\mathrm{d}\mu_y \otimes \mu_x^P}\right\|^2_{L^2(\mu_y \otimes \mu_x^P)} \leq \left\|\frac{\mathrm{d}\mu_{y,z}}{\mathrm{d}\mu_y \otimes \mu_x^P}\right\|^2_{L^2(\mu_y \otimes \mu_x^P)} < \infty,$$

where the first inequality follows from Jensen's inequality and the second by (7).

Another consequence of this assumption is that, for any $U \subseteq [P]$ and $\sigma \in \mathsf{P}([d], P)$, we have $L^2(\nu_0) \subseteq L^1(\nu_U^\sigma)$. This is because for any $\phi \in L^2(\nu_0)$

$$\|\phi\|_{L^1(\nu_U^\sigma)} = \mathbb{E}_{(y,\boldsymbol{x})\sim\nu_U^\sigma}\left[|\phi(y, \boldsymbol{x})|\right] = \mathbb{E}_{(y,\boldsymbol{x})\sim\nu_0}\left[\frac{\mathrm{d}\nu_U^\sigma}{\mathrm{d}\nu_0}(y, \boldsymbol{x}) \cdot |\phi(y, \boldsymbol{x})|\right] \leq \left\|\frac{\mathrm{d}\nu_U^\sigma}{\mathrm{d}\nu_0}\right\|_{L^2(\nu_0)} \|\phi\|_{L^2(\nu_0)} < \infty,$$

where we used Cauchy-Schwarz inequality and Eq. (21).

Algorithms in A: Throughout the proof, we consider $\mathsf{A} \in \{\mathsf{SQ}, \mathsf{CSQ}, \mathsf{DLQ}_\ell\}$ and $\mathcal{C}_\mathsf{A} := \mathcal{C}_\mathsf{A}(\mu)$ defined in Definition 2. First, observe that one can alternatively define $\mathcal{Q}_\mathsf{SQ} = L^2(\nu_0)$ because for any measurable $\phi \notin L^2(\nu_0)$, the received query response can infinite by (8) and the learner does not gain any information. Therefore, it suffices to prove lower bound against the query set $\mathcal{Q}_\mathsf{SQ} = L^2(\nu_0)$. For the same reason, consider the allowed query sets, $\mathcal{Q}_\mathsf{CSQ} \subseteq L^2(\nu_0)$ containing queries of the form $\phi(y, \boldsymbol{x}) = y\tilde{\phi}(\boldsymbol{x})$, and $\mathcal{Q}_{\mathsf{DLQ}_\ell} \subseteq L^2(\nu_0)$ with queries of the form

$$\phi(y, \boldsymbol{x}) = \left[\frac{\mathrm{d}}{\mathrm{d}\omega} f(\boldsymbol{x}, \omega) \right]_{\omega=0}^{\mathsf{T}} \nabla \ell(f(\boldsymbol{x}, 0), y). \tag{22}$$

We claim that, for $\boldsymbol{i} \in \mathcal{I}^d$ and $\psi_{\boldsymbol{i}}$ defined in Eq. (19) and any $\phi \in \mathcal{Q}_\mathsf{A}$, Definition 2 implies that if $\mathrm{supp}(\boldsymbol{i}) \notin \sigma(\mathcal{C}_\mathsf{A})$, then almost surely over $\tilde{\boldsymbol{x}} \sim \mu_x^d$

$$\mathbb{E}_{(y,\boldsymbol{x}) \sim \nu^\sigma} [\phi(y, \tilde{\boldsymbol{x}}) \psi_{\boldsymbol{i}}(\boldsymbol{x})] = 0. \tag{23}$$

This follows from Definition 2 of Ψ-detectability with Eq. (14), and using the fact that $\psi_{\boldsymbol{i}}$ is the product of $\psi_{i_j}(x_j)$ with $\psi_{i_j} \in L_0^2(\mu_x)$ for $j \in \mathrm{supp}(\boldsymbol{i})$. More specifically, since $\phi \in L^2(\nu_0)$, the function $\phi(\cdot, \tilde{\boldsymbol{x}}) \in L^2(\mu_y)$ almost surely over $\tilde{\boldsymbol{x}} \sim \mu_x^d$, and In particular, almost surely over $\tilde{\boldsymbol{x}} \sim \mu_x^d$:

> For any $\phi \in \mathcal{Q}_\mathsf{SQ}$,   we have $\phi(\cdot, \tilde{\boldsymbol{x}}) \in L^2(\mu_y) = \Psi_\mathsf{SQ}$,
>
> For any $\phi \in \mathcal{Q}_\mathsf{CSQ}$,   we have $\phi(\cdot, \tilde{\boldsymbol{x}}) = y\tilde{\phi}(\tilde{\boldsymbol{x}})$ (i.e. a scaled identity),
>
> For any $\phi \in \mathcal{Q}_{\mathsf{DLQ}_\ell}$,   we have $\phi(\cdot, \tilde{\boldsymbol{x}}) \in \Psi_{\mathsf{DLQ}_\ell}$ (i.e. $\phi(\cdot, \tilde{\boldsymbol{x}})$ of the form $\boldsymbol{v}^\mathsf{T} \nabla \ell(\boldsymbol{u}, y)$).

B.2   Theorem 5.1.(a): lower bound for adaptive queries

Definition of set $\mathcal{S}_*$: Let $k_* = \mathsf{Leap}(\mathcal{C}_\mathsf{A})$. Define $\mathcal{C}_* \subset \mathcal{C}_\mathsf{A}$ to be the maximal subset of $\mathcal{C}_\mathsf{A}$ such that $\mathsf{Leap}(\mathcal{C}_*) \leq k_* - 1$, and $\mathcal{S}_* = \cup_{U \in \mathcal{C}_*} U$. In words, $\mathcal{S}_*$ is the collection of all coordinates in the support that can be reached by adding at most $k_* - 1$ coordinates at a time. In particular, by definition of $\mathcal{C}_*$, we have $|U \setminus \mathcal{S}_*| \geq k_*$ for every $U \in \mathcal{C}_\mathsf{A} \setminus \mathcal{C}_*$. Denote $r = |\mathcal{S}_*|$ and without loss of generality, assume that $\mathcal{S}_* = \{1, \ldots, r\}$. We decompose the covariate into $\boldsymbol{x} = (\boldsymbol{x}_{\mathcal{S}_*}, \boldsymbol{x}_{\mathcal{S}_*^c})$ where $\mathcal{S}_*^c = [d] \setminus \mathcal{S}_*$. Using these notations, $(y_{\mathcal{S}_*}^\sigma, \boldsymbol{x}) \sim \nu_{\mathcal{S}_*}^\sigma$ has $(y_{\mathcal{S}_*}^\sigma, \boldsymbol{x}_{\sigma(\mathcal{S}_*)}) \sim \mathcal{D}_{y,\boldsymbol{z},\mathcal{S}_*}$ and $\boldsymbol{x}_{\sigma(\mathcal{S}_*^c)} \sim \mu_x^{d-r}$ independently.

Definition of $\mathcal{H}_{\mu,\mathcal{S}_*}^d$: We introduce $\mathsf{P}([d], P, \mathcal{S}_*) \subseteq \mathsf{P}([d], P)$ the subset of non-repeating sequence of length $[P]$ from elements $[d]$ such that $\sigma(i) = i$ for all $i \in \mathcal{S}_*$, and $\mathcal{H}_{\mu,\mathcal{S}_*}^d \subseteq \mathcal{H}_\mu^d$ the subset of hypotheses $\mathcal{D}_\sigma^d$ with $\sigma \in \mathsf{P}([d], P, \mathcal{S}_*)$. Recall that for all $\sigma \in \mathsf{P}([d], P, \mathcal{S}_*)$, we have $\nu_{\mathcal{S}_*}^\sigma = \nu_{\mathcal{S}_*}$. To prove our lower bound, we will lower bound the complexity of learning $\mathcal{H}_{\mu,\mathcal{S}_*}^d$ which implies a lower bound on $\mathcal{H}_\mu^d$. Specifically, we will show that with $q/\tau^2 \leq cd^{k_*}$ for some constant $c > 0$ that only depends on P, μ (and the loss ℓ for DLQ_ℓ), statistical query algorithms in A cannot distinguish for all $\sigma \in \mathsf{P}([d], P, \mathcal{S}_*)$ between $\nu^\sigma = \mathcal{D}_\sigma^d$ and $\nu_{\mathcal{S}_*}$.

Proof outline: For any $\sigma \in \mathsf{P}([d], P, \mathcal{S}_*)$ and query $\phi \in \mathcal{Q}_\mathsf{A} \subseteq L^2(\nu_0)$, define

$$\Delta_\sigma(\phi) := \mathbb{E}_{\nu^\sigma}[\phi] - \mathbb{E}_{\nu_{\mathcal{S}_*}}[\phi]. \tag{24}$$

We will show below that there exists a constant $C > 0$ that only depends on μ such that

$$\sup_{\phi \in \mathcal{Q}_\mathsf{A}} \frac{\mathbb{E}_\sigma[\Delta_\sigma(\phi)^2]}{\|\phi\|_{L^2(\nu_0)}^2} \leq Cd^{-k_*}, \tag{25}$$

where $\mathbb{E}_\sigma$ denote the expectation with respect to $\sigma \sim \mathrm{Unif}(\mathsf{P}([d], P, \mathcal{S}_*))$. The intuition behind this crucial bound in (25) is that we defined the null distribution as the distribution with matching marginals $(y, \boldsymbol{x})$ where the label only depends on the support of $\boldsymbol{x}$ discoverable with "leap" k_*. Hence, to make any progress, any new detectable set added to the support needs to contain at least k_* new coordinates. In the second moment, to have a non-zero contribution, we need the permutation to map the support to a detectable subset, i.e. the permutation to map to a subset of size at least k_*, which has probability $O(d^{-k_*})$ over permutation.

17

Let $\phi_1, \ldots, \phi_q$ be the sequence of adaptive queries with responses $\mathbb{E}_{\nu_{\mathcal{S}_*}}[\phi_t]$. Then, by Markov's inequality,

$$\mathbb{P}_\sigma\left(\exists t \in [q], \ |\Delta_\sigma(\phi_t)| > \tau \|\phi\|_{L^2(\nu_0)}\right) \leq \frac{q}{\tau^2} \sup_{\phi \in \mathcal{Q}_{\mathsf{A}}} \frac{\mathbb{E}_\sigma[\Delta_\sigma(\phi)^2]}{\|\phi\|_{L^2(\nu_0)}^2} \leq C \frac{q d^{-k_*}}{\tau^2}. \tag{26}$$

Hence, for $q/\tau^2 < d^{k_*}/C$, there exists $\sigma \in \mathsf{P}([d], P, \mathcal{S}_*)$ such that for every $t \in [q]$, we have $|\mathbb{E}_{\nu^\sigma}[\phi_t] - \mathbb{E}_{\nu_{\mathcal{S}_*}}[\phi_t]| \leq \tau \|\phi_t\|_{L^2(\nu_0)}$. Therefore Using Eq (26), we obtain that that there are two members from $\mathcal{H}_\mu^d$ whose query responses are compatible with the null $\nu_{\mathcal{S}_*}$, and hence also mutually compatible and indistinguishable from each other. This then establishes the failure of the learning algorithm in recovering the correct support.

Decomposing $\Delta_\sigma(\phi)$: We rewrite $\Delta_\sigma(\phi)$ in terms of $\mathbb{E}_{\nu_{\mathcal{S}_*}}$ using the Radon-Nikodym derivative:

$$\Delta_\sigma(\phi) = \mathbb{E}_{\nu_{\mathcal{S}_*}}\left[\frac{\mathrm{d}\nu^\sigma}{\mathrm{d}\nu_{\mathcal{S}_*}}\phi\right] - \mathbb{E}_{\nu_{\mathcal{S}_*}}[\phi] = \mathbb{E}_{\nu_{\mathcal{S}_*}}\left[\left(\frac{\mathrm{d}\nu^\sigma}{\mathrm{d}\nu_{\mathcal{S}_*}}(y_{\mathcal{S}_*}, \boldsymbol{x}) - 1\right)\phi(y_{\mathcal{S}_*}, \boldsymbol{x})\right]. \tag{27}$$

Recall that $(y_{\mathcal{S}_*}, \boldsymbol{x}_{\mathcal{S}_*})$ is independent of $\boldsymbol{x}_{\sigma(\mathcal{S}_*^c)} \sim \mu_x^{d-r}$ under $\nu_{\mathcal{S}_*}$, and that $\mathrm{d}\nu^\sigma/\mathrm{d}\nu_{\mathcal{S}_*} \in L^2(\nu_{\mathcal{S}_*})$ by Assumption 2.1. Therefore, we have the following orthogonal decomposition in $L^2(\nu_{\mathcal{S}_*})$ using the tensor basis (19):

$$\frac{\mathrm{d}\nu^\sigma}{\mathrm{d}\nu_{\mathcal{S}_*}}(y_{\mathcal{S}_*}, \boldsymbol{x}) - 1 = \sum_{\boldsymbol{i} \in \mathcal{I}^{d-r} \setminus \{\boldsymbol{0}\}} \xi_{\boldsymbol{i}}^\sigma(y_{\mathcal{S}_*}, \boldsymbol{x}_{\mathcal{S}_*})\psi_{\boldsymbol{i}}(\boldsymbol{x}_{\mathcal{S}_*^c}), \tag{28}$$

where

$$\xi_{\boldsymbol{i}}^\sigma(y_{\mathcal{S}_*}, \boldsymbol{x}_{\mathcal{S}_*}) := \mathbb{E}_{\nu_{\mathcal{S}_*}}\left[\frac{\mathrm{d}\nu^\sigma}{\mathrm{d}\nu_{\mathcal{S}_*}}(y_{\mathcal{S}_*}, \boldsymbol{x})\psi_{\boldsymbol{i}}(\boldsymbol{x}_{\mathcal{S}_*^c})\Big| y_{\mathcal{S}_*}, \boldsymbol{x}_{\mathcal{S}_*}\right] = \mathbb{E}_{\nu^\sigma}\left[\psi_{\boldsymbol{i}}(\boldsymbol{x}_{\mathcal{S}_*^c})\big| y_{\mathcal{S}_*}, \boldsymbol{x}_{\mathcal{S}_*}\right]. \tag{29}$$

Similarly, we decompose ϕ orthogonally in $L^2(\nu_{\mathcal{S}_*})$:

$$\phi(y_{\mathcal{S}_*}, \boldsymbol{x}) = \sum_{\boldsymbol{i} \in \mathcal{I}^{d-r}} \alpha_{\boldsymbol{i}}(y_{\mathcal{S}_*}, \boldsymbol{x}_{\mathcal{S}_*})\psi_{\boldsymbol{i}}(\boldsymbol{x}_{\mathcal{S}_*^c}), \tag{30}$$

where

$$\alpha_{\boldsymbol{i}}(y_{\mathcal{S}_*}, \boldsymbol{x}_{\mathcal{S}_*}) := \mathbb{E}_{\nu_{\mathcal{S}_*}}\left[\phi(y_{\mathcal{S}_*}, \boldsymbol{x})\psi_{\boldsymbol{i}}(\boldsymbol{x}_{\mathcal{S}_*^c})\big| y_{\mathcal{S}_*}, \boldsymbol{x}_{\mathcal{S}_*}\right]. \tag{31}$$

We deduce that we can decompose $\Delta_\sigma(\phi)$ as

$$\Delta_\sigma(\phi) = \sum_{\boldsymbol{i} \in \mathcal{I}^{d-r} \setminus \{\boldsymbol{0}\}} \mathbb{E}_{\nu_{\mathcal{S}_*}}\left[\xi_{\boldsymbol{i}}^\sigma(y_{\mathcal{S}_*}, \boldsymbol{x}_{\mathcal{S}_*})\alpha_{\boldsymbol{i}}(y_{\mathcal{S}_*}, \boldsymbol{x}_{\mathcal{S}_*})\right]. \tag{32}$$

For convenience, denote $m_{\boldsymbol{i}}^\sigma$ the summand. From the expressions (29) and (31) and by Fubini's theorem, we can rewrite $m_{\boldsymbol{i}}^\sigma$ as

$$\begin{aligned}
m_{\boldsymbol{i}}^\sigma &= \mathbb{E}_{y_{\mathcal{S}_*}, \boldsymbol{x}_{\mathcal{S}_*}}\left[\mathbb{E}_{\nu^\sigma}\left[\psi_{\boldsymbol{i}}(\boldsymbol{x}_{\mathcal{S}_*^c})\big| y_{\mathcal{S}_*}, \boldsymbol{x}_{\mathcal{S}_*}\right] \cdot \mathbb{E}_{\tilde{\boldsymbol{x}}_{\mathcal{S}_*^c} \sim \mu_x^{P-r}}\left[\phi(y_{\mathcal{S}_*}, \boldsymbol{x}_{\mathcal{S}_*}, \tilde{\boldsymbol{x}}_{\mathcal{S}_*^c})\psi_{\boldsymbol{i}}(\tilde{\boldsymbol{x}}_{\mathcal{S}_*^c})\right]\right] \\
&= \mathbb{E}_{(y, \boldsymbol{x}, \tilde{\boldsymbol{x}}_{\mathcal{S}_*^c}) \sim \nu^\sigma \otimes \mu_x^{P-r}}\left[\psi_{\boldsymbol{i}}(\boldsymbol{x}_{\mathcal{S}_*^c})\phi(y, \boldsymbol{x}_{\mathcal{S}_*}, \tilde{\boldsymbol{x}}_{\mathcal{S}_*^c})\psi_{\boldsymbol{i}}(\tilde{\boldsymbol{x}}_{\mathcal{S}_*^c})\right] \\
&= \mathbb{E}_{\tilde{\boldsymbol{x}}_{\mathcal{S}_*^c} \sim \mu_x^{P-r}}\left[\psi_{\boldsymbol{i}}(\tilde{\boldsymbol{x}}_{\mathcal{S}_*^c})\mathbb{E}_{(y, \boldsymbol{x}) \sim \nu^\sigma}\left[\phi(y, \boldsymbol{x}_{\mathcal{S}_*}, \tilde{\boldsymbol{x}}_{\mathcal{S}_*^c})\psi_{\boldsymbol{i}}(\boldsymbol{x}_{\mathcal{S}_*^c})\right]\right].
\end{aligned} \tag{33}$$

Let us decouple y from $\boldsymbol{x}_{\mathcal{S}_*}$ inside the query ϕ using the Radon-Nikodym derivative: conditional on $\tilde{\boldsymbol{x}}_{\mathcal{S}_*^c}$, we have

$$\begin{aligned}
&\mathbb{E}_{(y, \boldsymbol{x}) \sim \nu^\sigma}\left[\phi(y, \boldsymbol{x}_{\mathcal{S}_*}, \tilde{\boldsymbol{x}}_{\mathcal{S}_*^c})\psi_{\boldsymbol{i}}(\boldsymbol{x}_{\mathcal{S}_*^c})\right] \\
&= \mathbb{E}_{(y_{\mathcal{S}_*^c}, \boldsymbol{x}) \sim \nu_{\mathcal{S}_*^c}^\sigma}\left[\frac{\mathrm{d}\nu^\sigma}{\mathrm{d}\nu_{\mathcal{S}_*}^{\mathcal{S}^c}}(y_{\mathcal{S}_*^c}, \boldsymbol{x}) \cdot \phi(y_{\mathcal{S}_*^c}, \boldsymbol{x}_{\mathcal{S}_*}, \tilde{\boldsymbol{x}}_{\mathcal{S}_*^c})\psi_{\boldsymbol{i}}(\boldsymbol{x}_{\mathcal{S}_*^c})\right].
\end{aligned} \tag{34}$$

Similarly to Eq. (28), we have the following orthogonal decomposition in $L^2(\nu_{\mathcal{S}_*^c}^\sigma)$:

$$\frac{\mathrm{d}\nu^\sigma}{\mathrm{d}\nu_{\mathcal{S}_*^c}^{\mathcal{S}^c}}(y_{\mathcal{S}_*^c}, \boldsymbol{x}) = \sum_{\boldsymbol{j} \in \mathcal{I}^r} \xi_{\boldsymbol{j}}^\sigma(y_{\mathcal{S}_*^c}, \boldsymbol{x}_{\mathcal{S}_*^c})\psi_{\boldsymbol{j}}(\boldsymbol{x}_{\mathcal{S}_*}), \tag{35}$$

where
$$\xi_{\boldsymbol{j}}^{\sigma}(y_{\mathcal{S}_*^c}, \boldsymbol{x}_{\mathcal{S}_*^c}) := \mathbb{E}_{\nu^{\sigma}}\left[\psi_{\boldsymbol{j}}(\boldsymbol{x}_{\mathcal{S}_*}) \big| y_{\mathcal{S}_*^c}, \boldsymbol{x}_{\mathcal{S}_*^c}\right]. \tag{36}$$

By Fubini's theorem,
$$\mathbb{E}_{(y_{\mathcal{S}_*^c}, \tilde{\boldsymbol{x}}_{\mathcal{S}_*}, \boldsymbol{x}_{\mathcal{S}_*^c}) \sim \nu_{\mathcal{S}_*^c}^{\sigma}}\left[\xi_{\boldsymbol{j}}^{\sigma}(y_{\mathcal{S}_*^c}, \boldsymbol{x}_{\mathcal{S}_*^c})\psi_{\boldsymbol{j}}(\tilde{\boldsymbol{x}}_{\mathcal{S}_*}) \cdot \phi(y_{\mathcal{S}_*^c}, \tilde{\boldsymbol{x}}_{\mathcal{S}_*}, \tilde{\boldsymbol{x}}_{\mathcal{S}_*^c})\psi_{\boldsymbol{i}}(\boldsymbol{x}_{\mathcal{S}_*^c})\right]$$
$$= \mathbb{E}_{\tilde{\boldsymbol{x}}_{\mathcal{S}_*} \sim \mu_x^r}\left[\psi_{\boldsymbol{j}}(\tilde{\boldsymbol{x}}_{\mathcal{S}_*})\mathbb{E}_{(y,\boldsymbol{x}) \sim \nu^{\sigma}}\left[\phi(y, \tilde{\boldsymbol{x}}_{\mathcal{S}_*}, \tilde{\boldsymbol{x}}_{\mathcal{S}_*^c})\psi_{\boldsymbol{j}}(\boldsymbol{x}_{\mathcal{S}_*})\psi_{\boldsymbol{i}}(\boldsymbol{x}_{\mathcal{S}_*^c})\right]\right]. \tag{37}$$

Combining Eqs. (34), (35), and (37) into Eq. (33) yields the expression
$$m_{\boldsymbol{i}}^{\sigma} = \sum_{\boldsymbol{j} \in \mathcal{I}^r} \mathbb{E}_{\tilde{\boldsymbol{x}} \sim \mu_x^d}\left[\psi_{\boldsymbol{j}}(\tilde{\boldsymbol{x}}_{\mathcal{S}_*})\psi_{\boldsymbol{i}}(\tilde{\boldsymbol{x}}_{\mathcal{S}_*^c})\mathbb{E}_{(y,\boldsymbol{x}) \sim \nu^{\sigma}}\left[\phi(y, \tilde{\boldsymbol{x}})\psi_{\boldsymbol{j}}(\boldsymbol{x}_{\mathcal{S}_*})\psi_{\boldsymbol{i}}(\boldsymbol{x}_{\mathcal{S}_*^c})\right]\right]. \tag{38}$$

From Eq. (32), we deduce the following simple decomposition
$$\Delta_{\sigma}(\phi) = \sum_{\boldsymbol{i} \in \mathcal{I}^d,\ \operatorname{supp}(\boldsymbol{i}) \nsubseteq \mathcal{S}_*} \mathbb{E}_{\tilde{\boldsymbol{x}} \sim \mu_x^d}\left[\psi_{\boldsymbol{i}}(\tilde{\boldsymbol{x}})\mathbb{E}_{(y,\boldsymbol{x}) \sim \nu^{\sigma}}\left[\phi(y, \tilde{\boldsymbol{x}})\psi_{\boldsymbol{i}}(\boldsymbol{x})\right]\right]. \tag{39}$$

Bounding $\mathbb{E}_{\sigma}[\Delta_{\sigma}(\phi)^2]$: If $\operatorname{supp}(\boldsymbol{i}) \notin \sigma(\mathcal{C}_A \setminus \mathcal{C}_*)$, we have by Eq. (23) that
$$\mathbb{E}_{\tilde{\boldsymbol{x}} \sim \mu_x^d}\left[\psi_{\boldsymbol{i}}(\tilde{\boldsymbol{x}})\mathbb{E}_{(y,\boldsymbol{x}) \sim \nu^{\sigma}}\left[\phi(y, \tilde{\boldsymbol{x}})\psi_{\boldsymbol{i}}(\boldsymbol{x})\right]\right] = 0. \tag{40}$$

Therefore we can rewrite the sum in Eq. (39) with
$$\Delta_{\sigma}(\phi) = \sum_{\boldsymbol{i} \in \mathcal{I}^d} \sum_{S \in \mathcal{C}_A \setminus \mathcal{C}_*} \mathbb{E}_{(y,\boldsymbol{x},\tilde{\boldsymbol{x}}) \sim \nu^{\sigma} \otimes \mu_x^d}\left[\psi_{\boldsymbol{i}}(\tilde{\boldsymbol{x}})\phi(y, \tilde{\boldsymbol{x}})\psi_{\boldsymbol{i}}(\boldsymbol{x})\right] \cdot \mathbb{1}[\operatorname{supp}(\boldsymbol{i}) = \sigma(S)]$$
$$= \sum_{\boldsymbol{i} \in \mathcal{I}^d} \sum_{S \in \mathcal{C}_A \setminus \mathcal{C}_*} \mathbb{E}_{(y,\boldsymbol{x},\tilde{\boldsymbol{x}}) \sim \nu \otimes \mu_x^d}\left[\psi_{\boldsymbol{i}}(\tilde{\boldsymbol{x}})\phi(y, \tilde{\boldsymbol{x}})\psi_{\boldsymbol{i}}(\boldsymbol{x})\right] \cdot \mathbb{1}[\sigma(\operatorname{supp}(\boldsymbol{i})) = S]. \tag{41}$$

Recall that for every $S \in \mathcal{C}_A \setminus \mathcal{C}_*$ by construction. Hence,
$$\mathbb{E}_{\sigma}\left[\sum_{S_1, S_2 \in \mathcal{C}_A \setminus \mathcal{C}_*} \mathbb{1}[\sigma(\operatorname{supp}(\boldsymbol{i})) = S_1, \sigma(\operatorname{supp}(\boldsymbol{j})) = S_2]\right] \leq Cd^{-k_*}. \tag{42}$$

By squaring Eq. (41) and the previous display, we deduce that
$$\mathbb{E}_{\sigma}[\Delta_{\sigma}(\phi)^2]$$
$$\leq Cd^{-k_*}\left(\sum_{\boldsymbol{i} \in \mathcal{I}^d} |\mathbb{E}_{(y,\boldsymbol{x},\tilde{\boldsymbol{x}}) \sim \nu \otimes \mu_x^d}\left[\psi_{\boldsymbol{i}}(\tilde{\boldsymbol{x}})\phi(y, \tilde{\boldsymbol{x}})\psi_{\boldsymbol{i}}(\boldsymbol{x})\right]|\right)^2$$
$$\leq Cd^{-k_*}\left(\sum_{\boldsymbol{i} \in \mathcal{I}^d} \left|\mathbb{E}_{y \sim \mu_y}\left[\mathbb{E}_{\boldsymbol{x} \sim \mu_x^d}\left[\frac{\mathrm{d}\nu}{\mathrm{d}\nu_0}(y, \boldsymbol{x})\psi_{\boldsymbol{i}}(\boldsymbol{x})\right]\mathbb{E}_{\tilde{\boldsymbol{x}} \sim \mu_x^d}\left[\phi(y, \tilde{\boldsymbol{x}})\psi_{\boldsymbol{i}}(\tilde{\boldsymbol{x}})\right]\right]\right|\right)^2$$
$$\leq Cd^{-k_*}\left(\sum_{\boldsymbol{i} \in \mathcal{I}^d} \mathbb{E}_{\mu_y}\left[\mathbb{E}_{\mu_x^d}\left[\frac{\mathrm{d}\nu}{\mathrm{d}\nu_0}(y, \boldsymbol{x})\psi_{\boldsymbol{i}}(\boldsymbol{x})\right]^2\right]^{1/2} \mathbb{E}_{\mu_y}\left[\mathbb{E}_{\tilde{\boldsymbol{x}} \sim \mu_x^d}\left[\phi(y, \tilde{\boldsymbol{x}})\psi_{\boldsymbol{i}}(\tilde{\boldsymbol{x}})\right]^2\right]^{1/2}\right)^2 \tag{43}$$
$$\leq Cd^{-k_*}\left(\sum_{\boldsymbol{i} \in \mathcal{I}^d} \mathbb{E}_{\mu_y}\left[\mathbb{E}_{\mu_x^d}\left[\frac{\mathrm{d}\nu}{\mathrm{d}\nu_0}(y, \boldsymbol{x})\psi_{\boldsymbol{i}}(\boldsymbol{x})\right]^2\right]\right)\left(\sum_{\boldsymbol{i} \in \mathcal{I}^d} \mathbb{E}_{\mu_y}\left[\mathbb{E}_{\tilde{\boldsymbol{x}} \sim \mu_x^d}\left[\phi(y, \tilde{\boldsymbol{x}})\psi_{\boldsymbol{i}}(\tilde{\boldsymbol{x}})\right]^2\right]\right)$$
$$\leq Cd^{-k_*}\mathbb{E}_{y \sim \mu_y}\left[\left\|\frac{\mathrm{d}\nu}{\mathrm{d}\nu_0}(y, \cdot)\right\|_{L^2(\mu_x^d)}^2\right] \cdot \mathbb{E}_{y \sim \mu_y}\left[\|\phi(y, \cdot)\|_{L^2(\mu_x^d)}^2\right]$$
$$= Cd^{-k_*}\left\|\frac{\mathrm{d}\nu}{\mathrm{d}\nu_0}\right\|_{L^2(\nu_0)}^2 \|\phi\|_{L^2(\nu_0)}^2 \leq C'd^{-k_*}\|\phi\|_{L^2(\nu_0)}^2,$$

where we used in the third and fourth line Cauchy-Schwarz inequality, and in the last line
$$\frac{\mathrm{d}\nu}{\mathrm{d}\nu_0} = \frac{\mathrm{d}\mu_{y,\boldsymbol{z}}}{\mathrm{d}\mu_y \otimes \mu_x^P} \in L^2(\nu_0), \tag{44}$$

by Assumption 2.1. Note that the above bound is uniform over all $\phi \in L^2(\nu_0)$. This concludes the proof.

Remark B.1. *We remark that indeed these proofs, after alternatively defining $k_* = \mathsf{relLeap}(\mathcal{C}_\mathsf{A})$ and $\mathcal{C}_*$ to be the maximal subset of $\mathcal{C}_\mathsf{A}$ such that $\mathsf{relLeap}(\mathcal{C}_*) = k_* - 1$, shows that if $q/\tau^2 \leq cd^{k_*}$, then mthods in A fail to recover coordinates from $\mathcal{C}_\mathsf{A} \setminus \mathcal{C}_*$.*

B.3   Theorem 5.1.(b): lower bound for non-adaptive queries

The proof follows from a similar argument to the adaptive case.

Proof outline: Let $k_* = \mathsf{Cover}(\mathcal{C}_\mathsf{A})$ and l_* that maximizes $\min_{S \in \mathcal{C}_\mathsf{A}, l \in S} |S|$. By definition of the cover complexity, $|S| \geq k_*$ for all $S \in \mathcal{C}_\mathsf{A}$ with $l_* \in S$. Let $\mathcal{S}_* := [P] \setminus \{l_*\}$. We define in the non-adaptive case for all $\phi \in \mathcal{Q}_\mathsf{A}$ and $\sigma \in \mathsf{P}([d], P)$,

$$\Delta_\sigma(\phi) = \mathbb{E}_{\nu^\sigma}[\phi] - \mathbb{E}_{\nu^\sigma_{\mathcal{S}_*}}[\phi], \tag{45}$$

meaning that we compare the query expectation between ν^σ and $\nu^\sigma_{\mathcal{S}_*}$ where we decoupled the label y from coordinate $\sigma(l_*)$. We show again that the bound (25) holds on the second moment of $\Delta_\sigma(\phi)/\|\phi\|_{L^2(\nu_0)}$. Therefore, for any set of q queries $\{\phi_t\}_{t \in [q]} \subset \mathcal{Q}_\mathsf{A}$, we get a similar bound (26) on the probability of all queries being less or equal to τ in absolute value. We deduce that for $q/\tau^2 \leq cd^{k_*}$, the algorithm will not be able to distinguish for all $\sigma \in \mathsf{P}([d], P)$ between ν^σ and $\nu^\sigma_{\mathcal{S}_*}$, and therefore recover the coordinate $\sigma(l_*)$ in the support.

Bounding $\mathbb{E}_\sigma[\Delta_\sigma(\phi)^2]$:   Following the same decomposition of $\Delta_\sigma(\phi)$ as above, we first obtain

$$\Delta_\sigma(\phi) = \sum_{i \in \mathcal{I} \setminus \{0\}} \mathbb{E}_{\nu^\sigma_{\mathcal{S}_*}} \left[\xi_i^\sigma(y^\sigma_{\mathcal{S}_*}, \boldsymbol{x}_{\sigma(\mathcal{S}_*)}) \alpha_i(y^\sigma_{\mathcal{S}_*}, \boldsymbol{x}_{\sigma(\mathcal{S}_*)}) \right], \tag{46}$$

and then

$$\Delta_\sigma(\phi) = \sum_{\boldsymbol{i} \in \mathcal{I}^d,\, \sigma(l_*) \in \mathrm{supp}(\boldsymbol{i}) \not\subseteq \sigma(\mathcal{S}_*)} \mathbb{E}_{\tilde{\boldsymbol{x}} \sim \mu_x^d} \left[\psi_{\boldsymbol{i}}(\tilde{\boldsymbol{x}}) \mathbb{E}_{(y, \boldsymbol{x}) \sim \nu^\sigma}[\phi(y, \tilde{\boldsymbol{x}}) \psi_{\boldsymbol{i}}(\boldsymbol{x})] \right]. \tag{47}$$

Therefore, by Eq. (23), the summand is non zero only if $\mathrm{supp}(\boldsymbol{i}) \subseteq \sigma(\mathcal{C}_\mathsf{A})$ with $\sigma(l_*) \in \mathrm{supp}(\boldsymbol{i})$. Recall that in this case $|\mathrm{supp}(\boldsymbol{i})| \geq k_*$ by the choice of l_*. The rest of the proof follows using the same computation as in the previous section.

B.4   Upper Bounds for Adaptive and Non-Adaptive Queries

Define

$$\beta = \min_{U \in \mathcal{C}_\mathsf{A}} \left| \mathbb{E}_{\mu_{y,z}} \left[T^U(y) \prod_{i \in U} T_i^U(z_i) \right] \right|, \tag{48}$$

with $T^U \in \Psi_\mathsf{A}$ and $T_i^U \in L_0^2(\mu_x)$ chosen to make the expectation non-zero, where without loss of generality we normalize them so that $\|T^U\|_{L^2(\mu_y)}^2 \prod_{i \in S} \|T_i^U\|_{L^2(\mu_x)} = 1$. We can then emulate the support recovery algorithm described below Definition 1 in Section 4 using for every $U_* \in \mathcal{C}_\mathsf{A}$ and $|S| = |U_*|$, $\phi_S(y, \boldsymbol{x}) = T^{U_*}(y) \prod_{i \in S} T_i^{U_*}(x_i)$ (note that here S is an ordered subset) and precision $\tau < \beta/2$, so that if the response $|v_t| > \beta/2$, then $S \in \sigma_*(\mathcal{C}_\mathsf{A})$. Note that $\phi_S(y, \boldsymbol{x})$ are indeed in $\mathcal{Q}_\mathsf{A}$: for SQ, CSQ this is direct, for DLQ_ℓ, we can take $f_t(\boldsymbol{x}; \omega) = \boldsymbol{u} + \omega \cdot \boldsymbol{v} \prod_{i \in S} T_i^{U_*}(x_i)$.

If we take $\tau < \beta/2$, the number of queries only needs to scale as $O(d^{\mathsf{Cover}_\mathsf{A}})$ or $O(d^{\mathsf{Leap}_\mathsf{A}})$. We can further trade-off the precision and accuracy and get the result of several of the above queries with one query with small enough τ. For example, consider testing d subset $\{i\}$ to find $s_*(1)$. Then, one can group each of this query in $\Theta(\log(d))$ groups G_k following the binary representation of i. For each of these groups, consider the query

$$\frac{1}{\sqrt{d}} \sum_{s \in G_k} T^{\{1\}}(y) T_1^{\{1\}}(z_i).$$

Note that this is indeed a query in Ψ_A: it has $L^2(\nu_0)$ norm $O(1)$ (indeed $\mathbb{E}[T^{(i)}(z_i) T^{(j)}(z_j)] = 0$ for any $i \neq j$ with at least one not in the support). With precision $O(1/\sqrt{d})$, one can test if one of those expectation is non-zero. Repeating for each of the $\Theta(\log(d))$ groups allows to recover the binary representation of $\{s_*(1)\}$. This idea allows to trade-off the query and precision for any $q/\tau^2 \geq Cd^{k_*} \log(d)$.

C Proofs of Technical Results

Proposition 6.1: Denote $\mathcal{Y} = \{a, b\}$. For any $U \in \mathcal{C}_{\mathsf{SQ}}(\mu)$, by Definition 2, there exists $T \in L^2(\mu_y)$ and $\{T_i \in L_0^2(\mu_x)\}_{i \in S}$ such that

$$
\begin{aligned}
0 \neq \mathbb{E}\Big[T(y) \prod_{i \in U} T_i(z_i)\Big] & \\
&= T(a)\mu_{y,\mathbf{z}}(y=a)\mathbb{E}\Big[\prod_{i \in U} T_i(z_i)\Big|y=a\Big] + T(b)\mu_{y,\mathbf{z}}(y=b)\mathbb{E}\Big[\prod_{i \in U} T_i(z_i)\Big|y=b\Big] \quad (49) \\
&= (T(b) - T(a))\mu_{y,\mathbf{z}}(y=b)\mathbb{E}\Big[\prod_{i \in U} T_i(z_i)\Big|y=b\Big],
\end{aligned}
$$

where we used that $\mathbb{E}[T_i(z_i)] = 0$. Hence, this expectation is non-zero for any mapping $T(b) \neq T(a)$, in particular $T = \text{Id}$. We deduce that $U \in \mathcal{C}_{\mathsf{CSQ}}(\mu)$ too.

Consider $\mathcal{X} = \mathbb{R}$ and $\mu_x = \gamma$ the standard Gaussian measure. We assume that the label $y = z_1 z_2 \cdots z_P$. Then $\mathcal{C}_{\mathsf{CSQ}}$ only contains the entire support $[P]$ and $\mathsf{Cover}_{\mathsf{CSQ}}(\mu) = \mathsf{Leap}_{\mathsf{CSQ}}(\mu) = P$. On the other hand, consider the mappings $T(y) = \mathbb{1}[|y| \geq \tau]$ for some $\tau > 0$, and $T_i(z_i) = z_i^2 - 1$. Then

$$
\mathbb{E}[T(y)T(z_i)] = \mathbb{E}_{\mathbf{z}_{-i}}\Big[\mathbb{E}_{z_i}\Big[(z_i^2 - 1)\mathbb{1}\Big[|z_i| \cdot \prod_{j \neq i} |z_j| \geq \tau\Big]\Big]\Big] > 0, \quad (50)
$$

and $\mathcal{C}_{\mathsf{SQ}}(\mu)$ contains all the singletons $\{i\}$ for $i \in [P]$. We deduce $\mathsf{Cover}_{\mathsf{SQ}}(\mu) = \mathsf{Leap}_{\mathsf{SQ}}(\mu) = 1$.

Proposition 6.2.(a): Note that $\ell'(u, y) = (u - y)$ for the squared loss. For any $U \subseteq [P]$, and any $\{T_i \in L_2^0(\mu_x) : i \in [P]\}$, we have

$$
\mathbb{E}_{\mu_{y,\mathbf{z}}}\Big[(u - y)\prod_{i \in U} T_i(z_i)\Big] = \mathbb{E}_{\mu_{y,\mathbf{z}}}\Big[(u \prod_{i \in U} T_i(z_i)\Big] - \mathbb{E}_{\mu_{y,\mathbf{z}}}\Big[y \prod_{i \in U} T_i(z_i)\Big] = -\mathbb{E}_{\mu_{y,\mathbf{z}}}\Big[y \prod_{i \in U} T_i(z_i)\Big].
$$

Thus,

$$
\mathbb{E}_{\mu_{y,\mathbf{z}}}\Big[(u - y)\prod_{i \in U} T_i(z_i)\Big] \neq 0 \text{ if and only if } \mathbb{E}_{\mu_{y,\mathbf{z}}}\Big[(y \prod_{i \in U} T_i(z_i)\Big].
$$

This immediately implies $\mathcal{C}_{\mathsf{DLQ}_\ell} = \mathcal{C}_{\mathsf{SQ}}$ by Definition 2, and as a consequence $\mathsf{Leap}_{\mathsf{CSQ}} = \mathsf{Leap}_{\mathsf{DLQ}_\ell}$ and $\mathsf{Cover}_{\mathsf{CSQ}} = \mathsf{Cover}_{\mathsf{DLQ}_\ell}$.

Proposition 6.2.(b): Consider $U \in \mathcal{C}_{\mathsf{SQ}}$ with

$$
\mathbb{E}_{\mu_{y,\mathbf{z}}}\Big[T(y) \prod_{i \in U} T_i(z_i)\Big] \neq 0. \quad (51)
$$

Note that without loss of generality we can assume $T \in L_0^2(\mu_y)$. If $\text{span}(\Psi_{\mathsf{DLQ}_\ell})$ is dense in $L_0^2(\mu_y)$, then for every $\varepsilon > 0$, there exists $M_\varepsilon \in \mathbb{N}$ and $(\mathbf{v}_j, \mathbf{u}_j)_{j \in [M_\varepsilon]} \subseteq \mathcal{V} \times \mathcal{F}$ such that

$$
\tilde{T}_{M_\varepsilon}(y) = \sum_{j \in [M_\varepsilon]} \mathbf{v}_j^\mathsf{T} \nabla \ell(\mathbf{u}_j, y) \quad (52)
$$

has $\|T - \tilde{T}_{M_\varepsilon}\|_{L^2(\mu_y)} \leq \varepsilon$. In particular,

$$
\Big|\mathbb{E}_{\mu_{y,\mathbf{z}}}\Big[\tilde{T}_{M_\varepsilon}(y) \prod_{i \in U} T_i(z_i)\Big] - \mathbb{E}_{\mu_{y,\mathbf{z}}}\Big[T(y) \prod_{i \in U} T_i(z_i)\Big]\Big| \leq \|T - \tilde{T}_{M_\varepsilon}\|_{L^2(\mu_y)} \prod_{i \in U} \|T_i\|_{L^2(\mu_x)}. \quad (53)
$$

Therefore taking ε sufficiently small, $\tilde{T}_{M_\varepsilon}$ has non zero correlation with $\prod_{i \in U} T_i$, and in particular, one of the $\mathbf{v}_j^\mathsf{T} \nabla \ell(\mathbf{u}_j, y)$ must have non-zero correlation too. We deduce that $\mathcal{C}_{\mathsf{SQ}} \subseteq \mathcal{C}_{\mathsf{DLQ}_\ell}$ and we conclude using $\mathcal{C}_{\mathsf{DLQ}_\ell} \subseteq \mathcal{C}_{\mathsf{SQ}}$.

Let us assume that there exists nonzero bounded $T \in L_0^2(\mu_y)$, which we take without loss of generality $T(y) \subseteq [-1, 1]$, such that $\mathbb{E}[\mathbf{v}^\mathsf{T} \nabla \ell(\mathbf{u}, y)T(y)] = 0$ for all $\mathbf{u} \in \mathcal{F}, \mathbf{v} \in V$. The goal is to

define $\mu_{y,\boldsymbol{z}}$ such that $\mathcal{C}_{\mathsf{DLQ}_\ell} \subsetneq \mathcal{C}_{\mathsf{SQ}}$. Specifically, we will construct a joint distribution on (y, z_i) with $\{i\} \in \mathcal{C}_{\mathsf{SQ}}$ but $\{i\} \notin \mathcal{C}_{\mathsf{DLQ}_\ell}$. Let $A \subset \mathcal{X}$ with $\mu_x(A) \in (0,1)$ and consider y that only depends on z_i through $\mathbb{1}[z_i \in \mathcal{A}]$. Denote $P(A|y) = \mathbb{P}(z_i \in A|y)$. For any $T_i \in L^2_0(\mu_x)$,

$$\mathbb{E}[T_i(z_i)|y] = P(A|y)\mathbb{E}[T_i(z_i)|z_i \in A] + P(A^c|y)\mathbb{E}[T_i(z_i)|z_i \notin A]. \tag{54}$$

Denote $\kappa(A) = \mathbb{E}[T_i(z_i)|z_i \in A]$. By definition $\mathbb{E}[T_i(z_i)] = 0 = \mu_x(A)\kappa(A) + \mu_x(A^c)\kappa(A^c)$, and therefore $\kappa(A^c) = -\mu_x(A)\kappa(A)/(1 - \mu_x(A))$. Hence,

$$\mathbb{E}[T_i(z_i)|y] = \frac{\kappa(A)}{1 - \mu_x(A)}\left[P(A|y) - \mu_x(A)\right]. \tag{55}$$

Taking $\lambda > (1 - \mu_x(A))/\mu_x(A)$, we can set

$$P(A|y) := \lambda^{-1}(1 - \mu_x(A))T(y) + \mu_x(A) \in (0,1). \tag{56}$$

We then define the joint distribution of (y, z_i) as

$$\mu_{y,z_i} := [\mu_x(z_i|z_i \in A)P(A|y) + \mu_x(z_i|z_i \notin A)P(A^c|y)]\mu_y(y). \tag{57}$$

Note that $\mathbb{E}[P(A|y)] = \mu_x(A)$ using that $\mathbb{E}[T(y)] = 0$, and therefore the marginals of μ_{y,z_i} are indeed μ_y and μ_x. For any $T_i \in L^2_0(\mu_x)$ and $(\boldsymbol{u}, \boldsymbol{v}) \in \mathcal{F} \times V$, we have

$$\mathbb{E}_{\mu_{y,z_i}}[\boldsymbol{v}^\mathsf{T}\nabla\ell(\boldsymbol{u}, y)T_i(z_i)] = \frac{\kappa(A)}{\lambda}\mathbb{E}_{\mu_y}[\boldsymbol{v}^\mathsf{T}\nabla\ell(\boldsymbol{u}, y)T(y)] = 0, \tag{58}$$

and therefore $\{i\} \notin \mathcal{C}_{\mathsf{DLQ}_\ell}$. On the other hand $T \in L^2(\mu_y)$ and therefore $\{i\} \in \mathcal{C}_{\mathsf{SQ}}$. We conclude that the joint distribution (y, z_i) has $\mathsf{Leap}_{\mathsf{DLQ}_\ell} = \mathsf{Cover}_{\mathsf{DLQ}_\ell} = \infty$ while $\mathsf{Leap}_{\mathsf{SQ}} = \mathsf{Cover}_{\mathsf{SQ}} = 1$.

Proposition 6.2.(c): We construct a simple example with $\mathcal{X} = \{+1, -1\}$ and $\mu_x = \mathrm{Unif}(\mathcal{X})$, i.e., uniform on the discrete hypercube. Consider $P = 2k$ and label

$$y = z_1 z_2 \ldots z_{2k}\left(\sum_{i \in [2k]} z_i\right). \tag{59}$$

The set $\mathcal{C}_{\mathsf{CSQ}}(\mu)$ contains all subsets $[2k] \setminus \{i\}$ for all $i \in [2k]$, and therefore

$$\mathsf{Cover}_{\mathsf{CSQ}}(\mu) = \mathsf{Leap}_{\mathsf{CSQ}}(\mu) = 2k - 1. \tag{60}$$

Consider the loss function $\ell(u, y) = \frac{1}{2}(y - u)^2 + \frac{1}{4}(y - u)^4$. The derivative is given by

$$\ell'(u, y) = u - y + u^3 - 3u^2 y + 3uy^2 - y^3. \tag{61}$$

We have

$$y^2 = 2k + 2\sum_{i<j} z_i z_j, \qquad y^3 = z_1 z_2 \ldots z_{2k}\left(\sum_{i \in [2k]} z_i\right)^3. \tag{62}$$

For $k \geq 3$, y^3 only contains monomials over $\geq 2k - 3 \geq 3$ coordinates. Therefore $\mathcal{C}_{\mathsf{DLQ}_\ell}(\mu)$ contains all pairs $\{i, j\}$ but no singleton. Hence,

$$\mathsf{Cover}_{\mathsf{DLQ}_\ell}(\mu) = \mathsf{Leap}_{\mathsf{DLQ}_\ell}(\mu) = 2. \tag{63}$$

Finally, for $\mathcal{C}_{\mathsf{SQ}}$, taking $T(y) = y^{2k-1}$, it contains all singleton $\{i\}$ and therefore

$$\mathsf{Cover}_{\mathsf{SQ}}(\mu) = \mathsf{Leap}_{\mathsf{SQ}}(\mu) = 1. \tag{64}$$

C.1 Proof of Theorem 6.3

Note that $\mathcal{C}_A$ does not change if we add or remove constant functions to Ψ_A.

ℓ_1**-loss:** First consider $\ell(u, y) = |y - u|$. We have $\nabla\ell(u, y) := \ell'(u, y) = \mathrm{sign}(u - y)$, where we can set without loss of generality $\mathrm{sign}(0) = 0$. From Hornik [1991, Theorem 5] and Hornik [1991, Theorem 1], we directly conclude that for any probability measure μ_y on $\mathbb{R}$, we have $\mathrm{span}(\Psi_A)$ dense in $L^2(\mu_y)$ and we conclude using Proposition 6.2.(b).

Hinge loss: Consider the Hinge loss $\ell(u, y) = \max(1 - yu, 0)$. We get $\ell'(u, y) = -y\mathbb{1}_{uy \leq 1}$ where without loss of generality, we set $\ell'(1/y, y) = -y$. For $a > 0$ and $b < 0$, we have $\ell'(a, y) - \ell'(b, y) = y\mathbb{1}[y > a] - y\mathbb{1}[y < b]$. Taking $b \to -\infty$, we have $\ell'(a, y) - \ell'(b, y)$ that converges to $y\mathbb{1}[y > a]$ in $L^2(\mu_y)$ (recall that we assume that the second moment of μ_y is bounded). Similarly, we can construct $y\mathbb{1}[y > a]$ for $a \leq 0$. Following the proof of Hornik [1991, Theorem 1], if $\mathrm{span}(\Psi_A)$ is not dense in $L^2(\mu_y)$, there exists $g \in L^2(\mu_y)$ such that

$$\int y\mathbb{1}[y > u]g(y)\mathrm{d}\mu_y = 0, \qquad \forall u \in \mathbb{R}. \tag{65}$$

Define the signed measure $\sigma(B) = \int_B yg(y)\mathrm{d}\mu_y$. It is finite since $\int |yg(y)|\mathrm{d}\mu_y \leq \|y\|_{L^2(\mu_y)}\|g\|_{L^2(\mu_y)} < \infty$ by assumption. We deduce that

$$\int \mathbb{1}[ay - u > 0]\mathrm{d}\sigma(y) = 0 \qquad \forall(a, u) \in \mathbb{R}^2. \tag{66}$$

(The case $a = 0$ is obtained by taking $u \to -\infty$ in Eq. (65).) We then use that $\mathbb{1}[ay - u > 0]$ is discriminatory and therefore no such finite signed measure exists.

Exponential loss: Consider $\mathcal{Y} \subseteq [-M, M]$ and $\ell(u, y) = e^{-uy}$, so that $\ell'(u, y) = -ye^{-yu}$. Any Borel probability measure μ_y on $[-M, M]$ is regular, and in particular continuous functions and therefore polynomials are dense in $L^2(\mu_y)$. To show that $\mathrm{span}\{ye^{-yu} : u \in \mathbb{R}\}$ is dense in $L^2(\mu_y)$, it is sufficient to show that we can approximate any monomial y^k in $L^2(\mu_y)$. This is readily proven recursively, using that

$$ye^{-yu} = \sum_{l=0}^{k-1} \frac{(-1)^l}{l!} u^l y^{l+1} + O(u^l) \cdot e^M. \tag{67}$$

This concludes the proof of this theorem.

D SGD, mean-field and limiting dimension-free dynamics

Fix a link distribution $y|\boldsymbol{z} \sim \mu_{y|\boldsymbol{z}}$ and consider data distribution $(y, \boldsymbol{x}) \sim \mathcal{D}_{s_*}^d \in \mathcal{H}_\mu^d$ with $\mu = (\mu_x, \mu_{y|\boldsymbol{z}})$ and (unknown) $s_* \in \mathrm{P}([d], P)$. We consider learning using a two-layer neural network with parameters $\boldsymbol{\Theta} \in \mathbb{R}^{M(d+2)+1}$ and an activation $\sigma : \mathbb{R} \to \mathbb{R}$:

$$f_{\mathsf{NN}}(\boldsymbol{x}; \boldsymbol{\Theta}) = c + \frac{1}{M}\sum_{j \in [M]} a_j\sigma(\langle \boldsymbol{w}_j, \boldsymbol{x}\rangle + b_j), \qquad c, a_j, b_j, \in \mathbb{R}, \quad \boldsymbol{w}_j \in \mathbb{R}^d, \text{ for } j \in [M] \tag{NN1}$$

For convenience, we will reparametrize the neural network for $\boldsymbol{\Theta} = (\boldsymbol{\theta}_j)_{j \in [M]} \in \mathbb{R}^{M(d+3)}$ with $\boldsymbol{\theta}_j = (a_j, \boldsymbol{w}_j, b_j, c_j) \in \mathbb{R}^{d+3}$ and

$$f_{\mathsf{NN}}(\boldsymbol{x}; \boldsymbol{\Theta}) = \frac{1}{M}\sum_{j \in [M]}\sigma_*(\boldsymbol{x}; \boldsymbol{\theta}_j), \qquad \sigma_*(\boldsymbol{x}; \boldsymbol{\theta}_j) = a_j\sigma(\langle \boldsymbol{w}_j, \boldsymbol{x}\rangle + b_j) + c_j. \tag{NN2}$$

Indeed, the two neural networks (NN1) and (NN2) remain equal throughout training under the initialization $c_1 = \ldots = c_M = c$.

Also, recall the definitions of the risk and the excess risk, respectively, for a data distribution $(y, \boldsymbol{x}) \sim \mathcal{D}$, with respect to a loss function $\ell : \mathbb{R} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$.

$$\mathcal{R}(f) = \mathbb{E}_{\mathcal{D}}\left[\ell(f(\boldsymbol{x}), y)\right], \quad \text{and} \quad \overline{\mathcal{R}}(f) = \mathcal{R}(f) - \inf_{\bar{f}:\{\pm 1\}^d \to \mathbb{R}}\mathcal{R}(\bar{f}). \tag{68}$$

We start with the initialization specified by ρ_0

$$(a^0, b^0, \sqrt{d} \cdot \boldsymbol{w}^0, c^0) \sim \mu_a \otimes \mu_b \otimes \mu_w^{\otimes d} \otimes \delta_{c=\bar{c}} := \rho_0. \tag{69}$$

Batch Stochastic Gradient Descent. We train the parameters $\boldsymbol{\Theta}$ using batch-SGD with loss ℓ and the batch size b. Even in more generality than stated in (ℓ-bSGD), we allow for time-varying step size $(\boldsymbol{\eta}_k)_{k\geq 0}$, where $\boldsymbol{\eta}_k \in \mathbb{R}^{d+3}$ can be different for different parameters (e.g., different layers), and ℓ_2-regularization $\boldsymbol{\lambda} \in \mathbb{R}^{d+3}$. We initialize the weights $(\boldsymbol{\theta}_j)_{j\in[M]} \overset{\text{i.i.d.}}{\sim} \rho_0$ from (69), and at each step, given samples $(\{(\boldsymbol{x}_{ki}, y_{ki}) : i \in [b]\})_{k\geq 0}$, the weights are updated using

$$\boldsymbol{\theta}_j^{k+1} = \boldsymbol{\theta}_j^k - \frac{1}{b}\sum_{i\in[b]}\ell'(f_{\mathsf{NN}}(\boldsymbol{x}_{ki}; \boldsymbol{\Theta}^t), y_{ki}) \cdot \boldsymbol{H}_k\nabla_{\boldsymbol{\theta}}\sigma_*(\boldsymbol{x}_{ki}; \boldsymbol{\theta}_j^k) - \boldsymbol{H}_k\boldsymbol{\Lambda}\boldsymbol{\theta}_j^k, \qquad (\ell\text{-bSGD-g})$$

where we introduced $\boldsymbol{H}_k := \mathrm{diag}(\boldsymbol{\eta}_k)$ and $\boldsymbol{\Lambda} := \mathrm{diag}(\boldsymbol{\lambda})$.

Mean-Field Dynamics. A rich line work [Chizat and Bach, 2018, Mei et al., 2018, Rotskoff and Vanden-Eijnden, 2018, Sirignano and Spiliopoulos, 2020] has established a crisp approximation between the dynamics of one-pass-batch-SGD (ℓ-bSGD-g) and a continuous dynamics in the space of probability distributions in $\mathbb{R}^{d+3}$. To any distribution $\rho \in \mathcal{P}(\mathbb{R}^{d+3})$, we associate the infinite-width neural network

$$f_{\mathsf{NN}}(\boldsymbol{x}; \rho) = \int \sigma_*(\boldsymbol{x}; \boldsymbol{\theta})\rho(\mathrm{d}\boldsymbol{\theta}). \tag{70}$$

In particular, we recover the finite-width neural network (NN2) by taking $\hat{\rho}^{(M)} := \frac{1}{M}\sum_{j\in[M]}\delta_{\boldsymbol{\theta}_j}$.

Denote $\hat{\rho}_k^{(M)}$ the empirical distribution of the weights $\{\boldsymbol{\theta}_j^k\}_{j\in[M]}$ after k-steps of (ℓ-bSGD-g). In our (ℓ-bSGD-g) dynamics, we allow for a generic step-size schedule that is captured by

$$\boldsymbol{\eta}_k = \eta\boldsymbol{\xi}(k\eta) \quad \text{for some function} \quad \boldsymbol{\xi} : \mathbb{R} \to \mathbb{R}_{\geq 0}^{d+3}, \tag{71}$$

and that the data at each step are sampled i.i.d. from $\mathcal{D}$; the case of $\mathcal{D}$ empirical distribution on training data corresponds to multi-pass batch-SGD, while $\mathcal{D}$ population distribution corresponds to online batch-SGD. Under some reasonable assumptions (that we will list below on the activation, step-size schedule $\boldsymbol{\xi}$, initialization ρ_0, and the loss ℓ) on taking M sufficiently large and η sufficiently small, setting $k = t/\eta$, $\hat{\rho}^{(M)}$ is well approximated by a distribution $\rho_t \in \mathcal{P}(\mathbb{R}^{d+3})$ that evolves according to the PDE:

$$\partial_t\rho_t = \nabla_{\boldsymbol{\theta}} \cdot (\rho_t\boldsymbol{H}(t)\nabla_{\boldsymbol{\theta}}\psi(\boldsymbol{\theta}; \rho_t)),$$
$$\psi(\boldsymbol{\theta}; \rho_t) = \mathbb{E}_{\mathcal{D}}[\ell'(f_{\mathsf{NN}}(\boldsymbol{x}; \rho_t), y)\sigma_*(\boldsymbol{x}; \boldsymbol{\theta})] + \frac{1}{2}\boldsymbol{\theta}^\mathsf{T}\boldsymbol{\Lambda}\boldsymbol{\theta}, \tag{MF-PDE}$$

where the initial distribution is ρ_0 and we defined $\boldsymbol{H}(t) = \mathrm{diag}(\boldsymbol{\xi}(t))$. We will refer to the distribution dynamics (MF-PDE) as the mean-field (MF) dynamics. We now specify the assumptions on hyperparameters under which a non-asymptotic equivalence can be derived.

A D.1 (Activation). *Our $\sigma : \mathbb{R} \to \mathbb{R}$ is three times differentiable with $\|\sigma^{(k)}\|_\infty \leq K$, $k = 0,\ldots,3$.*

A D.2 (Bounded, Lipschitz step-size schedule). *$t \mapsto \boldsymbol{\xi}(t) = (\xi_i(t))_{i\in[d+3]}$ has bounded Lipschitz entries: $\|\xi_i\|_\infty \leq K$ and $\|\xi_i\|_{\mathrm{Lip}} \leq K$.*

A D.3 (Initialization). *The initial distribution $\rho_0 \in \mathcal{P}(\mathbb{R}^{d+3})$ is of the form:*

$$(a^0, b^0, \sqrt{d}\cdot\boldsymbol{w}^0, c^0) \sim \mu_a \otimes \mu_b \otimes \mu_w^{\otimes d} \otimes \delta_{c=\bar{c}},$$

where μ_a, μ_w, μ_b are independent of d. We further assume that μ_a supported on $|a| \leq K$, $|\bar{c}| \leq K$, and μ_w is symmetric and K^2-sub-Gaussian.

A D.4 (Loss). *The loss $\ell : \mathbb{R} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ is twice-differentiable in its first argument for all $y \in \mathcal{Y}$ and satisfies*

$$|\ell'(u, y)| \leq K(1 + |u|) \text{ and } |\ell''(u, y)| \leq K \text{ for all } y \in \mathcal{Y}.$$

Assumptions D.1-D.3 are similar to Abbe et al. [2022]. Informally, Assumption D.4 states that loss has a quadratic upper bound (which is analogues to the smoothness assumption in the convex optimization literature). Later, to show the learnability of leap one junta settings, we will further require $\ell(\cdot, y)$ to be convex and analytic. We note that together these assumptions holds for (i) the standard logistic loss for the classification setting when $\mathcal{Y} \in \{\pm 1\}$, and (ii) for the squared loss and any of its "analytic perturbations" for the regression setting when $\mathcal{Y} \subseteq [-B, B]$. In particular, we would like to mention $\ell(u, y) = (u - y)\mathrm{archsinh}(u - y)$, for which we have $\mathsf{Leap}_{\mathsf{DLQ}_\ell} = \mathsf{Leap}_{\mathsf{SQ}}$ and these assumptions also hold, and hence our results apply. For classification, by Proposition 6.1, anyway $\mathsf{LeapSQ} = \mathsf{LeapCSQ} = \mathsf{Leap}_{\mathsf{DLQ}_\ell}$ for any non-degenerate loss (including the standard logistic loss for which our results hold).

Limiting Dimension-Free Dynamics. In the junta learning setting, when y only depends on $P \lll d$ coordinates, following Abbe et al. [2022, Secion 3], the SGD dynamics (ℓ-bSGD) concentrates on an effective *dimension-free* (DF) dynamics as $M, d \to \infty$ and $\eta \to 0$. This equivalence holds under a certain assumption on the loss function, and other assumptions on the initialization and activation that are similar to the setup of Abbe et al. [2022].

Consider the isotropic parameters $\boldsymbol{H}_k = \mathrm{diag}(\eta_k^a, \eta_k^w \mathbf{I}_d, \eta_k^b, \eta_k^c)$ and $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda^a, \lambda^w \mathbf{I}_d, \lambda^b, \lambda^c)$. Using technical ideas from Abbe et al. [2022, Secion 3], the (ℓ-bSGD-g) concentrates on an effective *dimension-free* dynamics in the limit $M, d \to \infty$ and $\eta \to 0$. This limiting dynamics corresponds to the gradient flow on $\mathcal{R}(f) + \boldsymbol{\theta}^\mathsf{T} \boldsymbol{\Lambda} \boldsymbol{\theta}$ of the following effective *infinite-width* neural network (recall $\boldsymbol{z} \in \mathbb{R}^P$ is the support)

$$f_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}_t) = \int \overline{\sigma}_*(\boldsymbol{z}; \bar{\boldsymbol{\theta}}^t) \bar{\rho}_t(\mathrm{d}\bar{\boldsymbol{\theta}}^t), \qquad \overline{\sigma}_*(\boldsymbol{z}; \bar{\boldsymbol{\theta}}^t) = c^t + a^t \mathbb{E}_G[\sigma(\langle \boldsymbol{z}, \boldsymbol{u}^t \rangle + s^t G + b^t)], \quad (72)$$

where $G \sim \mathsf{N}(0, 1)$ and $\bar{\rho}_t \in \mathcal{P}(\mathbb{R}^{P+4})$ is the distribution over $\bar{\boldsymbol{\theta}}^t = (a^t, b^t, \boldsymbol{u}^t, c^t, s^t)$ with $\boldsymbol{u}^t \in \mathbb{R}^P$.

$$\partial_t \bar{\rho}_t = \nabla_{\bar{\boldsymbol{\theta}}} \cdot \left(\bar{\rho}_t \overline{\boldsymbol{H}}(t) \cdot \nabla_{\bar{\boldsymbol{\theta}}} \psi(\bar{\boldsymbol{\theta}}, \bar{\rho}_t) \right),$$

$$\psi(\bar{\boldsymbol{\theta}}, \bar{\rho}_t) = \mathbb{E}_{\mu_{y,z}} \left[\ell'(f_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}_t), y) \overline{\sigma}_*(\boldsymbol{z}; \bar{\boldsymbol{\theta}}) \right] + \frac{1}{2} \bar{\boldsymbol{\theta}}^\mathsf{T} \overline{\boldsymbol{\Lambda}} \bar{\boldsymbol{\theta}}, \qquad \text{(DF-PDE)}$$

where $\overline{\boldsymbol{H}}(t) = \mathrm{diag}(\xi^a, \xi^b, \xi^w \mathbf{I}_P, \xi^c, \xi^w)(t)$ and $\overline{\boldsymbol{\Lambda}} = \mathrm{diag}(\lambda^a, \lambda^b, \lambda^w \mathbf{I}_P, \lambda^w)$ from initialization

$$\bar{\rho}_0 := \mu_a \otimes \mu_b \otimes \delta_{\boldsymbol{u}^0 = \boldsymbol{0}} \otimes \delta_{c^0 = \bar{c}} \otimes \delta_{s^0 = \sqrt{m_2^w}},$$

where $m_2^w = \mathbb{E}_{W \sim \mu_w}[W^2]^{1/2}$ is the second moment of μ_w. The non-asymptotic equivalence is characterized by the following theorem that can be seen as a generalization of [Abbe et al., 2022, Theorem 5] to non-squared losses.

Theorem D.5. *Fix activation, step-size schedule, initialization, and the loss such that Assumptions D.1 to D.4 hold and consider any $T \geq 1$ independent of d. Let $(\bar{\rho}_t)_{t \geq 0}$ be the solution of (DF-PDE), and $\{\boldsymbol{\Theta}_k\}_{k \geq 0}$ the trajectory of SGD (ℓ-bSGD-g) with initialization $\{\boldsymbol{\theta}_j^0\}_{j \in [M]} \overset{i.i.d.}{\sim} \rho_0$. Then there exist a constant $C_{K,T} > 0$ that only depends on K, T, such that for any $d, M, \eta > 0$ with $M \leq e^d$ and $0 < \eta \leq 1/[C_{K,T}(d + \log(M))]$, and*

$$\left| \mathcal{R}(f_{\mathsf{NN}}(\cdot; \boldsymbol{\Theta}^k)) - \mathcal{R}(f_{\mathsf{NN}}(\cdot; \bar{\rho}_{\eta k})) \right| \leq C_{K,T} \left\{ \sqrt{\frac{P}{d}} + \sqrt{\frac{\log(M)}{M}} + \sqrt{\eta} \sqrt{\frac{d + \log(M)}{b} \vee 1} \right\} \quad (73)$$

for all $k \leq T/\eta$, with probability at least $1 - 1/M$.

This theorem follows by a straightforward modification of the proof of [Abbe et al., 2022, Theorem 16, Proposition 15] using propagation of chaos argument. We note that the only property about the loss that is needed is Assumption D.4.

D.1 DF-PDE alignment with the support.

In this subsection, we provide a formal statement of Theorem 7.1 and its proof. The following is an extension of [Abbe et al., 2022, Theorem 7] from square loss to general loss.

Theorem D.6 (Formal statement of Theorem 7.1)**.** *Assume that $\mathsf{Leap}_{\mathsf{DLQ}_\ell}(\mu) > 1$. Then there exists $U \subset [P]$, $|U| \geq 2$, such that for all $t \geq 0$, the DF dynamics solution remains independent of coordinates $(z_i)_{i \in U}$, i.e., $f_{\mathsf{NN}}(\boldsymbol{z}; \rho_t) = f_{\mathsf{NN}}((z_i)_{i \notin U}; \rho_t)$, and therefore fails at recovering the support.*

Conversely, if $\mathsf{Leap}_{\mathsf{DLQ}_\ell}(\mu) = 1$ and ℓ is analytic with respect to its first argument, then almost surely over $\bar{c}$, there exists $t > 0$ such that $f_{\mathsf{NN}}(\boldsymbol{z}; \rho_t)$ depends on all $[P]$.

Proof of Theorem D.6. The proof follows similarly to the proof of [Abbe et al., 2022, Theorem 7]. Assume $\mathsf{Leap}_{\mathsf{DLQ}_\ell}(\mu) > 1$. Consider $\mathcal{C}_* \subset \mathcal{C}_{\mathsf{DLQ}_\ell}$ the maximal subset such that $\mathsf{Leap}(\mathcal{C}_*) = 1$ and denote $U_* = \bigcup_{U \in \mathcal{C}_*} U$. In words, U_* contains the subset of coordinates that are reachable by doing leaps of at most 1. By the assumption that $\mathsf{Leap}_{\mathsf{DLQ}_\ell}(\mu) > 1$, we must have $|[P] \setminus U_*| \geq 2$. Let us show that the weights u_i for $i \in [P] \setminus U_*$ remain equal to 0 throughout the dynamics.

For simplicity, we forget about the other parameters which we set to $b = c = s = 0$. For convenience, denote $\Omega = [P] \setminus U_*$. First, we can bound by Gronwall's inequality, for $t \leq C$, $|a^t| \leq KC'$ for all a^t on the support of ρ_t. From the proof of Theorem 5.1, conditioning on already explored coordinates does not change the leap-complexity. In particular,

$$\mathbb{E}[\ell'(g(\boldsymbol{z}_{U_*}), y) z_i | \boldsymbol{z}_{U_*}] = 0, \qquad \forall i \in \Omega.$$

Consider the time derivative of u_i^t for $i \in \Omega$. We have

$$|\ell'(f_{\mathsf{NN}}(\boldsymbol{z}_{U_*}, \boldsymbol{z}_\Omega; \rho_t), y)\sigma'(\langle \boldsymbol{u}_{U_*}^t, \boldsymbol{z}_{U_*}\rangle + \langle \boldsymbol{u}_\Omega^t, \boldsymbol{z}_\Omega\rangle) - \ell'(f_{\mathsf{NN}}(\boldsymbol{z}_{U_*}, \boldsymbol{0}; \rho_t), y)\sigma'(\langle \boldsymbol{u}_{U_*}^t, \boldsymbol{z}_{U_*}\rangle)|$$

$$\leq K\|\sigma\|_\infty \int |\tilde{a}^t| |\sigma(\langle \tilde{\boldsymbol{u}}_{U_*}^t, \boldsymbol{z}_{U_*}\rangle + \langle \tilde{\boldsymbol{u}}_\Omega^t, \boldsymbol{z}_\Omega\rangle) - \sigma(\langle \tilde{\boldsymbol{u}}_{U_*}^t, \boldsymbol{z}_{U_*}\rangle)|\rho_t(\mathrm{d}\tilde{\boldsymbol{\theta}})$$

$$\quad + K\|f_{\mathsf{NN}}(\cdot; \rho_t)\|_\infty |\sigma(\langle \boldsymbol{u}_{U_*}^t, \boldsymbol{z}_{U_*}\rangle + \langle \boldsymbol{u}_\Omega^t, \boldsymbol{z}_\Omega\rangle) - \sigma(\langle \boldsymbol{u}_{U_*}^t, \boldsymbol{z}_{U_*}\rangle)|$$

$$\leq K \sup_{j \in \Omega, u_j^t \in \mathrm{supp}(\rho_t)} |u_j^t|,$$

where we used Assumptions D.1 and D.4. Hence,

$$\left| \frac{\mathrm{d}}{\mathrm{d}t} u_i^t \right| = \left| a^t \mathbb{E}[\ell'(f_{\mathsf{NN}}(\boldsymbol{z}, \rho_t), y)\sigma'(\langle \boldsymbol{u}^t, \boldsymbol{z}\rangle)z_i] \right|$$

$$\leq \left| a^t \mathbb{E}[\ell'(f_{\mathsf{NN}}(\boldsymbol{z}_{U_*}, \boldsymbol{0}; \rho_t), y)\sigma'(\langle \boldsymbol{u}_{U_*}^t, \boldsymbol{z}_{U_*}\rangle)z_i] \right| + K \sup_{j \in \Omega, u_j^t \in \mathrm{supp}(\rho_t)} |u_j^t|$$

The first expectation is equal to 0. Note that $m_\Omega^t := \sup_{j \in \Omega, u_j^t \in \mathrm{supp}(\rho_t)} |u_j^t|$ has $m_\Omega^0 = 0$ and therefore $m_\Omega^t = 0$ for all $t \geq 0$. $\qquad \square$

E Learning Leap One Juntas with SGD: Formalizing Theorem 7.2

In this section, we provide our precise layer-wise training algorithm and the proof that it succeeds in learning $\mathsf{Leap}_{\mathsf{DLQ}_\ell} = 1$ junta problems. This section follows the similar training procedure from Abbe et al. [2022] for the layer-wise training but adapted to training with general loss function $\ell : \mathbb{R} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ to show Theorem 7.2. The key difference in the training algorithm from Abbe et al. [2022] is the use of coordinate-wise perturbed step-sizes for the first layer weights to break the symmetry between the coordinates. This allows us to learn even some degenerate cases having coordinate symmetries with leap one, in contrast to Abbe et al. [2022].

Discrete Dimension-free Dynamics. We first consider the dimension-free dynamics similar to (DF-PDE) but for discrete step-size regime (see [Abbe et al., 2022, Appendix C]). We initialize to

$$(a^0, b^0, \boldsymbol{u}^0, c^0, s^0) \sim \bar{\rho}_0 \in \mathcal{P}(\mathbb{R}^{P+4}),$$

and $(\bar{\rho}_k)_{k \in \mathbb{N}}$ are induced distribution on $(a^0, b^0, \boldsymbol{u}^0, c^0, s^0)$ recursively defined as

$$a^{k+1} = a^k - \eta_k^a \left(\mathbb{E}_{\boldsymbol{z}, G}[\ell'(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}_k), y)\sigma(\langle \boldsymbol{u}^k, \boldsymbol{z}\rangle + s^k G + b^k)] + \lambda^a a^k \right)$$

$$\boldsymbol{u}^{k+1} = \boldsymbol{u}^k - \boldsymbol{\eta}_k^{\boldsymbol{u}} \circ \left(\mathbb{E}_{\boldsymbol{z}, G}[\ell'(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}_k), y)a^k \sigma'(\langle \boldsymbol{u}^k, \boldsymbol{z}\rangle + s^k G + b^k)\boldsymbol{z}] + \lambda^w \boldsymbol{u}^k \right)$$

$$s^{k+1} = s^k - \eta_k^w \left(\mathbb{E}_{\boldsymbol{z}, G}[\ell'(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}_k), y)a^k \sigma'(\langle \boldsymbol{u}^k, \boldsymbol{z}\rangle + s^k G + b^k)G] + \lambda^w s^k \right) \quad \text{(d-DF-PDE)}$$

$$b^{k+1} = b^k - \eta_k^b \left(\mathbb{E}_{\boldsymbol{z}, G}[\ell'(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}_k), y)a^k \sigma'(\langle \boldsymbol{u}^k, \boldsymbol{z}\rangle + s^k G + b^k)] + \lambda^b b^k \right)$$

$$c^{k+1} = c^k - \eta_k^c \left(\mathbb{E}_{\boldsymbol{z}, G}[\ell'(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}_k), y)] + \lambda^c c^k \right).$$

E.1 Training Algorithm

Choose a loss $\ell : \mathbb{R} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ and the activation $\sigma(x) = (1 + x)^L$ for $L \geq 2^{8P}$. We train for $\Theta(1)$ total steps in two phases and the batch size of $\Theta(d)$ with the following choice of hyperparameters.

- **No regularization:** set the regularization parameters $\lambda^w = \lambda^a = \lambda^b = \lambda^c = 0$.

- **Initialization:** Initialize the first layer weights and biases to deterministically 0.

$$(b_j^0, \sqrt{d}\boldsymbol{w}_j^0) \overset{\text{i.i.d.}}{\sim} \mu_b \otimes \mu_w^{\otimes d} \equiv \delta_{b=0} \otimes \delta_{w=0}^{\otimes d}.$$

The second layer weights are sampled uniformly from $[-1, 1]$, e.g. $a_j \overset{\text{i.i.d.}}{\sim} \mu_a \equiv$ Unif($[-1, 1]$). Finally, choose $c^0 \sim \delta_{c=\bar{c}}$ for the given global bias choice $\bar{c} \in \mathbb{R}$.

- For the dimension-free dynamics, this corresponds to taking

$$(a^0, b^0, \boldsymbol{u}^0, c^0, s^0) \sim \bar{\rho}^0 \equiv \text{Unif}([-1, 1]) \otimes \delta_{b=0} \otimes \delta_{w=0}^{\otimes P} \otimes \delta_{c=\bar{c}} \otimes \delta_{s=0}.$$

- We deploy a two-phase training procedure. Given parameters η and $\boldsymbol{\kappa} \in [1/2, 3/2]^d$:
    1. **Phase 1:** For all $k \in \{0, \ldots, k_1 - 1\}$, set $\eta_k^a = \eta_k^b = \eta_k^c = 0$ and $\boldsymbol{\eta}_k^{\boldsymbol{w}} \in \mathbb{R}^d$ such that $\eta_k^{w_i} = \eta\kappa_i$ for all $i \in [d]$. For the dimension-free dynamics, train the first layer weights $\boldsymbol{u}^k$ for k_1 steps, while keeping other parameters fixed, i.e $a^k = a^0, b^k = 0, c^k = \bar{c}$.
    2. **Phase 2:** Set $\eta_k^a = \eta$ and $\boldsymbol{\eta}_k^{\boldsymbol{w}} = \boldsymbol{0}, \eta_k^b = \eta_k^c = 0$ for $k \in \{k_1, \ldots, k_1 + k_2 - 1\}$. In words, train the second layer weights a^k for k_2 steps, while keeping the first layer weights fixed at $\boldsymbol{u}^k = \boldsymbol{u}^{k_1}$.

We will take $\eta > 0$ to be a small constant to be specified layer and $\kappa_i \in \text{Unif}([1/2, 3/2])$ be the random perturbation. We also let $b = \Omega(d)$, hiding constants in $\eta, \varepsilon, P, K, \mu$. For the first phase, it suffices to train for $k_1 = P$ time steps. For the second phase, we train for $k_2 = k_2(\eta, \varepsilon, P, K, \mu) = \Theta(1)$ time steps to be specified later as we analyze. Note that only unspecified hyperparameter is the global bias initialization $\bar{c}$ and the results holds almost surely over this choice.

Theorem E.1 (Formal statement of Theorem 7.2). *Assume ℓ is analytic, convex and that Assumption D.4 holds. Then for any $\text{Leap}_{\text{DLQ}_\ell}(\mu) = 1$ setting, for any $\mathcal{D}_{\mu,s_*}^d \in \mathcal{H}_\mu^d$, almost surely over the initialization $\bar{c} \in \mathbb{R}$, and $\boldsymbol{\kappa} \in [1/2, 3/2]^d$, the following holds. For any $\varepsilon > 0$, with*

$$b \geq \Omega(d), M = \Omega(1), \text{ and } k_2 = \Omega(1),$$

using $\Omega(\cdot)$ to hide constants in $\varepsilon, K, P, \mu, \eta, \boldsymbol{\kappa}$ and $\bar{c}$, the above specified layer-wise training dynamics reaches $\overline{\mathcal{R}}(f_{\text{NN}}(\cdot; \boldsymbol{\Theta}^{k_1 + k_2})) \leq \varepsilon$ with probability at least $9/10$.

E.2 Proof of Theorem E.1

The proof of the theorem proceeds by first showing a non-asymptotic approximation guarantee between (d-DF-PDE) and (ℓ-bSGD-g). To this end, we first start by noting a generalization of [Abbe et al., 2022, Theorem 23] to any loss under satisfying Assumption D.4.

Proposition E.2. *Assume that Assumption D.4 on the loss holds. Then for any $0 < \eta \leq K$ and for any $\Upsilon \in \mathbb{N}$, there exists a constant $C(K, \Upsilon)$ such that for (ℓ-bSGD-g) and (d-DF-PDE) of the specified layer-wise training dynamics, with probability $1 - 1/M$, we have*

$$\inf_{k=0,\ldots,\Upsilon} \left| \mathcal{R}(\hat{f}_{\text{NN}}(\cdot; \boldsymbol{\Theta}^k)) - \mathcal{R}(\hat{f}_{\text{NN}}(\cdot; \bar{\rho}_k)) \right| \leq C(K, \Upsilon) \left(\sqrt{\frac{P + \log d}{d}} + \sqrt{\frac{\log M}{M}} + \sqrt{\frac{d + \log M}{b}} \right)$$

The proof of the proposition follows the similar arguments like propagation of chaos used in Theorem D.5. A direct corollary of this proposition is that it suffices to focus only on the (d-DF-PDE) to show Theorem E.1.

Corollary E.3. *Assume that Assumption D.4 holds. Then for total Υ number of (ℓ-bSGD-g) iterations on a neural network with $M \geq C_0(K, \Upsilon, \varepsilon)$ and the batch-size $b \geq C_1(K, \Upsilon, \varepsilon) \cdot d$, the aforementioned layer-wise training dynamics, with probability at least $9/10$*

$$\inf_{k=0,\ldots,\Upsilon} \left| \mathcal{R}(\hat{f}_{\text{NN}}(\cdot; \boldsymbol{\Theta}^k)) - \mathcal{R}(\hat{f}_{\text{NN}}(\cdot; \bar{\rho}_k)) \right| \leq \frac{\varepsilon}{2}.$$

Proof. It is easy to see that it is possible to choose $M \geq C_0(K, \Upsilon, \varepsilon)$ and $b \geq C_1(K, \Upsilon, \varepsilon)d$ for sufficiently large constants such that the right hand side of equality in Proposition E.2 is bounded by $\varepsilon/2$ and $1/M \leq 1/10$. The corollary then follows from Proposition E.2. $\qquad\square$

In light of Corollary E.3, it suffices to simply focus on (d-DF-PDE) and show that it achieves the excess error of at most $\varepsilon/2$ within total steps $\Upsilon = O(1)$ hiding constants in K, μ, P, ε. This will be done by analyzing each phase separately—(i) showing that the training of the top layer weights for steps $k \in \{k_1, \ldots, k_1 + k_2 - 1\}$ in Phase 2 is a linear model trained with a convex loss so it succeeds in reaching a vanishing risk as long as the corresponding kernel matrix of the feature map is non-degenerate; (ii) indeed showing that the corresponding kernel matrix of the feature map after $k_1 = P$ steps of Phase 1 training on the first layer weights is non-degenerate.

Simplified dynamics. We first make a simple observation that $b^k = 0, s^k = 0$ and $c^k = \bar{c}$ throughout; this allows us to analyze a simpler dynamics. To see this, for (b^k, c^k), we use $\eta^c = \eta^b = 0$ throughout, and thus, they remain the same from the initialization. We start with $s^0 = 0$ and using (d-DF-PDE) (assuming $s^k = 0$), gives us

$$s^{k+1} = 0 - \eta_k^w \left(\mathbb{E}_{\boldsymbol{z}, G}[\ell'(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}_k), y) a^k \sigma'(\langle \boldsymbol{u}^k, \boldsymbol{z} \rangle + b^k) G] + \lambda^w 0 \right) = 0.$$

E.3 Phase 2 (linear training)

We first analyze a slightly simpler Phase 2 of training. For a convex loss ℓ, the second layer training just corresponds to linear dynamics with kernel $\boldsymbol{K}^{k_1} : \{+1, -1\}^P \times \{+1, -1\}^P \to \mathbb{R}$ given by

$$\boldsymbol{K}^{k_1}(\boldsymbol{z}, \boldsymbol{z}') = \mathbb{E}_{a \sim \mu_a}[\sigma(\langle \boldsymbol{u}^{k_1}(a), \boldsymbol{z} \rangle) \sigma(\langle \boldsymbol{u}^{k_1}(a), \boldsymbol{z}' \rangle)]. \tag{kernel}$$

We first show some helper lemmas, and using them, show the main lemma. We will then show the proof of the helper lemmas. We start by proving the existence of a sparse function with a small excess error.

Lemma E.4. *For any $\varepsilon > 0$, there exists a function $f_* : \{\pm 1\}^d \to \mathbb{R}$ such that*

$$\overline{\mathcal{R}}(f_*) = \left[\mathcal{R}(f_*) - \inf_{\bar{f} : \{\pm 1\}^d \to \mathbb{R}} \mathcal{R}(\bar{f}) \right] \leq \frac{\varepsilon}{4}. \tag{74}$$

Moreover, $f_*(\boldsymbol{x}) = h_*(\boldsymbol{z})$ for some $h_* : \{\pm 1\}^P \to \mathbb{R}$ indpendent of d, where $\boldsymbol{z}$ is the support of $\mathcal{D}_{s_*}^d$.

Approximation. The neural network output after training $\boldsymbol{u}$ for k_1 steps is given by:

$$\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}_{k_1}) = \bar{c} + \int a \sigma(\langle \boldsymbol{u}^{k_1}(a), \boldsymbol{z} \rangle) \mathrm{d}\mu_a = \bar{c} + \mathbb{E}_{a \sim \mu_a} \left[a \sigma(\langle \boldsymbol{u}^{k_1}(a), \boldsymbol{z} \rangle) \right].$$

Observing that for the next k_2 steps, only a is trained while keeping $\boldsymbol{u}^{k_1}$ fixed, the neural network output is of the following form

$$\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}) = \bar{c} + \int a \sigma(\langle \boldsymbol{u}, \boldsymbol{z} \rangle) \mathrm{d}\bar{\rho}(a, \boldsymbol{u}) = \bar{c} + \int a \sigma(\langle \boldsymbol{u}, \boldsymbol{z} \rangle) \mathrm{d}\bar{\rho}(a(\boldsymbol{u})) \mathrm{d}\bar{\rho}_{k_1}(\boldsymbol{u}). \tag{75}$$

now start by showing that our neural network, there exists $\bar{\rho}_*$ such that we can exactly represent $h_*(\boldsymbol{z})$ as long as the resultant (kernel) after 1$^{\text{st}}$ layer training is non-degenerate.

Lemma E.5. *If $\lambda_{\min}(\boldsymbol{K}^{k_1}) \geq c$ for some constant c, then there exists $\bar{\rho}_* \in \mathcal{P}(\mathbb{R}^{P+1})$ such that $\mathrm{d}\bar{\rho}_*(a, \boldsymbol{u}) = \mathrm{d}\bar{\rho}(a_*(\boldsymbol{u})) \mathrm{d}\bar{\rho}_{k_1}(\boldsymbol{u})$ and for all $\boldsymbol{z} \in \{\pm 1\}^P$*

$$h_*(\boldsymbol{z}) = \hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}_*) = \bar{c} + \int a_* \sigma(\langle \boldsymbol{u}, \boldsymbol{z} \rangle) \mathrm{d}\bar{\rho}_*(a_*(\boldsymbol{u})) \mathrm{d}\bar{\rho}_{k_1}(\boldsymbol{u}).$$

Moreover, $\|a^*\|_{\bar{\rho}_*}^2 = \int a_*^2(\boldsymbol{u}) \mathrm{d}\bar{\rho}_*(a_*(\boldsymbol{u})) \leq c_1$ for some constant $c_1 := c_1(\lambda_{\min}(\boldsymbol{K}^{k_1}), h_*, \bar{c})$.

Convexity and Smoothness. Finally, we will show that the risk objective in terms of $\bar{\rho}$ (of the form (75)) is convex and smooth in terms of $\bar{\rho}$. The convexity follows from the convexity of $\ell(\cdot, y)$ and observing that the neural network output in (75) is linear in $\bar{\rho}(a(\boldsymbol{u}))$. The smoothness follows from Assumption D.4.

28

Lemma E.6. *Consider $\ell : \mathbb{R} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ such that $\ell(\cdot, y)$ is convex for every $y \in \mathcal{Y}$ and that Assumption D.4 holds. Then*

$$\mathcal{R}(\bar{\rho}) = \mathbb{E}[\ell(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}), y)], \quad \text{where } \mathrm{d}\bar{\rho}(a, \boldsymbol{u}) = \mathrm{d}\bar{\rho}(a(\boldsymbol{u}))\mathrm{d}\bar{\rho}_{k_1}(\boldsymbol{u}),$$

is convex and H-smooth with $H = K^3$.

While we denote $\mathcal{R}(\bar{\rho})$ as an objective in terms of the joint measure $\bar{\rho}(a, \boldsymbol{u})$, it is actually an objective in $\bar{\rho}(a(\boldsymbol{u})))$ since it is of the form where $\mathrm{d}(a, \boldsymbol{u}) = \mathrm{d}\bar{\rho}(a(\boldsymbol{u}))\mathrm{d}\bar{\rho}_{k_1}(\boldsymbol{u})$. Also, the convexity and smoothness is in terms of the argument $\bar{\rho}(a(\boldsymbol{u}))$. Having established the convexity and smoothness of the 2nd layer weights training, we will directly apply the convergence rate guarantees for convex and smooth objectives.

Lemma E.7. *Consider any convex, differentiable objective $f(x)$ that is H-smooth. Then for any step-size $\eta \leq \frac{1}{H}$, the gradient descent from initialization x^0 reaches an iterate x^k after k steps such that for any $\tilde{x}$ (not necessarily a minimizer), we have*

$$f(x^T) - f(\tilde{x}) \leq \frac{\|x^0 - \tilde{x}\|^2}{2\eta k}.$$

This is a classical convergence rate guarantee for gradient descent on convex smooth objective [Bubeck et al., 2015, Theorem 3.3], but noting that a proof generalizes even when $\tilde{x}$ is not necessarily a minimizer. We will apply this guarantees with $\tilde{x}$ as a near minimizer of f, especially when the actual minimizer may not exist (for example with the logistic loss, the infimum is never achieved). We provide the proof with other lemmas for completeness.

Lemma E.8. *Assume that $\lambda_{\min}(\boldsymbol{K}^{k_1}) > c > 0$ for some constant c, where $\boldsymbol{K}^{k_1}$ is the (kernel) matrix. Then discrete dimension-free dynamics (d-DF-PDE) in Phase 2 of layer-wise training with step-size $\eta \leq \frac{1}{K^3}$ as specified before after $k_2 := k_2(\eta, \varepsilon, P, K, \mu, \bar{c})$ steps achieves*

$$\overline{\mathcal{R}}(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}^{k_1+k_2})) = \left[\mathcal{R}(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}^{k_1+k_2})) - \inf_{\bar{f}:\{\pm 1\}^d \to \mathbb{R}} \mathcal{R}(\bar{f})\right] \leq \frac{\varepsilon}{2}.$$

Proof. First of all, using Lemma E.4, there exists $\bar{\rho}_*$ such that

$$\overline{\mathcal{R}}(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}_*)) = \mathcal{R}(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}_*)) - \inf_{\bar{f}:\{\pm 1\}^d \to \mathbb{R}} \mathcal{R}(\bar{f}) \leq \frac{\varepsilon}{4}. \tag{76}$$

Moreover, by Lemma E.6, the objective $\mathcal{R}(\bar{\rho})$ is convex and K^3-smooth, and observe that (d-DF-PDE) are performing gradient descent with step-size $\eta_k^a = \eta \leq \frac{1}{K^3}$ on this objective. Thus, using Lemma E.7

$$\left[\mathcal{R}(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}^{k_1+k})) - \mathcal{R}(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}_*))\right] := \left[\mathcal{R}(\bar{\rho}^{k_1+k}) - \mathcal{R}(\bar{\rho}_*)\right] \leq \frac{\|a^{k_1} - a^*\|^2_{\bar{\rho}_{k_1}, \bar{\rho}_*}}{2\eta k}. \tag{77}$$

Recall that $a^{k_1} \sim \mu_a \equiv \mathrm{Unif}([-1, 1])$.

$$\|a^{k_1} - a^*\|^2_{\bar{\rho}_{k_1}, \bar{\rho}_*} = \int (a - a^*(\boldsymbol{u}))^2 \mathrm{d}\mu_a \mathrm{d}\bar{\rho}_*(a^*(\boldsymbol{u}))\mathrm{d}\bar{\rho}_{k_1}(\boldsymbol{u}) \leq 1 + 2\|a^*\|_{\bar{\rho}_*} + \|a^*\|^2_{\bar{\rho}_*} \leq c_2,$$

for some constant $c_2 := c_2(\lambda_{\min}(\boldsymbol{K}^{k_1}), P, \mu, \bar{c})$. Therefore, choosing $k_2 \geq \frac{2c_2}{\eta\varepsilon}$

$$\left[\mathcal{R}(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}^{k_1+k_2})) - \mathcal{R}(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}_*))\right] \leq \frac{\varepsilon}{4}. \tag{78}$$

Combining (78) and (76)

$$\overline{\mathcal{R}}(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}^{k_1+k_2})) = \left[\mathcal{R}(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}^{k_1+k_2})) - \mathcal{R}(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}_*))\right]$$

$$+ \left[\mathcal{R}(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}_*)) - \inf_{\bar{f}:\{\pm 1\}^d \to \mathbb{R}} \mathcal{R}(\bar{f})\right]$$

$$\leq \frac{\varepsilon}{2}.$$

$\square$

Thus, if we can show that $\lambda_{\min}(\boldsymbol{K}^{k_1}) \geq c > 0$ for some constant $c := c(\eta, P, K, \mu)$ for a sufficiently small constant $\eta > 0$ that only depends on K, P, μ, then it immediately implies the desired result by Corollary E.3 with $\Upsilon = k_1 + k_2$. The goal of Appendix E.4 is to show that this holds after $k_1 = P$ steps of the first layer weight training. We now return to the deferred proofs of helper lemmas.

E.3.1 Proof of helper lemmas

Proof of Lemma E.4. Note that Eq. (74) simply follows from the definition of infimum and the excess risk functional $\overline{\mathcal{R}}(\cdot)$. To show that $f_*(\boldsymbol{x}) = h_*(\boldsymbol{z})$, we make an observation that for any function $f : \{\pm 1\}^d \to \mathbb{R}$, one can define $h_* : \{\pm 1\}^P \to \mathbb{R}$ that only depend on the support and achieve risk no worse than f_*. Define $h_*(\boldsymbol{z}) = \mathbb{E}_{\boldsymbol{z}^c}[f_*(\boldsymbol{x})]$, where $\boldsymbol{z}^c = \boldsymbol{x} \setminus \boldsymbol{z}$ (the coordinates outside the support). In other words, h_* is the boolean function after ignoring the monominals that do not depend on $\boldsymbol{z}$. Then

$$\mathcal{R}(f_*) = \mathbb{E}_{(y,\boldsymbol{x}) \sim \mathcal{D}^d_{s_*}}[\ell(f_*(\boldsymbol{x}), y)] = \mathbb{E}_{\mu_{y,\boldsymbol{z}}}[\mathbb{E}_{\mu_{\boldsymbol{z}^c|y,\boldsymbol{z}}}[\ell(f_*(\boldsymbol{x}), y)]]$$

$$\geq \mathbb{E}_{\mu_{y,\boldsymbol{z}}}[\ell(\mathbb{E}_{\mu_{\boldsymbol{z}^c|y,\boldsymbol{z}}}[f_*(\boldsymbol{x})], y)] = \mathbb{E}[\ell(h_*(\boldsymbol{z}), y)] = \mathcal{R}(h),$$

where the only inequality follows from Jensen's inequality and convexity of $\ell(\cdot, y)$. $\qquad\square$

Proof of Lemma E.5. Define $g_*(\boldsymbol{z}) = h_*(\boldsymbol{z}) - \bar{c}$. Our goal is to show the existence of $\bar{\rho}_*$ such that
$$g_*(\boldsymbol{z}) = \mathbb{E}_{(a_*,\boldsymbol{u}) \sim \bar{\rho}_*}[a_* \sigma(\langle \boldsymbol{u}, \boldsymbol{z}\rangle)].$$
Consider the following objective least square objective for the domain

$$\mathcal{L}(a(\boldsymbol{u}); \lambda) = \sum_{\boldsymbol{z} \in \{\pm 1\}^P} \left(\mathbb{E}_{\boldsymbol{u} \sim \bar{\rho}_{k_1}(\boldsymbol{u})}[a(\boldsymbol{u})\sigma(\langle \boldsymbol{u}, \boldsymbol{z}\rangle)] - g_*(\boldsymbol{z})\right)^2 + \lambda \int a^2(\boldsymbol{u}) \mathrm{d}\bar{\rho}_{k_1}(\boldsymbol{u})$$

While this is an infinite dimensional problem, the Representer's theorem holds and the interpolating solution exists. We refer the reader to [Celentano et al., 2021] for a detailed analysis of interpolation with the random feature model. Moreover,

$$\|a^*\|_{\bar{\rho}_*} \leq \lambda_{\min}(\boldsymbol{K}^{k_1})^{-1}\sqrt{\sum_{\boldsymbol{z} \in \{\pm 1\}^P} g_*(\boldsymbol{z})^2} \leq c_1,$$

for some constant c_1 that only depends on $\lambda_{\min}(\boldsymbol{K}^{k_1}), h_*, P$ and $\bar{c}$. $\qquad\square$

Proof of Lemma E.6. For $t \in [0, 1]$ and any $\bar{\rho}^{(1)}$ and $\bar{\rho}^{(2)}$, consider the density $\bar{\rho} = t\bar{\rho}^{(1)} + (1-t)\bar{\rho}^{(2)}$. Then

$$\mathcal{R}(\bar{\rho}) = \mathcal{R}(t\bar{\rho}^{(1)} + (1-t)\bar{\rho}^{(2)}) = \mathbb{E}[\ell(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; t\bar{\rho}^{(1)} + (1-t)\bar{\rho}^{(2)}), y)]$$

$$= \mathbb{E}[\ell(t\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}^{(1)}) + (1-t)\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}^{(2)}), y)]$$

$$\leq \mathbb{E}[t\ell(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}^{(1)}), y) + (1-t)\ell(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}^{(2)}), y)] \qquad \text{(using convexity)}$$

$$= t\mathbb{E}\left[\ell(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}^{(1)}), y)\right] + (1-t)\mathbb{E}\left[\ell(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}^{(2)}), y)\right] = t\,\mathcal{R}(\bar{\rho}^{(1)}) + (1-t)\mathcal{R}(\bar{\rho}^{(2)}).$$

To show smoothness, first observe that a direct consequence of Assumption D.4 (the condition $|\ell''(u, y)| \leq K$) is that for any $u_1, u_2 \in \mathbb{R}$, we have

$$\ell(u_1, y) \leq \ell(u_2, y) + \ell'(u_2, y)(u_1 - u_2) + \frac{K}{2}(u_2 - u_1)^2. \tag{79}$$

Using this, for any $\bar{\rho}^{(1)}$ and $\bar{\rho}^{(2)}$

$$\mathcal{R}(\bar{\rho}^{(1)}) = \mathbb{E}[\ell(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}^{(1)}), y)]$$

$$\leq \mathbb{E}[\ell(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}^{(2)}), y)] + \mathbb{E}\left[\ell'(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}^{(2)}), y)\left(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}^{(1)}) - \hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}^{(2)})\right)\right]$$

$$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(using (79))}$$

$$+ \frac{K}{2}\mathbb{E}\left[\left(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}^{(1)}) - \hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}^{(2)})\right)^2\right] \tag{80}$$

Since $\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}^{(1)}) - \hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}^{(2)}) = \mathbb{E}_{a^{(1)},a^{(2)},\boldsymbol{u}}\left[(a^{(1)} - a^{(2)})\sigma(\langle \boldsymbol{u}, \boldsymbol{z}\rangle)\right]$, further simplifying the above

$$\mathbb{E}\left[\ell'(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}^{(2)}), y)\left(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}^{(1)}) - \hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}^{(2)})\right)\right]$$

$$= \mathbb{E}_{(y,\boldsymbol{z}) \sim \mu_{y,\boldsymbol{z}}}\left[\ell'(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}^{(2)}), y)\mathbb{E}_{a^{(1)},a^{(2)},\boldsymbol{u}}\left[(a^{(1)} - a^{(2)})\sigma(\langle \boldsymbol{u}, \boldsymbol{z}\rangle)\right]\right]$$

$$= \mathbb{E}_{a^{(1)},a^{(2)},\boldsymbol{u}}\left[\mathbb{E}_{(y,\boldsymbol{z}) \sim \mu_{y,\boldsymbol{z}}}\left[\ell'(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}^{(2)}), y)\sigma(\langle \boldsymbol{u}, \boldsymbol{z}\rangle)\right](a^{(1)} - a^{(2)})\right]$$

$$= \left\langle \nabla_a \mathcal{R}(\bar{\rho}^{(2)}), a^{(1)} - a^{(2)}\right\rangle_{\bar{\rho}^{(1)}, \bar{\rho}^{(2)}}. \tag{81}$$

Also, simplifying the third term of (80)

$$\frac{K}{2}\mathbb{E}\left[\left(\hat{f}_{\mathsf{NN}}(\boldsymbol{z};\bar{\rho}^{(1)}) - \hat{f}_{\mathsf{NN}}(\boldsymbol{z};\bar{\rho}^{(2)})\right)^2\right] = \frac{K}{2}\mathbb{E}_{(y,\boldsymbol{z})\sim\mu_{y,\boldsymbol{z}}}\left[\mathbb{E}_{a^{(1)},a^{(2)},\boldsymbol{u}}\left[\left(a^{(1)} - a^{(2)}\right)\sigma(\langle\boldsymbol{u},\boldsymbol{z}\rangle)\right]^2\right]$$

$$\leq \frac{K}{2}\mathbb{E}_{(y,\boldsymbol{z})\sim\mu_{y,\boldsymbol{z}}}\mathbb{E}_{a^{(1)},a^{(2)},\boldsymbol{u}}\left[(a^{(1)} - a^{(2)})^2\sigma^2(\langle\boldsymbol{u},\boldsymbol{z}\rangle)\right] \qquad \text{(by Jensen's ineqaulity)}$$

$$\leq \frac{K^3}{2}\|a^{(1)} - a^{(2)}\|^2_{\bar{\rho}^{(1)},\bar{\rho}^{(2)}}. \tag{82}$$

Finally, combining (81) and (82) with (80), we obtain

$$\mathcal{R}(\bar{\rho}^{(1)}) = \mathcal{R}(\bar{\rho}^{(2)}) + \left\langle\nabla_a\mathcal{R}(\bar{\rho}^{(2)}), a^{(1)} - a^{(2)}\right\rangle_{\bar{\rho}^{(1)},\bar{\rho}^{(2)}} + \frac{K^3}{2}\|a^{(1)} - a^{(2)}\|^2_{\bar{\rho}^{(1)},\bar{\rho}^{(2)}},$$

which, by definition, gives us H-smoothness with $H = K^3$. $\qquad\square$

Proof of Lemma E.7. First, the classical descent lemma holds due to H-smoothness for step-size $\eta \leq \frac{1}{H}$.

$$f(x^{j+1}) \leq f(x^j) + \langle\nabla f(x^j), x^{j+1} - x^j\rangle + \frac{H}{2}\|x^{j+1} - x^j\|^2_2$$

$$= f(x^j) - \eta\left(1 - \frac{\eta H}{2}\right)\|\nabla f(x^j)\|^2_2 \leq f(x^j) - \frac{\eta}{2}\|\nabla f(x^j)\|^2_2 \leq f(x^j). \tag{83}$$

By convexity, for any $\tilde{x}$ (not necessarily a minimizer), we also have

$$f(x^j) \leq f(\tilde{x}) + \langle\nabla f(x^j), x^j - \tilde{x}\rangle.$$

Substituting this in (83)

$$f(x^{j+1}) \leq f(\tilde{x}) + \langle\nabla f(x^j), x^j - \tilde{x}\rangle - \frac{\eta}{2}\|\nabla f(x^j)\|^2_2$$

$$= f(\tilde{x}) + \frac{1}{2\eta}\left(2\eta\langle\nabla f(x^j), x^j - \tilde{x}\rangle - \eta^2\|\nabla f(x^j)\|^2_2\right)$$

$$= f(\tilde{x}) + \frac{1}{2\eta}\left(2\eta\langle\nabla f(x^j), x^j - \tilde{x}\rangle - \eta^2\|\nabla f(x^j)\|^2_2 - \|x^j - \tilde{x}\|^2_2 + \|x^j - \tilde{x}\|^2_2\right)$$

$$= f(\tilde{x}) + \frac{1}{2\eta}\left(\|x^j - \tilde{x}\|^2_2 - \|x^{j+1} - \tilde{x}^2\|^2_2\right).$$

Therefore, summing over $j \in \{0,\ldots,k-1\}$, since the right hand side is a telescopic sum

$$\sum_{j=0}^{k-1}\left(f(x^{j+1}) - f(\tilde{x})\right) \leq \frac{1}{2\eta}\left(\|x^0 - \tilde{x}\|^2_2 - \|x^k - \tilde{x}\|^2_2\right) \leq \frac{\|x^0 - \tilde{x}\|^2_2}{2\eta}.$$

Note that the lemma statements trivially holds if $f(x^k) - f(\tilde{x}) \leq 0$ already. Therefore, in the case when $f(x^k) > f(\tilde{x})$ and using the descent lemma, we finally conclude the proof.

$$f(x^k) - f(\tilde{x}) \leq \frac{1}{k}\sum_{j=0}^{k-1}\left(f(x^j) - f(\tilde{x})\right) \leq \frac{\|x^0 - \tilde{x}\|^2}{2\eta k}.$$

$\qquad\square$

E.4 Phase 1 (non-linear training)

We now analyze Phase 1 of the algorithm and show that $\lambda_{\min}(\boldsymbol{K}^{k_1}) \geq c$, for a constant $c := c(\eta, \mu, P, K)$ where η is also a small constant (in terms of μ, P, K).

Writing the weight evolution with a polynomial. We will by considering a general polynomial activation $\sigma(x) = \sum_{l=0}^{L} m_l x^l$ of degree L, whose coefficients are given by $\boldsymbol{m} = (m_0, \ldots, m_L)$.

Lemma E.9. *(Training dynamics given by a polynomial) Let $\boldsymbol{\xi} = (\xi_{S,k})_{S \subseteq [P], 0 \leq k \leq k_1 - 1} \in \mathbb{R}^{2^P k_1}$, $\zeta \in \mathbb{R}$, $\boldsymbol{\rho} \in \mathbb{R}^{L+1}, \boldsymbol{\gamma} \in \mathbb{R}^P$ be variables. For each $i \in [P]$, define $p_{0,i}(\zeta, \boldsymbol{\xi}, \boldsymbol{\rho}, \boldsymbol{\gamma}) \equiv 0$. For each, $0 \leq k \leq k_1 - 1$,*

$$p_{k+1,i}(\zeta, \boldsymbol{\xi}, \boldsymbol{\rho}, \boldsymbol{\gamma}) = p_{k,i}(\zeta, \boldsymbol{\xi}, \boldsymbol{\rho}, \boldsymbol{\gamma}) + \zeta \gamma_i \rho_1 \boldsymbol{\xi}_{\{i\},k}$$

$$+ \zeta \gamma_i \sum_{r=1}^{L-1} \frac{\rho_{r+1}}{r!} \sum_{(i_1, \ldots, i_r) \in [P]^r} \xi_{i \oplus i_1 \oplus \cdots \oplus i_r, k} \prod_{l=1}^{r} p_{k, i_l}(\zeta, \boldsymbol{\xi}, \boldsymbol{\rho}, \boldsymbol{\gamma}).$$

Then there is a constant $c := c(k_1, P, K)$ such that for any $0 < \eta < c$

$$u_i^k(a) = p_{k,i}(\eta a, \boldsymbol{\beta}, \boldsymbol{m}, \boldsymbol{\kappa}),$$

and $\boldsymbol{\beta} = (\beta_{S,k})_{S \subseteq [P], 0 \leq k \leq k_1 - 1}$ is given by $\beta_{S,k} = -\mathbb{E}[\ell'(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}_k), y) \chi_S(\boldsymbol{z})]$.

The term $\hat{f}_{\mathsf{NN}}(\cdot; \bar{\rho}_k)$ evolves non-linearly, and difficult to analyze directly. However, for sufficiently small step size η, the interaction term $\hat{f}_{\mathsf{NN}}(\cdot; \bar{\rho}_k)$ is small, and we show that it can be ignored. Formally, we define the simplified dynamics $\hat{\boldsymbol{u}}^k(a)$ for each $0 \leq k \leq k_1$ by letting $\hat{\boldsymbol{u}}^0(a) = \boldsymbol{0}$ and inductively setting for each $k \in \{0, \ldots, k_1 - 1\}$,

$$\hat{\boldsymbol{u}}^{k+1}(a) = \hat{\boldsymbol{u}}^k(a) - \eta \boldsymbol{\kappa} \circ \mathbb{E}[\ell'(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}_0), y) \cdot a \cdot \sigma'(\langle \hat{\boldsymbol{u}}^k(a), \boldsymbol{z} \rangle) \boldsymbol{z}].$$

Using a similar argument, we may show:

Lemma E.10 (Simplified training dynamics are given by a polynomial). *There is a constant $c > 0$ depending only on k_1, P, K, such that for any $0 < \eta < c$, any $i \in [P]$ and any $0 \leq k \leq k_1$, we have*

$$\hat{u}_i^k(a) = p_{k,i}(\eta a, \boldsymbol{\alpha}, \boldsymbol{m}, \boldsymbol{\kappa}),$$

where we abuse notation (since $\boldsymbol{\alpha} = (\alpha_S)_{S \subseteq [P]}$ otherwise) and let $\boldsymbol{\alpha} = (\alpha_{S,k})_{S \subseteq [P], 0 \leq k \leq k_1 - 1}$ be given by

$$\alpha_{S,k} = \alpha_S = \mathbb{E}[-\ell'(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}_0), y) \chi_S(\boldsymbol{z})]$$

Reducing to analyzing simplified dynamics We lower-bound $\lambda_{\min}(\boldsymbol{K}^{k_1})$ in terms of the determinant of a certain random matrix. Let $\boldsymbol{\zeta} = [\zeta_1, \ldots, \zeta_{2^P}]$ be a vector of 2^P variables. Define $\boldsymbol{M} = \boldsymbol{M}(\boldsymbol{\zeta}, \boldsymbol{\xi}, \boldsymbol{\rho}, \boldsymbol{\gamma}) \in \mathbb{R}^{2^P \times 2^P}$ to be the matrix indexed by $\boldsymbol{z} \in \{+1, -1\}^P$ and $j \in [2^P]$ with entries

$$M_{\boldsymbol{z},j}(\boldsymbol{\zeta}, \boldsymbol{\xi}, \boldsymbol{\rho}, \boldsymbol{\gamma}) = \sum_{r=0}^{L} \frac{\rho_r}{r!} \left(\sum_{i=1}^{P} z_i p_{k,i}(\zeta_j, \boldsymbol{\xi}, \boldsymbol{\rho}, \boldsymbol{\gamma}) \right)^r.$$

This matrix is motivated by the following fact:

Lemma E.11. *There is a constant $c > 0$ depending only on k_1, P, K, such that for any $0 < \eta < c$, and any $\boldsymbol{a} = [a_1, \ldots, a_{2^P}] \in [-1, 1]^{2^P}$, we have*

$$M_{\boldsymbol{z},j}(\eta \boldsymbol{a}, \boldsymbol{\beta}, \boldsymbol{m}, \boldsymbol{\kappa}) = \sigma(\langle \boldsymbol{u}^{k_1}(a_j), \boldsymbol{z} \rangle)$$

$$M_{\boldsymbol{z},j}(\eta \boldsymbol{a}, \boldsymbol{\alpha}, \boldsymbol{m}, \boldsymbol{\kappa}) = \sigma(\langle \hat{\boldsymbol{u}}^{k_1}(a_j), \boldsymbol{z} \rangle)$$

Under this notation, using the exactly same analysis as in [Abbe et al., 2022, Lemma E.9], we can also show

Lemma E.12. *There is a constant $c > 0$ depending on K, P such that for any $0 < \eta < c$,*

$$\lambda_{\min}(\boldsymbol{K}^{k_1}) \geq c \mathbb{E}_{\boldsymbol{a} \sim \mu_a^{\otimes 2^P}}[\det(\boldsymbol{M}(\eta \boldsymbol{a}, \boldsymbol{\beta}, \boldsymbol{m}, \boldsymbol{\kappa}))^2].$$

On the other hand, we can prove a lower-bound on $\mathbb{E}[\det(\boldsymbol{M}(\eta \boldsymbol{a}, \boldsymbol{\beta}, \boldsymbol{m}, \boldsymbol{\kappa}))^2]$ simply by lower-bounding the sum of magnitudes of coefficients of $\det(\boldsymbol{M}(\boldsymbol{\zeta}, \boldsymbol{\alpha}, \boldsymbol{m}, \boldsymbol{\kappa}))$ when viewed as a polynomial in $\boldsymbol{\zeta}$ almost surely over the choice of $\bar{c}$. This is because of (a) the fact that $\det(\boldsymbol{M}(\boldsymbol{\zeta}, \boldsymbol{\alpha}, \boldsymbol{m}, \boldsymbol{\kappa}))$ and $\det(\boldsymbol{M}(\boldsymbol{\zeta}, \boldsymbol{\beta}, \boldsymbol{m}, \boldsymbol{\kappa}))$ have coefficients in $\boldsymbol{\zeta}$ that are $O(\eta)$-close for η small, and (b) the fact that polynomials anti-concentrate over random inputs.

Analysis of simplified dynamics. The proof is now reduced to showing that $\det(\boldsymbol{M}(\boldsymbol{\zeta}, \boldsymbol{\alpha}, \boldsymbol{m}, \boldsymbol{\kappa})) \not\equiv 0$, as a polynomial in $\boldsymbol{\zeta}$. In other words, by Lemma E.11, we are now focusing on only the simplified dynamics $\hat{\boldsymbol{u}}^k$. If we can show that $\det(\boldsymbol{M}(\boldsymbol{\zeta}, \boldsymbol{\alpha}, \boldsymbol{m}, \boldsymbol{\kappa})) \not\equiv 0$ almost surely over the choice of $\boldsymbol{\kappa} \in [1/2, 3/2]^P$, then Theorem E.1 follows. Therefore, introducing the variables $\boldsymbol{\gamma} \in \mathbb{R}^P$, it suffices to show that

$$\det(\boldsymbol{M}(\boldsymbol{\zeta}, \boldsymbol{\alpha}, \boldsymbol{m}, \boldsymbol{\gamma})) \not\equiv 0, \text{ as a polynomial in } \boldsymbol{\zeta}, \boldsymbol{\gamma}. \tag{84}$$

We start by following simple observation.

Claim 1. *Almost surely over the choice of initialization $\bar{c}$, the set of subsets $\mathcal{K} := \{S : \alpha_S \neq 0\}$ has* $\mathsf{Leap}(\mathcal{K}) = 1$.

Proof. This basically follows from the fact that ℓ is piecewise analytic. The argument is as follows: we have that $\mathsf{Leap}_{\mathsf{DLQ}_\ell} = \mathsf{Leap}(\mathcal{C}_{\mathsf{DLQ}_\ell}) = 1$, where $\mathcal{C}_{\mathsf{DLQ}_\ell} = \{S : \exists u \in \mathbb{R} \text{ such that } \mathbb{E}[\ell'(u, y)\chi_S(\boldsymbol{z})] \neq 0\}$. Since ℓ is piece-wise analytic, so is $\ell'(u, y)$ in the first argument. Thus, for any $S \in \mathcal{C}_{\mathsf{DLQ}_\ell}$, the set $\{u : \mathbb{E}[\ell'(u, y)\chi_S(\boldsymbol{z})] = 0\}$ has measure 0. Thus, almost surely over the choice of $\bar{c} = \hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}_0)$, we have that for any $S \in \mathcal{C}_{\mathsf{DLQ}_\ell}$

$$\mathbb{E}[\ell'(\bar{c}, y)\chi_S(\boldsymbol{z})] = \mathbb{E}[\ell'(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}_0), y)\chi_S(\boldsymbol{z})] = -\alpha_S \neq 0.$$

This implies that $S \in \mathcal{K}$ as well. We finally conclude that $\mathsf{Leap}_{\mathsf{DLQ}_\ell} = \mathsf{Leap}(\mathcal{K}) = 1$. $\qquad\square$

Using ideas similar to Abbe et al. [2022], we now prove (84) by analyzing the recurrence relations for $p_{k,i}$ to show that to first-order the polynomials $p_{k_1,i}$ are distinct for all $i \in [P]$, but now doing smooth analysis over $\boldsymbol{\gamma}$ instead. Formally we show the following lemma.

Lemma E.13. *Suppose that $L \geq 2^{8P}$ and let $m_i = i!\binom{L}{i}$ for all $0 \leq i \leq L$, which correspond to the activation function $\sigma(x) = (1 + x)^L$. Let $k_1 = P$. Then $\det(\boldsymbol{M}(\boldsymbol{\zeta}, \boldsymbol{\alpha}, \boldsymbol{m}, \boldsymbol{\gamma})) \not\equiv 0$, i.e. (84) holds.*

Claim 2. *For any time step $0 \leq k \leq k_1$ any $j \in [N]$, a learning rate $\eta < 1/(4K^2(1 + K)Pk)$, and any $a \in [-1, 1]$ we have*

$$\|\boldsymbol{u}^k(a)\|_1, \|\hat{\boldsymbol{u}}^k(a)\|_1 \leq 2\eta K^2(1 + K)Pk \leq 1/2.$$

Proof. We prove this by induction on k. For $k = 0$, we have $\boldsymbol{u}^0 = \hat{\boldsymbol{u}}^0 = \boldsymbol{0}$ and the claim holds trivially. For the inductive step, $\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}_k) \leq \mathbb{E}_{a \sim \mu_a}[|a||\sigma(\langle \boldsymbol{u}^k, \boldsymbol{z}\rangle)|] \leq \|\sigma\|_\infty \leq K$, since $a \sim \mathsf{Unif}([-1, 1])$. Therefore

$$\|\boldsymbol{u}^{k+1}(a)\|_1 \leq \|\boldsymbol{u}^k(a)\|_1 + \eta\|\boldsymbol{\kappa} \circ \mathbb{E}_{y,\boldsymbol{z}}[-\ell'(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}_k), y)a\sigma'(\langle \boldsymbol{u}^k, \boldsymbol{z}\rangle)\boldsymbol{z}]\|_1$$
$$\leq \|\boldsymbol{u}^k(a)\|_1 + \eta\|\boldsymbol{\kappa}\|_\infty \mathbb{E}_{y,\boldsymbol{z}}[\| - \ell'(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}_k), y)a\sigma'(\langle \boldsymbol{u}^k, \boldsymbol{z}\rangle)\boldsymbol{z}\|_1]$$
$$\leq \|\boldsymbol{u}^k(a)\|_1 + 2\eta K(1 + K) \cdot K \cdot \mathbb{E}_{\boldsymbol{z}}[\|\boldsymbol{z}\|_1] \leq \|\boldsymbol{u}^k(a)\|_1 + 2\eta K^2(1 + K)P \leq 2\eta K^2(1 + K)Pk.$$

Note that here we used $\|\boldsymbol{\kappa}\|_\infty \leq 2$, $|\ell'(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}_k), y)| \leq K(1 + |\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}_k)|) \leq K(1 + K)$ using A2, and $|a\sigma'(\langle \boldsymbol{u}^k, \boldsymbol{z}\rangle)| \leq |a| \cdot |\sigma'(\langle \boldsymbol{u}^k, \boldsymbol{z}\rangle)| \leq K$. The bound for $\|\hat{\boldsymbol{u}}^k(a)\|_1$ follows exactly the same argument but now we use $|\ell'(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}_0), y)| \leq K$. $\qquad\square$

Proof of Lemmas E.9 and E.10. We have $\zeta \in \mathbb{R}$, $\boldsymbol{\xi} = (\xi_{S,k})_{S \subseteq [P], k \in \{0, \ldots, k_1 - 1\}}$, $\boldsymbol{\rho} \in \mathbb{R}^{L+1}$, and $\boldsymbol{\gamma} \in \mathbb{R}^P$ as variables. Let $s_0, \ldots, s_{k_1-1} : \{+1, -1\}^P \to \mathbb{R}$ be $s_k(\boldsymbol{z}) = \sum_{S \subseteq [P]} \xi_{S,k}\chi_S(\boldsymbol{z})$. Consider the recurrence relation $\boldsymbol{\nu}^k \in \mathbb{R}^P$, where we initialize $\boldsymbol{\nu}^0 = \boldsymbol{0}$ and, for $0 \leq k \leq k_1 - 1$,

$$\boldsymbol{\nu}^{k+1} = \boldsymbol{\nu}^k + \zeta\boldsymbol{\gamma} \circ \mathbb{E}_{\boldsymbol{z}}\Big[s_k(\boldsymbol{z})\sum_{r=0}^{L-1}\frac{\rho_{r+1}}{r!}\langle \boldsymbol{\nu}^k, \boldsymbol{z}\rangle^r \boldsymbol{z}\Big]. \tag{85}$$

This recurrence is satisfied by $\zeta = \eta a$, $\boldsymbol{\rho} = \boldsymbol{m}$, $\boldsymbol{\gamma} = \boldsymbol{\kappa}$, and $s_k(\boldsymbol{z}) = \sum_S \beta_{S,k} \chi_S(\boldsymbol{z})$. This is because

$$
\boldsymbol{u}^{k+1} = \boldsymbol{u}^k - \eta \cdot \boldsymbol{\kappa} \circ \mathbb{E}[\ell'(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}_k), y) a \sigma'(\langle \boldsymbol{u}^k, \boldsymbol{z} \rangle) \boldsymbol{z}]
$$

$$
= \boldsymbol{u}^k + \eta a \cdot \boldsymbol{\kappa} \circ \mathbb{E}[-\ell'(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}_k), y) \sum_{r=0}^{L-1} \frac{\boldsymbol{m}_{r+1}}{r!} \langle \boldsymbol{u}^k, \boldsymbol{z} \rangle^r \boldsymbol{z}]
$$

$$
= \boldsymbol{u}^k - \eta \cdot \boldsymbol{\kappa} \circ \mathbb{E}[\ell'(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}_k), y) a \sigma'(\langle \boldsymbol{u}^k, \boldsymbol{z} \rangle) \boldsymbol{z}]
$$

$$
= \boldsymbol{u}^k + \eta a \cdot \boldsymbol{\kappa} \circ \mathbb{E}[-\ell'(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}_k), y) \sum_{r=0}^{L-1} \frac{\boldsymbol{m}_{r+1}}{r!} \langle \boldsymbol{u}^k, \boldsymbol{z} \rangle^r \boldsymbol{z}]
$$

$$
= \boldsymbol{u}^k + \eta a \cdot \boldsymbol{\kappa} \circ \mathbb{E}_{\boldsymbol{z}}[\mathbb{E}_{y|\boldsymbol{z}}[-\ell'(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}_k), y)] \sum_{r=0}^{L-1} \frac{\boldsymbol{m}_{r+1}}{r!} \langle \boldsymbol{u}^k, \boldsymbol{z} \rangle^r \boldsymbol{z}]
$$

Finally, noting that $\mathbb{E}_{y|\boldsymbol{z}}[-\ell'(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}_k), y)] = \sum_S \beta_{S,k} \chi_S(\boldsymbol{z})$, we showed that recurrence (85) holds with specified value of variables. Similarly, $\hat{\boldsymbol{u}}^k(a)$ with $s_k(\boldsymbol{z}) = \sum_S \alpha_S \chi_S(\boldsymbol{z})$ for $\alpha_S = -\mathbb{E}[\ell'(\hat{f}_{\mathsf{NN}}(\boldsymbol{z}; \bar{\rho}_0, y) \chi_S(\boldsymbol{z})]$. This is because $|\langle \boldsymbol{u}^k, \boldsymbol{z} \rangle|, |\langle \hat{\boldsymbol{u}}^k, \boldsymbol{z} \rangle| \leq 1/2 < 1$ by Claim 2 and in the interval $(-1, 1)$, we have $\sigma(x) = \sum_{r=0}^L \frac{m_r}{r!} x^r$. It finally remains to show that

$$
\nu_i^k = p_{k,i}(\zeta, \boldsymbol{\xi}, \boldsymbol{\rho}, \boldsymbol{\gamma}).
$$

The proof is by induction on k. For $k = 0$, it is true that $p_{0,i}(\zeta, \boldsymbol{\xi}, \boldsymbol{\rho}, \boldsymbol{\gamma}) = 0 = \nu_i^0$. For the inductive step, for any $r \geq 1$ and $i \in [d]$, we can write

$$
\gamma_i \cdot \mathbb{E}_{\boldsymbol{z}}[s_k(\boldsymbol{z}) \langle \boldsymbol{\nu}^k, \boldsymbol{z} \rangle^r z_i] = \gamma_i \mathbb{E}_{\boldsymbol{z}} \Big[s_k(\boldsymbol{z}) z_i \sum_{(i_1, \ldots, i_r) \in [P]^r} \prod_{l=1}^r \nu_{i_l}^k z_{i_l} \Big]
$$

$$
= \gamma_i \sum_{(i_1, \ldots, i_r) \in [P]^r} \mathbb{E}_{\boldsymbol{z}} \Big[s_k(\boldsymbol{z}) \chi_i(\boldsymbol{z}) \prod_{l=1}^r \chi_{i_l}(\boldsymbol{z}) \Big] \prod_{l=1}^r p_{k,i_l}(\zeta, \boldsymbol{\xi}, \boldsymbol{\rho}, \boldsymbol{\gamma})
$$

$$
= \gamma_i \sum_{(i_1, \ldots, i_r) \in [P]^r} \xi_{\{i\} \oplus \{i_1\} \oplus \cdots \oplus \{i_r\}, k} \prod_{l=1}^r p_{k,i_l}(\zeta, \boldsymbol{\xi}, \boldsymbol{\rho}, \boldsymbol{\gamma}),
$$

and $\gamma_i \mathbb{E}_{\boldsymbol{z}}[s_k(\boldsymbol{z}) \langle \boldsymbol{\nu}^k, \boldsymbol{z} \rangle^0 z_i] = \gamma_i \mathbb{E}_{\boldsymbol{z}}[s_k(\boldsymbol{z}) z_i] = \gamma_i \xi_{\{i\}, k}$. The inductive step follows by linearity of expectation. $\qquad \square$

E.5 Proof of Lemma E.13

In Lemma E.13, we have already fixed $\boldsymbol{m} \in \mathbb{R}^{L+1}$ to be $m_i = i! \binom{L}{i}$ for all $i \in \{0, \ldots, L\}$. This corresponds to the activation function $\sigma(x) = (1 + x)^L$.

E.5.1 Reducing to minimal Leap 1 structure

To show that $\det(\boldsymbol{M}(\boldsymbol{\zeta}, \boldsymbol{\alpha}, \boldsymbol{m}, \boldsymbol{\gamma})) \not\equiv 0$ as a polynomial in $\boldsymbol{\zeta}, \boldsymbol{\gamma}$, we first show that it suffices to consider "minimal" Leap 1 set structure, without loss of generality.

Claim 3. *Let $\mathcal{K}' \subseteq \mathcal{K}$ such that $\mathsf{Leap}(\mathcal{K}') = 1$. Then if*

$$
\det(\boldsymbol{M}(\boldsymbol{\zeta}, \boldsymbol{\alpha}, \boldsymbol{m}, \boldsymbol{\gamma})) \mid_{\alpha_S = 0 \text{ for all } \alpha \in \mathcal{K} \backslash \mathcal{K}'} \not\equiv 0, \text{ as a polynomial of } \boldsymbol{\zeta}, \boldsymbol{\gamma}
$$

we have

$$
\det(\boldsymbol{M}(\boldsymbol{\zeta}, \boldsymbol{\alpha}, \boldsymbol{m}, \boldsymbol{\gamma})) \not\equiv 0 \text{ as a polynomial of } \boldsymbol{\zeta}, \boldsymbol{\gamma}.
$$

Proof. Directly substituting $\alpha_S = 0$ for all $S \in \mathcal{K} \setminus \mathcal{K}'$. $\qquad \square$

Therefore, without loss of generality (up to relabelling the indices of the variables), we assume that

$$
\mathcal{K} = \{S_1, \ldots, S_P\}, \text{ where, for all } i \in [P], i \in S_i \text{ and } S_i \subseteq [i].
$$

Otherwise, we could remove a set from $\mathcal{K}$ while still having $\mathsf{Leap}(\mathcal{K}) = 1$.

E.5.2 Weights to leading order

Let us define the polynomials $q_{k,i}$ in variables $\boldsymbol{\zeta}, \boldsymbol{\phi}, \boldsymbol{\rho}, \boldsymbol{\gamma}$ where $\boldsymbol{\phi} = (\phi_S)_{S \in \mathcal{K}}$. For all $k \in \{0, \ldots, k_1 - 1\}$ and $i \in [P]$,

$$q_{k,i}(\boldsymbol{\zeta}, \boldsymbol{\phi}, \boldsymbol{\rho}, \boldsymbol{\gamma}) = p_{k,i}(\boldsymbol{\zeta}, \boldsymbol{\xi}, \boldsymbol{\rho}, \boldsymbol{\gamma}) \mid_{\xi_{S,k}=0 \text{ for all } S \notin \mathcal{K} \text{ and } \xi_{S,k}=\phi_S \text{ for all } S \in \mathcal{K}} \cdot$$

Therefore $\boldsymbol{M}(\boldsymbol{\zeta}, \boldsymbol{\phi}, \boldsymbol{\rho}, \boldsymbol{\gamma})$ has entries $M_{z,j}(\boldsymbol{\zeta}, \boldsymbol{\phi}, \boldsymbol{\rho}, \boldsymbol{\gamma}) = \sum_{r=0}^{L} \frac{\rho_r}{r!} \left(\sum_{i=1}^{P} q_{k,i}(\zeta_j, \boldsymbol{\phi}, \boldsymbol{\rho}, \boldsymbol{\gamma}) \right)^r$. We will explicitly compute the lowest degree term in ζ. First, we show that many terms are zero. To this end, consider the following notation: Recursively define

$$o_i = 1 + \sum_{i' \in S_i \setminus \{i\}} o_{i'} \text{ for all } i \in [P],$$

where the sum over an empty set is 0 by convention. Additionally, let $\tilde{q}_{k,i}(\boldsymbol{\phi}, \boldsymbol{\rho}, \boldsymbol{\gamma})$ to be the coefficient of the term in $q_{k,i}(\boldsymbol{\zeta}, \boldsymbol{\phi}, \boldsymbol{\rho}, \boldsymbol{\gamma})$ whose degree in ζ is o_i. Furthermore, recursively define

$$g_i(\boldsymbol{\gamma}) = \gamma_i \text{ if } \{i\} \in \mathcal{K}, \text{ and otherwise } g_i(\boldsymbol{\gamma}) = \gamma_i \prod_{i' \in S_i \setminus \{i'\}} g_{i'}(\boldsymbol{\gamma}).$$

Lemma E.14. *Under the above notation, we have that $q_{k,i}(\boldsymbol{\zeta}, \boldsymbol{\phi}, \boldsymbol{m}, \boldsymbol{\gamma})$ has no nonzero terms of degree less than o_i in ζ. Furthermore, $\tilde{q}_{k,i}(\boldsymbol{\phi}, \boldsymbol{\rho}, \boldsymbol{\gamma})$ can be decomposed as $\tilde{q}_{k,i}(\boldsymbol{\phi}, \boldsymbol{\rho}, \boldsymbol{\gamma}) = g_i(\boldsymbol{\gamma}) \cdot \hat{q}_{k,i}(\boldsymbol{\phi}, \boldsymbol{\rho})$ for some $\hat{q}_{k,i}(\boldsymbol{\phi}, \boldsymbol{\rho})$.*

Proof. We will prove this by induction on k. The base case for $k = 0$ trivially holds because $q_{0,i} \equiv 0$. For the inductive step, we assume the statement holds for all $k' \in \{0, \ldots, k\}$ and we prove the statement for $k + 1$. By the recurrence relation of the polynomials $q_{k,i}$, we have

$$q_{k+1,i}(\boldsymbol{\zeta}, \boldsymbol{\phi}, \boldsymbol{\rho}, \boldsymbol{\gamma}) = q_{k,i}(\boldsymbol{\zeta}, \boldsymbol{\phi}, \boldsymbol{\rho}, \boldsymbol{\gamma}) + \gamma_i \zeta \rho_1 \phi_{\{i\}} \mathbb{1}(\{i\} \in \mathcal{K})$$

$$+ \gamma_i \zeta \sum_{r=1}^{L-1} \frac{\rho_{r+1}}{r!} \sum_{(i_1, \ldots, i_r) \in [P]^r} \phi_{\{i\} \oplus \{i_1\} \oplus \cdots \oplus \{i_r\}} \mathbb{1}(\{i\} \oplus \{i_1\} \oplus \cdots \oplus \{i_r\} \in \mathcal{K}) \prod_{l=1}^{r} q_{k,i_l}(\boldsymbol{\zeta}, \boldsymbol{\phi}, \boldsymbol{\rho}, \boldsymbol{\gamma}).$$

It suffices to show that the claim holds for each one of the three terms above.

1. The first term, $q_{k,i}(\boldsymbol{\zeta}, \boldsymbol{\phi}, \boldsymbol{\rho}, \boldsymbol{\gamma})$, is handled by the inductive hypothesis. We can conclude that all the terms of degree less than o_i in ζ are zero. Moreover, $\tilde{q}_{k,i} = g_i(\boldsymbol{\gamma})\hat{q}_{k,i}(\boldsymbol{\rho}, \boldsymbol{\gamma})$.

2. The second term is nonzero only in the case that $\{i\} \in \mathcal{K}$, in which case $S_i = \{i\} \in \mathbb{K}$ and $o_i = 1$, so we do not have a contradiction since the $\zeta^{o_i} = \zeta$ only and we do not have terms with degree less than o_i. Moreover, the coefficient of ζ is $\gamma_i \rho_1 \phi_{\{i\}} = g_i(\boldsymbol{\gamma})\rho_1 \phi_{\{i\}}$.

3. We break into cases. *Case a.* If $\{i\} \oplus \{i_1\} \oplus \cdots \oplus \{i_r\} = S_i$, then $S_i \setminus \{i\} \subset \{i_1, \ldots, i_r\}$, so $1 + \sum_{l=1}^{r} o_{i_l} \geq o_i$, and so no new terms of degree less than o_i are added. Moreover, the only way the third term has degree o_i in ζ is when $i \neq i_1 \neq \ldots \neq i_r$ while still having $\{i, i_1, \ldots, i_r\} = S_i$; otherwise the power of ζ is $1 + \sum_{l=1}^{r} o_{i_l} > o_i$. In that case, letting $r' = |S_i| - 1$, the coefficient of ζ^{o_i} in the third term is

$$\gamma_i \left(\frac{\rho_{r'+1}}{r'!} \phi_{S_i} \prod_{i' \in S_i \setminus \{i\}} \tilde{q}_{k,i'}(\boldsymbol{\phi}, \boldsymbol{\rho}, \boldsymbol{\gamma}) \sum_{(i_1, \ldots, i_{r'}) \in [P]^{r'}} \mathbb{1}(\{i, i_1, \ldots, i_{r'}\} = S_i) \right)$$

$$= \gamma_i \left(\frac{\rho_{r'+1}}{r'!} \phi_{S_i} \prod_{i' \in S_i \setminus \{i\}} g_{i'}(\boldsymbol{\gamma})\hat{q}_{k,i'}(\boldsymbol{\rho}, \boldsymbol{\gamma}) \sum_{(i_1, \ldots, i_{r'}) \in [P]^{r'}} \mathbb{1}(\{i, i_1, \ldots, i_{r'}\} = S_i) \right)$$

$$= \gamma_i \prod_{i' \in S_i \setminus \{i'\}} g_{i'}(\boldsymbol{\gamma}) \left(\frac{\rho_{r'+1}}{r'!} \phi_{S_i} \prod_{i' \in S_i \setminus \{i\}} \hat{q}_{k,i'}(\boldsymbol{\rho}, \boldsymbol{\gamma}) \sum_{(i_1, \ldots, i_{r'}) \in [P]^{r'}} \mathbb{1}(\{i, i_1, \ldots, i_{r'}\} = S_i) \right)$$

$$= g_i(\boldsymbol{\gamma}) \cdot \hat{q}(\boldsymbol{\rho}, \boldsymbol{\gamma}) \text{ for a certain } \hat{q}.$$

35

Case b. If $\{i\} \oplus \{i_1\} \oplus \cdots \oplus \{i_r\} = S_{i'}$ for some $i' \neq i$, then either $i \in \{i_1, \ldots, i_r\}$, in which case $1 + \sum_{l=1}^{r} o_{i_l} > o_i$. Otherwise, we must have $i' > i$. But in this case $o_{i'} > o_i$ since $i \in S_{i'}$, so we also have $\sum_{l=1}^{r} o_{i_l} > o_i$ and again no new terms of degree less than o_i are added. In fact, only terms of degree strictly more than o_i are added. Thus, in either case the coefficient of the term with degree o_i in ζ is simply 0.

Finally, we showed that each of the three terms have at least the degree o_i in ζ. Moreover, each term can be decomposed as $g_i(\boldsymbol{\gamma})\hat{q}(\boldsymbol{\rho}, \boldsymbol{\gamma})$ for a certain $\hat{q}$. Thus, overall we can decompose

$$\tilde{q}_{k,i}(\boldsymbol{\phi}, \boldsymbol{\rho}, \boldsymbol{\gamma}) = g_i(\boldsymbol{\gamma}) \cdot \hat{q}_{k+1,i}(\boldsymbol{\phi}, \boldsymbol{\rho}), \text{concluding the proof of the lemma.}$$

$\square$

Claim 4. *Let*

$$r_{\boldsymbol{z}}(\zeta, \boldsymbol{\phi}, \boldsymbol{\rho}, \boldsymbol{\gamma}) = \sum_i z_i q_{k_1, i}(\zeta, \boldsymbol{\phi}, \boldsymbol{\rho}, \boldsymbol{\gamma}).$$

Then, for each distinct pair $\boldsymbol{z}, \boldsymbol{z}' \in \{+1, -1\}^P$, we have $r_{\boldsymbol{z}}(\zeta, \boldsymbol{\phi}, \boldsymbol{m}) - r_{\boldsymbol{z}'}(\zeta, \boldsymbol{\phi}, \boldsymbol{m}) \not\equiv 0$ as a polynomial in ζ and $\boldsymbol{\gamma}$.

Proof. Recall the definition of o_i from Lemma E.14. Fix $i \in [P]$ be such that $z_i \neq z_i'$ and o_i is minimized. It is always possible to choose such an i since $\boldsymbol{z} \neq \boldsymbol{z}'$. Again $\tilde{r}_{\boldsymbol{z}}(\boldsymbol{\phi}, \boldsymbol{\rho}, \boldsymbol{\gamma})$ is the term with degree o_i in ζ. Using Lemma E.14 then

$$\tilde{r}_{\boldsymbol{z}}(\boldsymbol{\phi}, \boldsymbol{\rho}, \boldsymbol{\gamma}) - \tilde{r}_{\boldsymbol{z}'}(\boldsymbol{\phi}, \boldsymbol{\rho}, \boldsymbol{\gamma}) = \sum_{i' \text{ s.t. } o_{i'}=o_i, z_{i'} \neq z_{i'}'} (z_{i'} - z_{i'}') \tilde{q}_{k_1, i'}(\boldsymbol{\phi}, \boldsymbol{\rho}, \boldsymbol{\gamma}),$$

but $\tilde{q}_{k_1, i'}$ are non-zero polynomials in $\boldsymbol{\gamma}$ with different dependence that is captured by $g_{i'}(\boldsymbol{\gamma})$ according to Lemma E.14. Thus, $r_{\boldsymbol{z}}(\zeta, \boldsymbol{\phi}, \boldsymbol{m}) - r_{\boldsymbol{z}'}(\zeta, \boldsymbol{\phi}, \boldsymbol{m}) \not\equiv 0$. $\square$

E.5.3 Linear Independence of Powers of Polynomials

Similar to Abbe et al. [2022], we finish our proof using the classical result of Newman and Slater [1979] about the linear independence of large powers of polynomials.

Proposition E.15 (Remark 5.2 in Newman and Slater [1979])**.** *Let $R_1, \ldots, R_m \in \mathbb{C}[\zeta]$ be non-constant polynomials such that for all $i \neq i' \in [m]$ we have $R_i(\zeta)$ is not a constant multiple of $R_{i'}(\zeta)$. Then for $L \geq 8m^2$ we have that $(R_1)^L, \ldots, (R_m)^L \in \mathbb{C}[\zeta]$ are $\mathbb{C}$-linearly independent.*

We will finally show that $\det(\boldsymbol{M}(\zeta, \boldsymbol{\alpha}, \boldsymbol{m}, \boldsymbol{\gamma})) \not\equiv 0$.

Proof of Lemma E.13. Let us fix $\boldsymbol{\kappa} \in \mathbb{R}^P$ such that for all $\boldsymbol{z} \neq \boldsymbol{z}'$, we have $r_{\boldsymbol{z}}(\zeta, \boldsymbol{\alpha}, \boldsymbol{m}, \boldsymbol{\kappa}) - r_{\boldsymbol{z}'}(\zeta, \boldsymbol{\alpha}, \boldsymbol{m}, \boldsymbol{\kappa}) \not\equiv 0$ as polynomials in ζ. This can be ensured by drawing $\boldsymbol{\kappa} \sim \text{Unif}([0.5, 1.5]^P)$, since for all $\boldsymbol{z} \neq \boldsymbol{z}'$ we have $r_{\boldsymbol{z}}(\zeta, \boldsymbol{\alpha}, \boldsymbol{m}, \boldsymbol{\gamma}) - r_{\boldsymbol{z}'}(\zeta, \boldsymbol{\alpha}, \boldsymbol{m}, \boldsymbol{\gamma}) \not\equiv 0$ as polynomials in $\zeta, \boldsymbol{\gamma}$ by Claim 4. We now rewrite $\tilde{r}_{\boldsymbol{z}}(\zeta) = r_{\boldsymbol{z}}(\zeta, \boldsymbol{\alpha}, \boldsymbol{m}, \boldsymbol{\kappa})$ to further indicate that the variables $\boldsymbol{\phi} = \boldsymbol{\alpha}, \boldsymbol{\rho} = \boldsymbol{m}$ and $\boldsymbol{\gamma} = \boldsymbol{\kappa}$ are instantiated, and that we are looking at a polynomial over ζ. We have also chosen $m_i = i!\binom{L}{i}$ for all $i \in \{0, \ldots, L\}$. We then have

$$M_{\boldsymbol{z}, j}(\zeta, \boldsymbol{\alpha}, \boldsymbol{m}, \boldsymbol{\kappa}) = (1 + \tilde{r}_{\boldsymbol{z}}(\zeta_j))^L.$$

Also, observe from the recurrence relations ζ divides $q_{k_1, i}(\zeta, \boldsymbol{\alpha}, \boldsymbol{m}, \boldsymbol{\kappa})$ for each $i \in [P]$, so ζ_j divides $\tilde{r}_{\boldsymbol{z}}(\zeta_j) = \sum_{i=1}^{P} z_i q_{k_1, i}(\zeta_j, \boldsymbol{\alpha}, \boldsymbol{m}, \boldsymbol{\kappa})$. Therefore, polynomials $(1 + \tilde{r}_{\boldsymbol{z}}(\zeta_j))$ and $, (1 + \tilde{r}_{\boldsymbol{z}'}(\zeta_j))$ are not constant multiples of each other for any $\boldsymbol{z} \neq \boldsymbol{z}'$.

Now see ζ as variables, and construct the Wronskian matrix over the L-th power polynomials $\{(1 + \tilde{r}_{\boldsymbol{z}}(\zeta))^L\}_{\boldsymbol{z} \in \{+1, -1\}^P}$. This is a $2^P \times 2^P$ matrix $\boldsymbol{H}(\zeta)$ whose entries are indexed by $\boldsymbol{z}$ and $l \in [2^P]$ and defined by:

$$H_{\boldsymbol{z}, l}(\zeta) = \frac{\partial^{l-1}}{\partial \zeta^{l-1}} (1 + \tilde{r}_{\boldsymbol{z}}(\zeta))^L.$$

Applying Proposition E.15 implies the polynomials $\{(1 + \tilde{r}_{\boldsymbol{z}}(\zeta))^L\}_{\boldsymbol{z} \in \{+1, -1\}^P}$ are linearly-independent, and so the Wronskian determinant is nonzero as a polynomial in ζ:

$$\det(\boldsymbol{H}(\zeta)) \not\equiv 0.$$

Also, observe that we can write $\det(\boldsymbol{H}(\zeta)) = \frac{\partial}{\partial \zeta_2} \frac{\partial^2}{\partial \zeta_3^2} \cdots \frac{\partial^{2^P - 1}}{\partial \zeta_{2^P}^{2^P - 1}} \det(\boldsymbol{M}(\boldsymbol{\zeta}, \boldsymbol{\alpha}, \boldsymbol{m}, \boldsymbol{\kappa})) \mid_{\zeta = \zeta_1 = \cdots = \zeta_{2^P}}$.

This finally gives us $\det(\boldsymbol{M}(\boldsymbol{\zeta}, \boldsymbol{\alpha}, \boldsymbol{m}, \boldsymbol{\kappa})) \not\equiv 0$ as a polynomial in $\boldsymbol{\zeta}$, and thus, $\det(\boldsymbol{M}(\boldsymbol{\zeta}, \boldsymbol{\alpha}, \boldsymbol{m}, \boldsymbol{\gamma})) \not\equiv 0$ as a polynomial in $\boldsymbol{\zeta}$ and $\boldsymbol{\gamma}$. $\qquad \square$

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and precede the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist",**
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes, it does.

   Guidelines:
   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We do specify the limitations of our work.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Yes, we do provide all the assumptions and correct proofs.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We do provide the code and the experiments are fully reproducible.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We attach the code in the supplementary material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We specify our training algorithm and related details.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We do report the error bars whenever necessary.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: All experiments are relatively small scale and can be reproduced under 1 hr of compute on a standard computer with 16 GB RAM.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: It does.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: This is a theoretical work with no societal impact.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This paper does not have models or data with safety concerns.

Guidelines:

- The answer NA means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [NA]

Justification: The paper does not use any existing assets.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: The paper does not release new asset.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This does not require crowd sourcing nor research with human subject.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: Our study does not require IRB approval.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.