ATTENTION LAYERS PROVABLY SOLVE SINGLE-LOCATION REGRESSION

Pierre Marion
Institute of Mathematics
EPFL
Lausanne, Switzerland
pierre.marion@epfl.ch

Raphaël Berthier
Sorbonne Université, Inria
Centre Inria de Sorbonne Université
Paris, France
raphael.berthier@inria.fr

Gérard Biau
Sorbonne Université
Institut Universitaire de France
Paris, France
gerard.biau@upmc.fr

Claire Boyer
Université Paris-Saclay
Institut Universitaire de France
Orsay, France

ABSTRACT

Attention-based models, such as Transformer, excel across various tasks but lack a comprehensive theoretical understanding, especially regarding token-wise sparsity and internal linear representations. To address this gap, we introduce the *single-location regression* task, where only one token in a sequence determines the output, and its position is a latent random variable, retrievable via a linear projection of the input. To solve this task, we propose a dedicated predictor, which turns out to be a simplified version of a non-linear self-attention layer. We study its theoretical properties, by showing its asymptotic Bayes optimality and analyzing its training dynamics. In particular, despite the non-convex nature of the problem, the predictor effectively learns the underlying structure. This work highlights the capacity of attention mechanisms to handle sparse token information and internal linear structures.

1 INTRODUCTION

Attention-based models (Bahdanau et al., 2015), such as Transformer (Vaswani et al., 2017), have achieved unprecedented performance in various learning tasks, including natural language processing (NLP), e.g., text generation (Bubeck et al., 2023), translation (Luong et al., 2015), sentiment analysis (Song et al., 2019; Sun et al., 2019; Xu et al., 2019), and audio/speech analysis (Bahdanau et al., 2016). These developments have led to many architectural and algorithmic variants of attention-based models (see the review by Lin et al., 2022). At a high level, the success of attention has been linked to its ability to manage long-range dependencies in input sequences (Bahdanau et al., 2015; Vaswani et al., 2017), since it consists in computing pairwise dependence between input tokens according to their projection in learned directions, independently of their location in the sequence.

On the theoretical front, however, a deeper understanding of attention-based neural networks is still in its infancy. This limited progress is due both to the complexity of the architectures and to the disturbing diversity of relevant tasks. A common approach to tackle these challenges is to introduce a simplified task that models certain features of real-world tasks, followed by demonstrating a simplified version of the attention mechanism capable of solving the task. Prominent examples of this pattern include studying in-context learning with linearized attention (Ahn et al., 2023; von Oswald et al., 2023; Zhang et al., 2024), topic understanding with single-layer attention and alternate minimization scheme (Li et al., 2023b), learning spatial structure with positional attention (Jelassi et al., 2022), next-token prediction with latent bigram (Bietti et al., 2023; Tian et al., 2023) or causal graph (Nichani et al., 2024) structures, and sparse token selection (Wang et al., 2024). We refer to Appendix F for additional discussion on some of these related works. While these works shed light on some abilities of Transformer, they do not encompass all the characteristics of tasks where

Transformer performs well, in particular in NLP. Two features of particular interest, which to our knowledge have not been addressed in previous theoretical studies on Transformer, are token-wise sparsity, where relevant information is contained in a limited number of tokens, and internal linear representations, which are interpretable representations of the input constructed by the model.

Contributions. To understand why attention is a suitable architecture for addressing these features, we introduce *single-location regression*, a novel statistical task where attention-based predictors excel (Section 2). In a nutshell, this task is a regression problem with a sequence of tokens as input. The key novelty is that only one token determines the prediction, and the location of this token is a latent random variable that changes based on the input sequence. Consequently, solving the task requires first identifying the location of the relevant token, which can be done by learning a latent linear projection, followed by performing regression on that token.

To tackle this problem, we propose a dedicated predictor, which turns out to be a simplified version of a non-linear self-attention layer. We show that this attention-based predictor is asymptotically Bayes optimal, whereas more standard linear regressors fail to perform better than the null predictor. We then analyze the training dynamics of the proposed predictor, when trained to minimize the theoretical risk by projected gradient descent. Despite the non-convexity of the problem and the non-linearity of this transformer-based method, we show that the learned predictor successfully retrieves the underlying structure of the task and thus solves single-location regression.

Organization. Section 2 presents the mathematical framework of single-location regression, followed by motivations from language processing. Section 3 is dedicated to defining our predictor and explaining its connection with attention. We then move on to the mathematical study, from both statistical (Section 4) and optimization (Section 5) points of view. Section 6 concludes the paper.

2 Single-location regression task

In this section, we describe our statistical task, and connect it to language processing motivations.

2.1 Statistical setting

We consider a regression scenario where the inputs are sequences of L random *tokens*[1] $(X_1, \dots, X_L)$ taking values in $\mathbb{R}^d$. The output $Y \in \mathbb{R}$ is assumed to be given by

$$Y = X_{J_0}^\top v^\star + \xi, \qquad (P_{\text{learn}})$$

where J_0 is a latent discrete random variable on $\{1, \dots, L\}$ and, conditionally on J_0,

$$\begin{cases} X_{J_0} & \sim & \mathcal{N}\Big(\sqrt{\frac{d}{2}} k^\star, \gamma^2 I_d\Big) \\ X_\ell & \sim & \mathcal{N}(0, I_d) \quad \text{for} \quad \ell \neq J_0 \,. \end{cases}$$

In the above formulation, $\mathcal{N}(\mu, \Sigma)$ denotes the normal distribution with expectation μ and covariance matrix Σ, and I_d is the identity matrix of size $d \times d$. All vectors are considered as column matrices, and the noise term ξ is assumed to be a centered random variable independent of X and J_0, with finite second-order moment ε^2. Conditionally on J_0, the tokens $(X_j)_{1 \leqslant j \leqslant L}$ are assumed to be independent.

The parameters of the regression problem (P_{learn}) are the *unknown* vectors $k^\star$ and $v^\star$, both assumed to be on the unit sphere $\mathbb{S}^{d-1}$ in dimension d, i.e., $\|k^\star\|_2 = \|v^\star\|_2 = 1$. The output is determined by a specific token in the sentence, indexed by the discrete random variable J_0 on $\{1, \dots, L\}$. This token can be detected via its mean, which is proportional to $k^\star$, contrarily to the others which have zero mean. Once X_{J_0} is identified, the prediction is formed as a linear projection in the direction $v^\star$. Therefore, the originality and difficulty of this task lies in the fact that the response Y is linearly related to a single informative token X_{J_0}, whose location varies from sequence to sequence—in this sense, the problem is sparse, but with a random support.

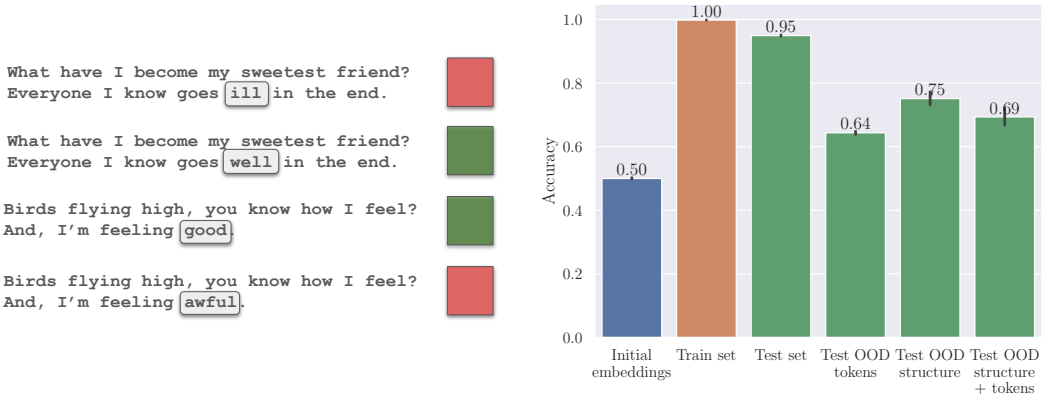
A knee-jerk reaction would be to fit a linear model to the pair $(X_1^\top, \dots, X_L^\top, Y)$. One might also consider tackling the problem with classical statistical approaches dedicated to sparsity, such

[1] For the sake of simplicity, we interchangeably use the terms "token" and "embedding", although they have slightly different meanings in the NLP community.

as a Lasso estimator or a group-Lasso technique (Hastie et al., 2009). However, as we will see (in Section 4), all linear predictors fail due to the unknown and changing location of J_0. We note in addition that $\mathbb{E}[\|X_\ell\|_2^2] = d$ when $\ell \neq J_0$, while $\mathbb{E}[\|X_{J_0}\|_2^2] = d/2 + \gamma^2 d$. Therefore, choosing $\gamma^2 = 1/2$ implies that tokens have the same squared norm in expectation, whether they are discriminatory of not. This shows that any approach based on comparing the magnitude of the tokens does not yield meaningful results. Ultimately, it is necessary to implement a more sophisticated approach, capable of taking into account the characteristics of the problem.

2.2 Language processing motivation

The structure of the task (P_{learn}) is motivated by natural language processing (NLP), and more specifically by two features, token-wise sparsity and internal linear representations, as we detail next.



(a) Examples of input-output pairs. The input is a text containing two sentences (e.g., a question and an answer), and the task is to perform sentiment analysis only for the second sentence. The Y output is symbolized here by a color code, where green (resp. red) corresponds to positive (resp. negative) feelings. The relevant information is sparse, typically concentrated in a single token: changing the grey token flips the output.

(b) Accuracy of logistic regression on embeddings of [CLS] tokens in the hidden layers of a pretrained Transformer model. Initial embeddings of [CLS] (at layer 0) are not context-aware, so they have a pure-chance accuracy of 50%. In hidden layers, the [CLS] token contains a representation of the sentence that achieves high scores and is robust to out-of-distribution changes in token distribution and sentence structure.

Figure 1: A simple sentiment analysis task with synthetic data, which exemplifies (a) token-wise sparsity and (b) internal linear representations. We refer to Appendix E for details on the experiment.

Token-wise sparsity. In language tasks, the relevant information is often contained in few tokens, where we recall that tokens correspond to small text units (typically, words or subwords), which are embedded in $\mathbb{R}^d$ using a learned dictionary. This sparsity is revealed by the success of sparse attention (Martins & Astudillo, 2016; Niculae & Blondel, 2017; Correia et al., 2019; Child et al., 2019; Jaszczur et al., 2021; Kim et al., 2022; Farina et al., 2024), which is competitive with full attention while attending to fewer tokens. As an illustration, we consider a simple sentiment analysis task in Figure 1a, and observe that changing one token flips the output. This is modeled in (P_{learn}) by having the output Y depend on a single token J_0, whose location furthermore varies with the input.

Internal linear representations. Linear projections of internal representations of Transformer (a.k.a. linear probing) contain interpretable information (Bolukbasi et al., 2021; Burns et al., 2023; Li et al., 2023a). Such a linear structure is also present in the learned token embeddings that are fed as input to language models (Mikolov et al., 2013a;b; Bolukbasi et al., 2016; Nanda et al., 2023; Wen-Yi & Mimno, 2023). In our task (P_{learn}), the two directions $k^\star$ and $v^\star$ have to be learned by the model in order to solve the task. Figure 2 gives an example of possible such directions for the toy task described above. While this illustration relies on initial embeddings, similar structures also appear in the intermediate representations of Transformer. This is shown in Figure 1b, where we observe that pretrained Transformer architectures indeed build internal representations that are sufficient to solve the task with a linear classifier.

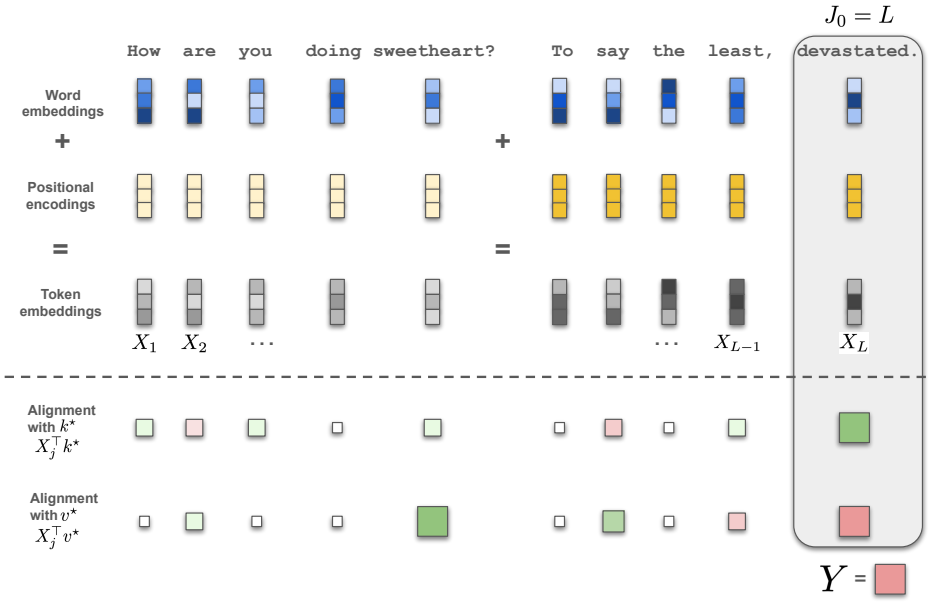


Figure 2: Modeling of an NLP task within our statistical setting (P_{learn}). The token embeddings $X_1, \dots, X_L$ are constructed by adding the embeddings of each word and a positional encoding. For illustration purposes, we assume that each token corresponds to a word, and that the positional encoding solely depends on the part of the sentence (before or after the question mark), which differs from usual practice. Then, let the direction $k^\star$ encode both the notion of sentiment and the position in the second part of the sentence. Thus only the last token of the sentence is aligned (positively) with $k^\star$, and we have $J_0 = L$. As for $v^\star$, it encodes whether the word is associated with a positive or negative sentiment. Note that several tokens are positively or negatively aligned with $v^\star$, but the output Y only depends on the token J_0. This illustrates the interest of having two latent directions $k^\star$ and $v^\star$, one that filters the informative token and one that aligns with the output Y.

We acknowledge that our statistical task presents limitations such as fixed sequence length, independent tokens, and output depending only on a single token. More complex models could be considered, but at significant technical cost. Moreover, as argued above, our problem (P_{learn}) preserves interesting aspects of NLP tasks, which makes it relevant for theoretical study of Transformer. Furthermore, it is an original statistical task requiring the implementation of a customized estimation strategy. It is precisely in this context that attention models prove their effectiveness, as we show next.

3 An attention-based predictor to solve the regression task

In this section, we propose a predictor adapted to the problem (P_{learn}) and discuss its connection with attention. In order to make our point as clear as possible, the construction is divided into three steps. We represent the input sequence in a matrix format $\mathbb{X} \in \mathbb{R}^{L \times d}$, where $\mathbb{X} = (X_1 | X_2 | \cdots | X_L)^\top$.

Step 1: An oracle non-differentiable predictor. If the vectors $(k^\star, v^\star) \in (\mathbb{S}^{d-1})^2$ were known, then a natural procedure to solve the task (P_{learn}) would be to predict Y from $\mathbb{X}$ via

$$T(\mathbb{X}) = (\mathbb{X}v^\star)_{j_0(\mathbb{X})} = X_{j_0(\mathbb{X})}^\top v^\star \,, \quad \text{where} \quad j_0(\mathbb{X}) = \operatorname*{arg\,max}_{1 \leqslant \ell \leqslant L} (\mathbb{X}k^\star)_\ell \,. \tag{1}$$

The $\arg\max$ part detects the location J_0 by exploiting the fact that all X_ℓ have zero mean except X_{J_0}, while the $\mathbb{X}v^\star$ part exploits the linear relationship $Y = X_{J_0}^\top v^\star + \xi$. In a more compact format, this ideal predictor can be rewritten as $T(\mathbb{X}) = \sum_{\ell=1}^{L} \mathbb{1}_{\arg\max(\mathbb{X}k^\star)=\ell} (\mathbb{X}v^\star)_\ell$, which is a linear regression in the direction $v^\star$ with non-differentiable weights depending on $k^\star$.

Step 2: A trainable predictor. In practice, the vectors $k^\star$ and $v^\star$ are unknown and must be estimated from the data. In addition, the non-differentiability of the $\arg\max$ function poses significant

optimization challenges. To solve this problem, the most common approach in machine learning is to replace $\arg\max$ with a softmax function with inverse temperature $\lambda > 0$, i.e., for $z = (z_1, \dots, z_L) \in \mathbb{R}^L$, $[\mathrm{softmax}(\lambda z)]_j = e^{\lambda z_j} / \sum_{\ell=1}^L e^{\lambda z_\ell}$. This leads us to the model

$$T_\lambda^{(\mathrm{soft},k,v)}(\mathbb{X}) = \sum_{\ell=1}^L [\mathrm{softmax}(\lambda \mathbb{X}k)]_\ell (\mathbb{X}v)_\ell = \mathrm{softmax}\big(\lambda \underbrace{\mathbb{X}k}_{L\times 1}\big)^\top \underbrace{\mathbb{X}v}_{L\times 1}, \tag{2}$$

where $k, v \in \mathbb{S}^{d-1}$, and the superscript 'soft' is used to indicate the presence of the softmax function.

Step 3: The final predictor. The softmax nonlinearity, by inducing a coupling between all tokens, significantly complicates the mathematical analysis. To alleviate this difficulty, we replace it by the component-wise nonlinear function $\mathrm{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt$, which is differentiable, increasing on $\mathbb{R}$, and such that $\mathrm{erf}(-\infty) = -1$ and $\mathrm{erf}(\infty) = 1$. We are therefore led to our operational model

$$T_\lambda^{(k,v)}(\mathbb{X}) = \mathrm{erf}\big(\lambda \mathbb{X}k\big)^\top \mathbb{X}v = \sum_{\ell=1}^L \mathrm{erf}\big(\lambda X_\ell^\top k\big) X_\ell^\top v\,, \tag{3}$$

where the erf function is applied component-wise. The choice of this activation function enables closed-form expectations for functions of Gaussian random variables (see, e.g., Lemma 18). Note that the role of softmax in attention is an open question in the community. Several empirical papers investigate simplifying softmax into a component-wise nonlinearity (Qin et al., 2022; Shen et al., 2023; Wortsman et al., 2023; Ramapuram et al., 2024), and have observed a similar performance. These works emphasize the importance of the normalization λ when replacing softmax, which we also find out to play an important role (see Corollary 2 and Section 5).

Connection to attention. It turns out that our estimation method finds a natural interpretation in terms of attention models. To see this, consider a model consisting of a single attention layer with a single head (Vaswani et al., 2017)

$$T_\lambda^{(Q,K,V,O)}(\mathbb{X}) = \mathrm{softmax}\Big(\lambda \underbrace{\mathbb{X}Q}_{L\times p} \underbrace{K^\top \mathbb{X}^\top}_{p\times L}\Big) \underbrace{\mathbb{X}V}_{L\times p} \underbrace{O^\top}_{p\times o}, \tag{4}$$

where the dimensions $p, o \in \mathbb{N}^*$ are hyperparameters of the model, the softmax function is applied row by row, $Q, K, V \in \mathbb{R}^{d\times p}$ and $O \in \mathbb{R}^{o\times p}$ are the regular query, key, value, and output matrices, and λ is usually taken to be $1/\sqrt{p}$. In practice, the attention head is added to $\mathbb{X}$ via a skip connection, which enforces $o = d$. In a nutshell, K detects which tokens are relevant in the sentence, V encodes the regression coefficient, and Q encodes where to store the information.

In a supervised context, it is classical in practice to concatenate in first position an additional token [CLS] to the tokenized sentence $\mathbb{X}$ (see, e.g., Devlin et al., 2019). In this context, only the first coordinate of the output is used for the prediction task. Thus, we focus on the first row of (4), corresponding to the embedding of [CLS], namely

$$T_\lambda^{(Q,K,V,O)}(\mathbb{X})_1 = \mathrm{softmax}\left(\lambda\, a\, K^\top \mathbb{X}^\top\right) \mathbb{X}VO^\top, \tag{5}$$

with $a = X_{[\mathrm{CLS}]}^\top Q \in \mathbb{R}^{1\times p}$, where $X_{[\mathrm{CLS}]} \in \mathbb{R}^d$ denotes the embedding of the [CLS] token.

It is important to note that only considering the first output coordinate is a mathematically valid simplification for a single attention layer, but not when multiple layers are stacked, as all coordinates of the attention output contribute. Nevertheless, even in this latter more realistic case, the [CLS] token—or the similar concepts of attention sinks and registers—has been empirically shown to play a crucial role (Clark et al., 2019; Darcet et al., 2024; Xiao et al., 2024). This is also confirmed by our experiment in Figure 1b, where we show that the [CLS] token in pretrained Transformer architectures stores an internal representation of the sentence that is sufficient to solve simple NLP tasks with a linear classifier. This further motivates the need to understand how information is stored in this token.

It turns out that there is a direct connection between the model $T_\lambda^{(\mathrm{soft},k,v)}(\mathbb{X})$ defined in (2) and the attention model $T_\lambda^{(Q,K,V,O)}(\mathbb{X})_1$ described in (5). To see this, take $o = 1$, to adapt the model (5) for univariate regression, and set $p = 1$, a reasonable assumption given both empirical and theoretical

evidence suggesting that Transformer parameter matrices are low-rank (Aghajanyan et al., 2021; Kajitsuka & Sato, 2024). Then, let $Q \in \mathbb{R}^{d\times 1}$ be any vector with positive correlation with $X_{[\mathrm{CLS}]}$ (for instance it suffices to take $Q = X_{[\mathrm{CLS}]}$), and $O = 1$. We then deduce that

$$T_\lambda^{(Q,K,V,O)}(\mathbb{X})_1 = T_{\lambda X_{[\mathrm{CLS}]}^\top Q}^{(\mathrm{soft},K,V)}(\mathbb{X})\,.$$

In other words, *the attention layer* (5) *matches the considered predictor in* (2) with a softmax inverse temperature proportional to the scalar product between $X_{[\mathrm{CLS}]}$ and Q. Thus, our results, in particular the study of the training dynamics in Section 5, can be seen as a model of how Transformer builds internal representations of the input during training. This is also supported by numerical experiments showing that Transformer layers behave similarly to our predictor (see Appendix E).

4 Risk of the oracle and of the linear predictors

Now that we have constructed our predictor $T_\lambda^{(k,v)}$ (see Eq. (3)), a first key question is to assess its statistical performance. Recall that $k, v \in \mathbb{S}^{d-1}$ are the two parameters of the model, and their purpose is to approximate their theoretical counterparts $k^\star$ and $v^\star$ defined in (1). This begs in particular the question of the performance of the *oracle predictor* $T_\lambda^{(k^\star,v^\star)}$. To answer these questions, we introduce the risk of the predictor, which is measured by the mean squared error

$$\mathcal{R}_\lambda(k,v) = \mathbb{E}\Big[\Big(Y - T_\lambda^{(k,v)}(\mathbb{X})\Big)^2\Big]\,. \tag{6}$$

To proceed with the analysis, we make the following assumption.

Assumption 1. *The vectors $k^\star, v^\star \in \mathbb{S}^{d-1}$ are orthogonal, i.e., $k^{\star\top} v^\star = 0$.*

This assumption is made everywhere in the remainder of the paper, even though it is not reminded explicitly at each result. It is a relatively mild assumption in a high-dimensional setting where any two independent vectors uniformly distributed on the sphere are close to being orthogonal.

Oracle predictor. Our first result characterizes the risk of the proposed transformer model (3) with oracle parameters $(k^\star, v^\star)$. All the proofs of the paper are deferred to the Appendix.

Theorem 1. *There exists a function $\mathcal{R}_\lambda^< : \mathbb{R}^5 \to \mathbb{R}$ such that, for any $(k,v) \in (\mathbb{S}^{d-1})^2$,*

$$\mathcal{R}_\lambda(k,v) = \mathcal{R}_\lambda^<(\kappa,\nu,\theta,\eta,\rho)\,,$$

where $\kappa := k^\top k^\star$, $\nu := v^\top v^\star$, $\theta := v^\top k^\star$, $\eta := k^\top v^\star$, and $\rho := k^\top v$. A closed-form expression of $\mathcal{R}_\lambda^<$ is given in Appendix C. In particular,

$$\begin{aligned}\mathcal{R}_\lambda(k^\star,v^\star) &= \mathcal{R}_\lambda^<(1,1,0,0,0)\\ &= \gamma^2 - 2\gamma^2\,\mathrm{erf}\left(\lambda\sqrt{\frac{d}{2(1+2\lambda^2\gamma^2)}}\right) + \gamma^2\zeta\left(\lambda\sqrt{\frac{d}{2}}, \lambda^2\gamma^2\right) + (L-1)\zeta(0,\lambda^2) + \varepsilon^2\,,\end{aligned}$$

where, for $t, \gamma \in \mathbb{R}$,

$$\zeta(t,\gamma^2) := \mathbb{E}\left[\mathrm{erf}^2(t+G)\right]\,, \qquad G \sim \mathcal{N}(0,\gamma^2)\,. \tag{7}$$

This result is fundamental for the analysis of gradient descent studied in the next section since it reduces the dimension of the dynamical system defined by the optimization dynamics. Before delving into the optimization analysis, we study below the statistical optimality of the estimator $\mathcal{R}_\lambda(k^\star, v^\star)$ and its comparison with linear regression.

Asymptotic Bayes optimality. Let us start by observing that the Bayes risk associated with problem (P_{learn}) is larger than ε^2, which follows from elementary properties of the conditional expectation (Le Gall, 2022, Chapter 11). Indeed, using the Pythagorean theorem, one easily shows that

$$\mathbb{E}[(Y - \mathbb{E}[Y|X])^2] \geqslant \mathbb{E}[(Y - \mathbb{E}[Y|X, J_0])^2] = \mathbb{E}[\xi^2] = \varepsilon^2\,. \tag{8}$$

Then, the following corollary to Theorem 1 shows that the oracle predictor achieves the Bayes-optimal risk in the asymptotic scaling $L \ll 1/\lambda^2 \ll d$.

Corollary 2. *Assume a joint asymptotic scaling where $d \to \infty$ and $L = o(d)$. Taking λ such that $\lambda\sqrt{d} \to \infty$ and $\lambda\sqrt{L} \to 0$, we have*

$$\mathcal{R}_\lambda(k^\star, v^\star) \to \varepsilon^2 \,.$$

Thus, in this asymptotic regime, the oracle predictor $T_\lambda^{(k^\star,v^\star)}$ is asymptotically Bayes optimal.

Note that Corollary 2 holds for any finite $L \in \mathbb{N}_{>0}$, but L may also tend to infinity, as long as $L = o(d)$. Let us give an intuition on why this result holds and where the scalings of L and λ intervene. The oracle predictor can be decomposed as

$$T_\lambda^{(k^\star,v^\star)}(\mathbb{X}) = \underbrace{X_{J_0}^\top v^\star}_{=\mathbb{E}[Y|\mathbb{X},J_0]} \operatorname{erf}(\underbrace{\lambda X_{J_0}^\top k^\star}_{=\Theta(\lambda\sqrt{d})}) + \sum_{j\neq J_0} \underbrace{X_j^\top v^\star}_{=\Theta(1)} \operatorname{erf}(\underbrace{\lambda X_j^\top k^\star}_{=\Theta(\lambda)}) \tag{9}$$

With the scaling $\lambda\sqrt{d} \to \infty$, the argument of the first erf nonlinearity diverges to infinity with d. Thus it reaches the saturating part of erf, so the first term in (9) converges to $\mathbb{E}[Y|\mathbb{X}, J_0]$. On the other hand, the argument of the erf nonlinearities inside the sum are of order $\lambda = o(1)$. Thus they are in the linear part of erf. Therefore, the sum consists of $L-1$ independent terms, each of magnitude λ. As a consequence, by the central limit theorem, the whole sum is of order $\Theta(\lambda\sqrt{L})$, and we get

$$T_\lambda^{(k^\star,v^\star)}(\mathbb{X}) \approx \mathbb{E}[Y|\mathbb{X}, J_0] + \Theta(\lambda\sqrt{L}) \,.$$

Due to the scaling $\lambda\sqrt{L} \to 0$, the second term decays to zero, and the oracle predictor implements the conditional expectation of Y given $\mathbb{X}$ and J_0. This is the best that we can hope for: the predictor succeeds in inferring the latent variable J_0, then gives the best possible prediction of Y given $\mathbb{X}$ and J_0. We also see the crucial role played by the nonlinearity of erf, whose linear part acts for $j \neq J_0$ and saturating part for $j = J_0$. In particular, the reasoning would not hold for a linear activation.

Linear model. The asymptotic optimality of our oracle predictor is particularly striking in comparison to the risk of the optimal linear predictor. More precisely, let

$$\beta^\star \in \operatorname*{arg\,min}_{\beta\in\mathbb{R}^{dL}} \mathbb{E}\Big[(Y - (X_1^\top, \dots, X_L^\top)\beta)^2\Big]$$

be the optimal linear predictor for the regression task (P_{learn}). Its associated risk is $\mathcal{R}(\beta^\star) = \mathbb{E}\big[(Y - (X_1^\top, \dots, X_L^\top)\beta^\star)^2\big]$. Both the optimal predictor and its risk can be made explicit as follows.

Proposition 3. *Let $p_j = \mathbb{P}(J_0 = j)$ for $j \in \{1, \dots, L\}$. Then the optimal linear predictor is parameterized by $\beta^\star = (b_1 v^\star, \dots, b_L v^\star)$, with $b_j = \frac{\gamma^2 p_j}{1+p_j(\gamma^2-1)}$, and its risk is*

$$\mathcal{R}(\beta^\star) = \varepsilon^2 + \gamma^2 - \gamma^4 \sum_{j=1}^{L} \frac{p_j^2}{1 + p_j(\gamma^2 - 1)} \,.$$

In particular,

$$\mathcal{R}(\beta^\star) \geqslant \varepsilon^2 + \gamma^2 - \gamma^2(\gamma^2+1) \max_{j=1,\dots,L} p_j \,.$$

This result calls for a few comments. If the number of tokens is $L = 1$ or if J_0 is a constant location (meaning that one p_j is equal to 1 while the others are equal to 0), then the learning problem (P_{learn}) corresponds to a standard linear regression. In this case, $\mathcal{R}(\beta^\star) = \varepsilon^2$, and the linear predictor $(X_1, \dots, X_L) \mapsto (X_1^\top, \dots, X_L^\top)\beta^\star$ achieves the Bayes risk. At the other end of the spectrum, in the case where J_0 is uniform over $\{1, \dots, L\}$, the formula for the risk of the linear predictor simplifies to $\mathcal{R}(\beta^\star) = \varepsilon^2 + \gamma^2 - \frac{\gamma^4}{\gamma^2+L-1}$. When $L \to \infty$, this risk tends to $\varepsilon^2 + \gamma^2$, that is, the performance of the null predictor. In other words, the optimal linear predictor performs no better than always predicting zero. More generally, this conclusion is true in any limit where $L \to \infty$ and $\max p_j \to 0$. This can be explained by the fact that the location of the relevant token for prediction is random, varying from sentence to sentence. Unable to leverage this latent information, the linear regressor balances all its coefficients, resulting in poor prediction performance. This stands in sharp contrast to Corollary 2, which shows that the oracle predictor $T_\lambda^{(k^\star,v^\star)}$ is able to account for the complexity of

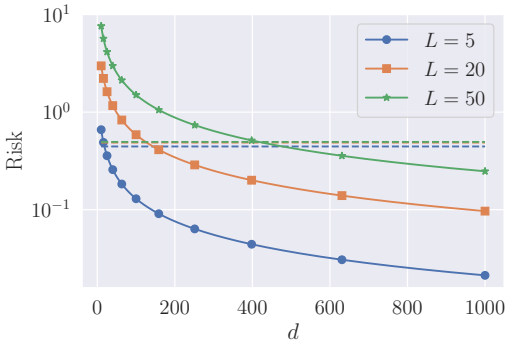


Figure 3: Risk of the oracle predictor (Theorem 1, solid lines) and of the best linear predictor (Proposition 3, dashed lines), depending on the dimensions d and L. The oracle predictor outperforms the linear predictor when scaling d. We take $\varepsilon^2 = 0$, $\gamma = 1/\sqrt{2}$, $\lambda = 1/d^{0.4}$, and all p_j equal to $1/L$.

the task, at least asymptotically. This is also illustrated by Figure 3, which compares the value of the risks given by Theorem 1 and Proposition 3.

Naturally, implementing the attention-based oracle predictor $T_\lambda^{(k^\star,v^\star)}$ requires knowledge of the parameters $k^\star$ and $v^\star$. Our goal in the next section is therefore to show that gradient descent is able to recover these parameters.

5 Gradient descent provably recovers the oracle predictor

This section is devoted to the analysis of the optimization dynamics in $(k, v) \in (\mathbb{S}^{d-1})^2$ of the risk

$$\mathcal{R}_\lambda(k,v) = \mathbb{E}\Big[\Big(Y - T_\lambda^{(k,v)}(\mathbb{X})\Big)^2\Big] = \mathbb{E}\Big[\Big(Y - \mathrm{erf}\big(\lambda \mathbb{X} k\big)^\top \mathbb{X} v\Big)^2\Big] .$$

We emphasize that $\mathcal{R}_\lambda(k, v)$ is a theoretical risk, which depends on the distribution of the pair $(\mathbb{X}, Y)$ (defined in Section 2). In practice, an empirical version of this risk is minimized. As we show experimentally (see Figure 5), the stochastic dynamics induced by the empirical version of the risk are qualitatively similar to the deterministic dynamics of the theoretical risk. In the remainder of the article, we focus on the theoretical risk for simplicity, and leave the empirical risk for future research.

Our optimization method is the Projected (Riemannian) Gradient Descent (PGD), described below.

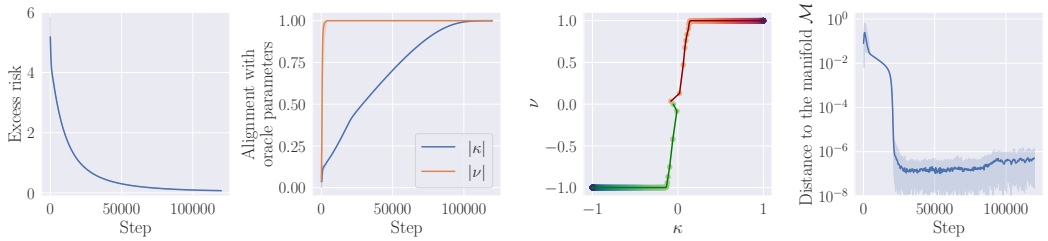
Definition 1 (PGD). *Given an initialization $(k_0, v_0) \in (\mathbb{S}^{d-1})^2$, a step size $\alpha > 0$, and an inverse temperature sequence $(\lambda_t)_{t\geqslant 0}$, the sequence $(k_t, v_t)_{t\geqslant 0} \in (\mathbb{S}^{d-1})^2$ is recursively defined by*

$$\begin{aligned} k_{t+1} &= \mathrm{Proj}_{\mathbb{S}^{d-1}}(k_t - \alpha(I_d - k_t k_t^\top)\nabla_k \mathcal{R}_{\lambda_t}(k_t, v_t)) , \\ v_{t+1} &= \mathrm{Proj}_{\mathbb{S}^{d-1}}(v_t - \alpha(I_d - v_t v_t^\top)\nabla_v \mathcal{R}_{\lambda_t}(k_t, v_t)) , \end{aligned} \tag{10}$$

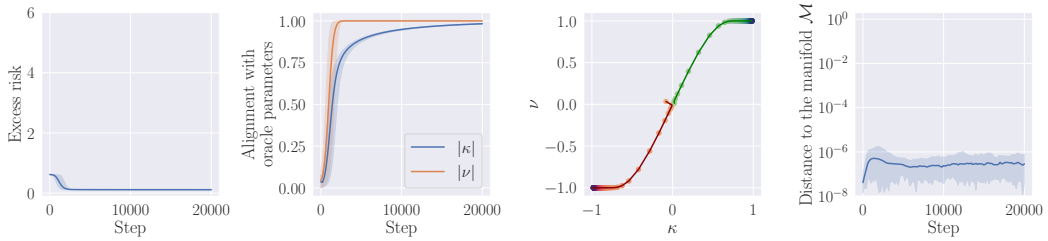
where $\mathrm{Proj}_{\mathbb{S}^{d-1}} : x \mapsto x/\|x\|_2$ denotes the Euclidean projection on the unit sphere of $\mathbb{R}^d$.

The operators $(I_d - k_t k_t^\top)$ and $(I_d - v_t v_t^\top)$ correspond to Riemannian gradient descent (Boumal, 2023, Section 4.3), meaning that we compute the gradient of the risk on the Riemannian manifold $(\mathbb{S}^{d-1})^2$. In other words, the gradient step is performed on the tangent space to the sphere at the current iterate. This is a precaution we are taking because, in the analysis of the dynamics, we rely on an expression of the risk (6) that is valid only on this manifold. In addition, this ensures that the subsequent projection on $\mathbb{S}^{d-1}$ is always well-defined, despite the fact that the sphere is a non-convex set, because iterates always avoid the pathological cases $k = 0$ or $v = 0$.

Experimentally, we observe in Figure 4a that PGD is able to recover the oracle parameters $(k^\star, v^\star)$. Note that running the PGD iterates (10) involves computing the gradients $\nabla_k \mathcal{R}_{\lambda_t}(k_t, v_t)$ and $\nabla_v \mathcal{R}_{\lambda_t}(k_t, v_t)$, which is non-trivial a priori. A direct approach using Monte Carlo simulations would require a large number of sample points to reduce variance, which is computationally intractable in particular in high-dimension, and in any case gives an approximate result. Instead, we leverage our closed form formula for $\mathcal{R}_\lambda^<$ from Theorem 1 to get exact values for the gradients (up to numerical errors). Interestingly, we also observe in Figure 4a that v aligns with $v^\star$ much faster than k aligns with $k^\star$. This is typical of two-timescale dynamics, which is a common framework in analysis of non-convex learning dynamics (Heusel et al., 2017; Dagréou et al., 2022; Hong et al., 2023; Marion & Berthier, 2023; Berthier et al., 2024; Marion et al., 2024).



(a) From a random initialization on $(\mathbb{S}^{d-1})^2$.



(b) From a random initialization on $\mathcal{M}$ (see Eq. (11)).

Figure 4: Convergence of PGD to the oracle parameters. **Left:** Excess risk as a function of the number of steps. **Middle left:** Alignment $|\kappa| = |k^\top k^\star|$ and $|\nu| = |v^\top v^\star|$ with the oracle parameters. **Middle right:** Trajectories of κ and ν in two repetitions of the experiments. Each repetition corresponds to a color, the trajectory starts in the middle and ends at a corner of the plot. **Right:** Distance to the invariant manifold $\mathcal{M}$. In all plots except the middle right ones, the experiment is repeated 30 times with independent random initializations, and 95% percentile intervals are plotted (but are not visible when the variance is too small). Parameters are $d = 400$, $L = 10$, $\gamma = \sqrt{1/2}$, and (a) $\lambda_t = 1/(1 + 10^{-4}t)$, (b) $\lambda_t = 0.1$. More details are given in Appendix E.

Moving on to the mathematical study, even with the formula for $\mathcal{R}_\lambda^<$, a full analysis of the dynamics (10) is difficult. For instance, the dynamics (10) can be formulated in terms of the five variables of $\mathcal{R}_\lambda^<$, but then one needs to study a 5-dimensional highly nonlinear dynamical system. In the following, we consider the case where the parameters are initialized on the submanifold of $(\mathbb{S}^{d-1})^2$

$$\mathcal{M} = \{(k, v) \in \mathbb{S}^{d-1} \times \mathbb{S}^{d-1}, k^\top v^\star = 0, v^\top k^\star = 0, k^\top v = 0\}\,. \tag{11}$$

We introduce this manifold on the one hand owing to the observation in Figure 4a (right) that the dynamics converge to this manifold even when initialized on the sphere, and on the other hand because this allows to reduce the problem to a lower-dimensional subspace and to simplify the expression of the risk. Clearly, due to Assumption 1, the oracle parameters $(k^\star, v^\star)$ belong to $\mathcal{M}$. A first key property of this manifold is invariance under the PGD dynamics.

Lemma 4. *The manifold $\mathcal{M}$ is invariant under the PGD dynamics (10), in the sense that if $(k_t, v_t) \in \mathcal{M}$, then $(k_{t+1}, v_{t+1}) \in \mathcal{M}$.*

This lemma shows that, if the initialization is taken on the manifold, then it is enough to understand the dynamics on the manifold to conclude. Such analysis on the manifold is tractable. This yields Theorem 5, our main result, which shows that the sequence $(k_t, v_t)_{t\geqslant 0}$ converges to the oracle values $(k^\star, v^\star)$ (up to a sign) as $t \to \infty$, for any small enough step size, and a constant inverse temperature.

Theorem 5. *Take a constant inverse temperature $\lambda_t \equiv \lambda > 0$. Then there exists $\overline{\alpha} > 0$ such that, for any step size $\alpha \leqslant \overline{\alpha}$, and for a generic initialization $(k_0, v_0) \in \mathcal{M}$, $(k_t, v_t) \xrightarrow[t\to\infty]{} \pm(k^\star, v^\star)$.*

This result shows that, despite the non-convexity of the risk, the attention layer trained by PGD can recover the underlying structure of the problem. Convergence to $(k^\star, v^\star)$ or $(-k^\star, -v^\star)$ is not at all problematic, since $T_\lambda^{(k^\star, v^\star)} = T_\lambda^{(-k^\star, -v^\star)}$ by symmetry of the erf function. Furthermore, recovery is guaranteed for a generic initialization on $\mathcal{M}$, in the sense that the pathological pairs $(k_0, v_0) \in \mathcal{M}$ such that PGD fails to recover the oracle parameters are of Lebesgue measure zero. The results of Theorem 5 are illustrated by Figure 4b. We observe that, due to roundoff errors, the dynamics are not exactly on the manifold but stay very close to the manifold.

We emphasize that the manifold $\mathcal{M}$ depends on the unknown parameters $k^\star$ and $v^\star$, making it impractical to initialize directly on the manifold. If the initialization is not on $\mathcal{M}$, more diverse phenomena are possible. As already pointed out in Figure 4a, it is possible to obtain recovery of $(k^\star, v^\star)$ and convergence to the manifold $\mathcal{M}$ from a general initialization on the sphere. This suggests that our analysis on the manifold is relevant, and completing the analysis for a general initialization is left for future work. However, we note that using a decreasing inverse temperature sequence λ_t is crucial for the recovery of $(k^\star, v^\star)$ when initialized out of $\mathcal{M}$. Indeed, to the best of our experiments, an iteration-independent choice of λ does not consistently lead to the recovery of $k^\star$ and $v^\star$ in this case (see Appendix E). This contrasts with the dynamics on the manifold proven in Theorem 5.

To investigate these behaviors, a fruitful direction would be to investigate the (local) stability of the manifold $\mathcal{M}$ for the PGD dynamics. If the manifold is indeed stable, one can hope to transfer the analysis on the manifold to dynamics initialized close to the manifold. Furthermore, recall that, in high dimension, random vectors on the sphere are close to being orthogonal. Thus, with high probability, a uniform initialization in $(\mathbb{S}^{d-1})^2$ falls in the neighborhood of the manifold $\mathcal{M}$, so that the local analysis should allow to conclude.

The proof of the theorem relies on a detailed analysis of the dynamics of the PGD algorithm on the invariant manifold $\mathcal{M}$, in particular the properties of its stationary points. These arguments, which lie at the intersection of dynamical systems and topology, are of independent interest. A key idea is to reduce the problem to a two-dimensional system depending only on $\kappa = k^\top k^\star$ and $\nu = v^\top v^\star$.

Finally, numerical experiments show that a full Transformer layer is able to solve the single-location regression task. Similarly to our simplified predictor, the weights align with the oracle parameters $k^\star$ and $v^\star$. This supports the connection drawn in Section 3 between our predictor and attention layers. We refer to Appendix E for details and plots, as well as experiments on multiple-location regression, a variant of single-location regression where the output depends on several tokens.

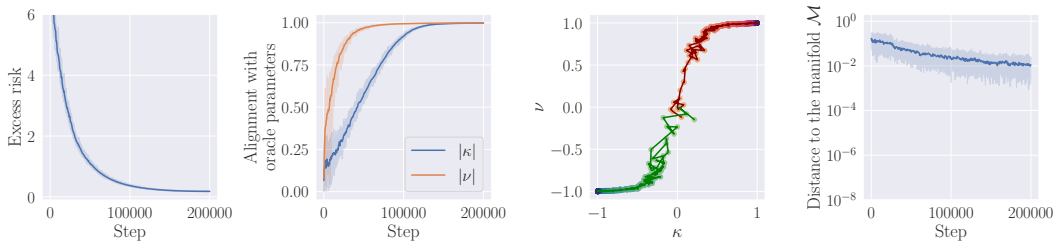


Figure 5: Convergence of online stochastic PGD to the oracle parameters from a random initialization on $(\mathbb{S}^{d-1})^2$. **Left:** Excess risk as a function of the number of steps. **Middle left:** Alignment $|\kappa| = |k^\top k^\star|$ and $|\nu| = |v^\top v^\star|$ with the oracle parameters. **Middle right:** Trajectories of κ and ν in two repetitions of the experiment. Each repetition corresponds to a color, the trajectory starts in the middle and ends at a corner of the plot. **Right:** Distance to the invariant manifold $\mathcal{M}$. In all plots except the middle right one, the experiment is repeated 30 times with independent random initializations, and 95% percentile intervals are plotted. Parameters are $d = 80$, $L = 10$, $\gamma = \sqrt{1/2}$, $\lambda_t = 2/(1 + 10^{-4}t)$, and a batch size of 5. More details are given in Appendix E.

6 Conclusion

This paper introduced *single-location regression*, a novel statistical task where the relevant information in the input sequence is supported by a single token. We analyzed the statistical properties and optimization dynamics of a natural estimator for this task, which can be seen as a basic attention layer. We hope this work encourages further research into how Transformer architectures address sparsity and long-range dependencies, while simultaneously constructing internal linear representations of their input——an aspect with significant implications for interpretability. Beyond NLP, potential applications include problems connected to sparse sequential modeling such as anomaly detection in time series. A natural extension of our framework is when relevant information is spread across a few input tokens rather than just one, which relates to multi-head attention. Future mathematical analyses should also consider extensions to general initialization schemes and stochastic dynamics. Our experiments (Figures 4a, 5, and Appendix E) yield encouraging results in all these directions.

ACKNOWLEDGMENTS

Authors thank Peter Bartlett, Linus Bleistein, Alex Damian, Spencer Frei, and Clément Mantoux for fruitful discussions and feedback. P.M. is supported by a Google PhD Fellowship.

REFERENCES

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In C. Zong, F. Xia, W. Li, and R. Navigli (eds.), Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 7319–7328. Association for Computational Linguistics, 2021.

Kwangjun Ahn, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. Transformers learn to implement preconditioned gradient descent for in-context learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), Advances in Neural Information Processing Systems, volume 36, pp. 45614–45650. Curran Associates, Inc., 2023.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Y. Bengio and Y. LeCun (eds.), 3rd International Conference on Learning Representations, 2015.

Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philémon Brakel, and Yoshua Bengio. End-to-end attention-based large vocabulary speech recognition. In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4945–4949, 2016.

Raphaël Berthier, Andrea Montanari, and Kangjie Zhou. Learning time-scales in two-layers neural networks. Foundations of Computational Mathematics, pp. 1–84, 2024.

Alberto Bietti, Vivien Cabannes, Diane Bouchacourt, Hervé Jégou, and Léon Bottou. Birth of a transformer: A memory viewpoint. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), Advances in Neural Information Processing Systems, volume 36, pp. 1560–1588. Curran Associates, Inc., 2023.

Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. In D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett (eds.), Advances in Neural Information Processing Systems, volume 29, pp. 4356–4364. Curran Associates, Inc., 2016.

Tolga Bolukbasi, Adam Pearce, Ann Yuan, Andy Coenen, Emily Reif, Fernanda Viégas, and Martin Wattenberg. An interpretability illusion for BERT. arXiv:2104.07143, 2021.

Nicolas Boumal. An Introduction to Optimization on Smooth Manifolds. Cambridge University Press, Cambridge, 2023.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/jax-ml/jax.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with GPT-4. arXiv:2303.12712, 2023.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In The Eleventh International Conference on Learning Representations, 2023.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. arXiv:1904.10509, 2019.

Lénaïc Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc., 2019.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? An analysis of BERT's attention. In T. Linzen, G. Chrupała, Y. Belinkov, and D. Hupkes (eds.), Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pp. 276–286. Association for Computational Linguistics, 2019.

Gonçalo M. Correia, Vlad Niculae, and André F.T. Martins. Adaptively sparse transformers. In K. Inui, J. Jiang, V. Ng, and X. Wan (eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 2174–2184. Association for Computational Linguistics, 2019.

Mathieu Dagréou, Pierre Ablin, Samuel Vaiter, and Thomas Moreau. A framework for bilevel optimization that enables stochastic and global variance reduction algorithms. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), Advances in Neural Information Processing Systems, volume 35, pp. 26698–26710. Curran Associates, Inc., 2022.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In The Twelfth International Conference on Learning Representations, 2024.

Richard D. De Veaux. Mixtures of linear regressions. Computational Statistics & Data Analysis, 8: 227–245, 1989.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186. Association for Computational Linguistics, 2019.

Mirko Farina, Usman Ahmad, Ahmad Taha, Hussein Younes, Yusuf Mesbah, Xiao Yu, and Witold Pedrycz. Sparsity in transformers: A systematic literature review. Neurocomputing, 582:127468, 2024.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. The Elements of Statistical Learning. Data Mining, Inference, and Prediction. Springer, New York, 2 edition, 2009.

Bobby He and Thomas Hofmann. Simplifying transformer blocks. In The Twelfth International Conference on Learning Representations, 2024.

Bobby He, James Martens, Guodong Zhang, Aleksandar Botev, Andrew Brock, Samuel L Smith, and Yee Whye Teh. Deep transformers without shortcuts: Modifying self-attention for faithful signal propagation. In The Eleventh International Conference on Learning Representations, 2023.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017.

Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. A two-timescale stochastic algorithm framework for bilevel optimization: Complexity analysis and application to actor-critic. SIAM Journal on Optimization, 33:147–180, 2023.

Sebastian Jaszczur, Aakanksha Chowdhery, Afroz Mohiuddin, Lukasz Kaiser, Wojciech Gajewski, Henryk Michalewski, and Jonni Kanerva. Sparse is enough in scaling transformers. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), Advances in Neural Information Processing Systems, volume 34, pp. 9895–9907. Curran Associates, Inc., 2021.

Samy Jelassi, Michael Sander, and Yuanzhi Li. Vision transformers provably learn spatial structure. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), Advances in Neural Information Processing Systems, volume 35, pp. 37822–37836. Curran Associates, Inc., 2022.

Tokio Kajitsuka and Issei Sato. Are transformers with one layer self-attention using low-rank weight matrices universal approximators? In The Twelfth International Conference on Learning Representations, 2024.

Sehoon Kim, Sheng Shen, David Thorsley, Amir Gholami, Woosuk Kwon, Joseph Hassoun, and Kurt Keutzer. Learned token pruning for transformers. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 784–794. Association for Computing Machinery, 2022.

Kenneth Lange. Optimization. Springer, New York, 2 edition, 2013.

Jean-François Le Gall. Measure Theory, Probability, and Stochastic Processes. Springer Cham, 2022.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), Advances in Neural Information Processing Systems, volume 36, pp. 41451–41530. Curran Associates, Inc., 2023a.

Yuchen Li, Yuanzhi Li, and Andrej Risteski. How do transformers learn topic structure: Towards a mechanistic understanding. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett (eds.), Proceedings of the 40th International Conference on Machine Learning, volume 202 of Proceedings of Machine Learning Research, pp. 19689–19729. PMLR, 2023b.

Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. A survey of transformers. AI Open, 3: 111–132, 2022.

Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In L. Màrquez, C. Callison-Burch, and J. Su (eds.), Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1412–1421. Association for Computational Linguistics, 2015.

Pierre Marion and Raphaël Berthier. Leveraging the two timescale regime to demonstrate convergence of neural networks. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), Advances in Neural Information Processing Systems, volume 36, pp. 64996–65029. Curran Associates, Inc., 2023.

Pierre Marion, Anna Korba, Peter Bartlett, Mathieu Blondel, Valentin De Bortoli, Arnaud Doucet, Felipe Llinares-López, Courtney Paquette, and Quentin Berthet. Implicit diffusion: Efficient optimization through stochastic sampling. arXiv:2402.05468, 2024.

Andre Martins and Ramon Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In M.F. Balcan and K.Q. Weinberger (eds.), Proceedings of The 33rd International Conference on Machine Learning, volume 48 of Proceedings of Machine Learning Research, pp. 1614–1623. PMLR, 2016.

Peter McCullagh and John A. Nelder. Generalized Linear Models. Chapman & Hall, London, 2 edition, 1983.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. arXiv:1309.4168, 2013a.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In L. Vanderwende, H. Daumé III, and K. Kirchhoff (eds.), Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 746–751. Association for Computational Linguistics, 2013b.

Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent linear representations in world models of self-supervised sequence models. In Y. Belinkov, S. Hao, J. Jumelet, N. Kim, A. McCarthy, and H. Mohebbi (eds.), Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP, pp. 16–30. Association for Computational Linguistics, 2023.

Eshaan Nichani, Alex Damian, and Jason D. Lee. How transformers learn causal structure with gradient descent. In International Conference on Machine Learning. PLMR, 2024.

Vlad Niculae and Mathieu Blondel. A regularized framework for sparse and structured neural attention. In I. Guyon, U. von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(85):2825–2830, 2011.

Mary Phuong and Marcus Hutter. Formal algorithms for transformers. arXiv:2207.09238, 2022.

William Press, Saul Teukolsky, William Vetterling, and Brian Flannery. Numerical Recipes: The Art of Scientific Computing. Cambridge University Press, Cambridge, 3 edition, 2007.

Zhen Qin, Weixuan Sun, Hui Deng, Dongxu Li, Yunshen Wei, Baohong Lv, Junjie Yan, Lingpeng Kong, and Yiran Zhong. cosformer: Rethinking softmax in attention. arXiv:2202.08791, 2022.

Jason Ramapuram, Federico Danieli, Eeshan Dhekane, Floris Weers, Dan Busbridge, Pierre Ablin, Tatiana Likhomanenko, Jagrit Digani, Zijin Gu, Amitis Shidani, et al. Theory, analysis, and best practices for sigmoid self-attention. arXiv:2409.04431, 2024.

Kai Shen, Junliang Guo, Xu Tan, Siliang Tang, Rui Wang, and Jiang Bian. A study on ReLU and Softmax in Transformer. arXiv:2302.06461, 2023.

Michael Shub. Global Stability of Dynamical Systems. Springer, New York, 1987.

Youwei Song, Jiahai Wang, Tao Jiang, Zhiyue Liu, and Yanghui Rao. Attentional encoder network for targeted sentiment classification. arXiv:1902.09314, 2019.

Charles M. Stein. Estimation of the mean of a multivariate normal distribution. The Annals of Statistics, 9:1135–1151, 1981.

Chi Sun, Luyao Huang, and Xipeng Qiu. Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. In J. Burstein, C. Doran, and T. Solorio (eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 380–385. Association for Computational Linguistics, 2019.

Yuandong Tian, Yiping Wang, Beidi Chen, and Simon S. Du. Scan and snap: Understanding training dynamics and token composition in 1-layer transformer. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), Advances in Neural Information Processing Systems, volume 36, pp. 71911–71947. Curran Associates, Inc., 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), Advances in Neural Information Processing Systems, volume 30, pp. 6000–6010. Curran Associates, Inc., 2017.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, Joao Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett (eds.), Proceedings of the 40th International Conference on Machine Learning, volume 202 of Proceedings of Machine Learning Research, pp. 35151–35174. PMLR, 2023.

Zixuan Wang, Stanley Wei, Daniel Hsu, and Jason D. Lee. Transformers provably learn sparse token selection while fully-connected nets cannot. In International Conference on Machine Learning. PLMR, 2024.

Andrea W Wen-Yi and David Mimno. Hyperpolyglot LLMs: Cross-lingual interpretability in token embeddings. In H. Bouamor, J. Pino, and K. Bali (eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pp. 1124–1131. Association for Computational Linguistics, 2023.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 38–45. Association for Computational Linguistics, 2020.

Mitchell Wortsman, Jaehoon Lee, Justin Gilmer, and Simon Kornblith. Replacing softmax with ReLU in vision transformers. arXiv:2309.08586, 2023.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In The Twelfth International Conference on Learning Representations, 2024.

Hu Xu, Bing Liu, Lei Shu, and Philip Yu. BERT post-training for review reading comprehension and aspect-based sentiment analysis. In J. Burstein, C. Doran, and T. Solorio (eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 2324–2335. Association for Computational Linguistics, 2019.

Ge Yang, Edward Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tuning large neural networks via zero-shot hyperparameter transfer. Advances in Neural Information Processing Systems, 34:17084–17097, 2021.

Ruiqi Zhang, Spencer Frei, and Peter L. Bartlett. Trained transformers learn linear models in-context. Journal of Machine Learning Research, 25(49):1–55, 2024.

Appendix

Organization of the Appendix. Section A presents the main steps of Theorem 5. The intermediate results of this proof, as well as the other statements of the main text, are proven in Section B. Section C provides an expression of the risk $\mathcal{R}$ beyond the manifold $\mathcal{M}$ that extends the one provided in Lemma 6 on $\mathcal{M}$. Section D gives some useful technical lemmas. Experimental details and additional results are in Section E. Finally, Section F discusses additional related models.

Notation. In the whole Appendix, we consider a constant inverse temperature schedule $\lambda_t \equiv \lambda > 0$, as in Theorem 5. For this reason, it is not necessary to make explicit the dependence of $\mathcal{R}_\lambda$ and $\mathcal{R}_\lambda^<$ on λ, and we use the lighter notations $\mathcal{R}$ and $\mathcal{R}^<$ instead.

A Outline of the proof of Theorem 5

This section outlines the essential steps for the proof of Theorem 5. For clarity, the proofs are to be found in Appendix B, except the proof of Proposition 10.

Step 1: Invariant manifold & reparameterization. We first show that the risk $\mathcal{R}(k, v)$ has a simpler expression when considered on the manifold $\mathcal{M}$.

Lemma 6. *The risk $\mathcal{R}(k, v)$ restricted to $\mathcal{M}$ has the form*

$$\mathcal{R}(k,v) = \gamma^2 - 2\gamma^2 v^\top v^\star \operatorname{erf}\left(\lambda\sqrt{\frac{d}{2(1+2\lambda^2\gamma^2)}}k^\top k^\star\right) + \gamma^2\zeta\Big(\lambda\sqrt{\frac{d}{2}}\,k^\top k^\star, \lambda^2\gamma^2\Big) + (L-1)\zeta(0,\lambda^2) + \varepsilon^2,$$

where, for $t, \gamma \in \mathbb{R}$,

$$\zeta(t,\gamma^2) := \mathbb{E}\left[\operatorname{erf}^2(t+G)\right], \qquad G \sim \mathcal{N}(0,\gamma^2)\,.$$

This expression has two main consequences. First, we use it to prove that the manifold $\mathcal{M}$ is invariant by PGD, according to Lemma 4. Second, we observe that the risk on the manifold depends on the variables $(k, v) \in \mathbb{S}^{d-1} \times \mathbb{S}^{d-1}$ only through the two scalar quantities

$$\kappa = k^\top k^\star \quad \text{and} \quad \nu = v^\top v^\star\,.$$

This suggests studying the dynamics in terms of the reduced variables $(\kappa, \nu) \in [-1, 1]^2$. More precisely, in the following, we denote by $\mathcal{R}^<$ the risk function $\mathcal{R}$ reparameterized as a function of (κ, ν), i.e., we let

$$\mathcal{R}^<(\kappa,\nu) = \gamma^2 - 2\gamma^2\nu \operatorname{erf}\left(\lambda\sqrt{\frac{d}{2(1+2\lambda^2\gamma^2)}}\kappa\right) + \gamma^2\zeta\Big(\lambda\sqrt{\frac{d}{2}}\,\kappa, \lambda^2\gamma^2\Big) + (L-1)\zeta(0,\lambda^2) + \varepsilon^2\,.$$

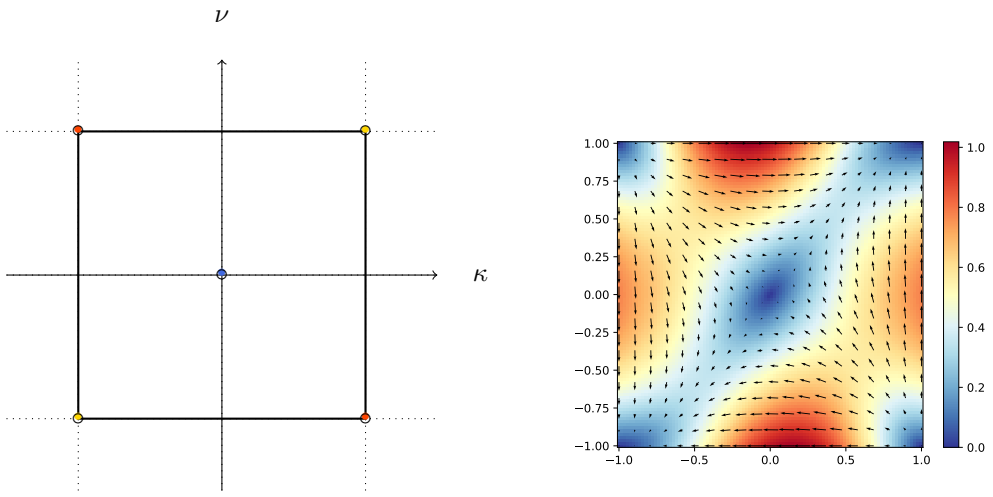
Note that, with a slight abuse of notation, we use $\mathcal{R}^<$ to denote both the function of five variables $(\kappa, \nu, \theta, \rho, \eta)$ (as in Theorem 1) and the function of only the first two variables (κ, ν). There should be no confusion, as both functions coincide on the manifold $\mathcal{M}$ where $\theta = \rho = \eta = 0$. We also denote the corresponding PGD iterates using this reparameterization by $(\kappa_t, \nu_t) := (k_t^\top k^\star, v_t^\top v^\star)$. With this notation, the following lemma reformulates the PGD iterations as an autonomous discrete dynamical system in terms of (κ_t, ν_t).

Lemma 7. *When initialized on the manifold $\mathcal{M}$, the PGD iterations (10) can be reformulated in terms of the autonomous discrete dynamical system*

$$(\kappa_{t+1}, \nu_{t+1}) = g(\kappa_t, \nu_t)\,, \tag{12}$$

where the mapping $g : [-1, 1]^2 \to [-1, 1]^2$ is given by

$$g(\kappa,\nu) = \left(\frac{\kappa - \alpha(\partial_\kappa\mathcal{R}^<(\kappa,\nu))(1-\kappa^2)}{\sqrt{1+\alpha^2(\partial_\kappa\mathcal{R}^<(\kappa,\nu))^2(1-\kappa^2)}}\,,\quad \frac{\nu - \alpha(\partial_\nu\mathcal{R}^<(\kappa,\nu))(1-\nu^2)}{\sqrt{1+\alpha^2(\partial_\nu\mathcal{R}^<(\kappa,\nu))^2(1-\nu^2)}}\right). \tag{13}$$



(a) (b)

Figure 6: Dynamics in (κ, ν) on the manifold $\mathcal{M}$. In (a), the fixed points of the dynamics are represented; the minimizers, saddle point, and maximizers are respectively depicted in yellow, blue and red. In (b), the vector field $(\kappa, \nu) \mapsto -(\partial_\kappa \mathcal{R}^<(\kappa, \nu)(1-\kappa^2), \partial_\nu \mathcal{R}^<(\kappa, \nu)(1-\nu^2))$ is displayed (the colormap corresponds to the magnitude of the vector field).

Step 2: Analysis of the stationary points. Regarding the dynamics restricted to the invariant manifold $\mathcal{M}$, we can characterize the limit points of the PGD iterates as follows.

Proposition 8. *For a sufficiently small step size α and for any $(k_0, v_0) \in \mathcal{M}$, the risk $\mathcal{R}^<$ is decreasing along the PGD iterates. Furthermore, the distance between successive PGD iterates tends to zero, and, if (κ, ν) is an accumulation point of the sequence of iterates $(\kappa_t, \nu_t)_{t \geqslant 0}$, then*

$$(1-\kappa^2)\partial_\kappa \mathcal{R}^<(\kappa, \nu) = 0 \quad \text{and} \quad (1-\nu^2)\partial_\nu \mathcal{R}^<(\kappa, \nu) = 0\,. \tag{14}$$

We stress that the system (14) of equations corresponds to fixed points of the dynamics (12)–(13). We next solve this system of equations.

Proposition 9. *The points $(\kappa, \nu) \in [-1, 1]^2$ satisfying (14) are $(\kappa, \nu) = (\pm 1, \pm 1)$[2] and $(\kappa, \nu) = (0, 0)$.*

The identity $(\kappa, \nu) = (\pm 1, \pm 1)$ corresponds to the situation where the variables (k, v) are aligned (up to sign) with the targets $(k^\star, v^\star)$. As the next proposition shows, these are the only global minima of $\mathcal{R}^<$.

Proposition 10. *The fixed points of the dynamics can be classified as follows:*

- (i) *The points $(\kappa, \nu) = (-1, 1)$ and $(1, -1)$ are global maxima of $\mathcal{R}^<$ on $[-1, 1]^2$.*
- (ii) *The points $(\kappa, \nu) = (1, 1)$ and $(-1, -1)$ are global minima of $\mathcal{R}^<$ on $[-1, 1]^2$.*
- (iii) *The point $(\kappa, \nu) = (0, 0)$ is a saddle point of $\mathcal{R}^<$ on $[-1, 1]^2$.*

The fixed points of the dynamics as well as the vector field

$$(\kappa, \nu) \mapsto -(\partial_\kappa \mathcal{R}^<(\kappa, \nu)(1-\kappa^2), \partial_\nu \mathcal{R}^<(\kappa, \nu)(1-\nu^2))$$

are displayed in Figure 6.

Step 3: Convergence to global minima. The convergence of the sequence of iterates $(\kappa_t, \nu_t)_{t \geqslant 0}$ to a global minimum is shown in two stages. First, we show that the iterates converge to one of the five fixed points described in Proposition 10.

[2]This notation is used to designate any extreme point of the square $[-1, 1]^2$, i.e., $(\kappa, \nu) = (1, 1)$, $(1, -1)$, $(-1, 1)$, and $(-1, -1)$.

Proposition 11. *For a sufficiently small step size α, the sequence of iterates $(\kappa_t, \nu_t)_{t\geqslant 0}$ converges to one of the five fixed points $\{(\pm 1, \pm 1), (0,0)\}$.*

Proof. According to Proposition 8, the distance between successive iterates (κ_t, ν_t) tends to zero. Therefore, the set of accumulation points of the sequence $(\kappa_t, \nu_t)_{t\geqslant 0}$ is connected (Lange, 2013, Proposition 12.4.1). Since there is a finite number of possible accumulation points (by Proposition 9), we deduce that the sequence has a unique accumulation point. Furthermore, the sequence belongs to a compact. Thus, it converges, and its limit is one of the five fixed points. □

It remains to precisely characterize the limit of the sequence $(\kappa_t, \nu_t)_{t\geqslant 0}$. To this aim, we begin by showing key properties of the gradient mapping g.

Proposition 12. *For a sufficiently small step size α, the mapping g is a local diffeomorphism around $(0,0)$, whose Jacobian matrix has one eigenvalue in $(0,1)$ and one eigenvalue in $(1,\infty)$. Furthermore, it is injective on $[-1,1]^2$, differentiable, and its Jacobian is non-degenerate.*

These properties enable us to apply the Center-Stable Manifold theorem (Shub, 1987, Theorem III.7), a tool from dynamical systems theory, to deduce the next proposition.

Proposition 13. *For a sufficiently small step size α, the set of initializations such that the sequence $(\kappa_t, \nu_t)_{t\geqslant 0}$ converges to $(-1,1)$, $(1,-1)$, or $(0,0)$ has Lebesgue measure zero (with respect to the Lebesgue measure on the manifold $\mathcal{M}$).*

Combining Proposition 11 and Proposition 13, we conclude that, provided the step size α is chosen small enough, the sequence $(\kappa_t, \nu_t)_{t\geqslant 0}$ almost surely converges to one of the minimizers, $(1,1)$ or $(-1,-1)$. This convergence is almost sure with respect to the Lebesgue measure on the manifold $\mathcal{M}$. Indeed, Proposition 13 ensures that the pathological initializations converging towards a maximizer or a saddle point are of Lebesgue measure zero. This concludes the proof of Theorem 5.

The use of the Center-Stable Manifold theorem is crucial to our proof. Unfortunately, this tool does not provide quantitative rates of convergence. Obtaining a rate is a challenging task as it would require quantifying the distance of the iterates to the saddle points of the risk (the dynamics is indeed slower near saddle points), which in turn requires other tools of analysis and potentially additional assumptions.

B Proofs of the main results

B.1 Proof of Lemma 6 and Theorem 1

We recall the formula for the risk

$$\mathcal{R}(k,v) = \mathbb{E}\Big[\Big(Y - \sum_{\ell=1}^{L} \operatorname{erf}(\lambda X_\ell^\top k) X_\ell^\top v\Big)^2\Big]$$

and the data model

$$Y = X_{J_0}^\top v^\star + \xi,$$

where

$$J_0 \in \mathcal{P}(\{1,\dots,L\}) \quad \text{and} \quad \begin{cases} X_{J_0} & \sim & \mathcal{N}\Big(\sqrt{\frac{d}{2}} k^\star, \gamma^2 I_d\Big) \\ X_\ell & \sim & \mathcal{N}(0, I_d) \quad \text{for} \quad \ell \neq J_0. \end{cases}$$

In the above expression for the risk, we can condition on the value of J_0. Actually, the conditioned risk is independent of J_0. Thus in this section, we assume without loss of generality that $J_0 = 1$ a.s.:

$$\mathcal{R}(k,v) = \mathbb{E}\Big[\Big(X_1^\top v^\star + \xi - \operatorname{erf}(\lambda X_1^\top k) X_1^\top v - \sum_{\ell=2}^{L} \operatorname{erf}(\lambda X_\ell^\top k) X_\ell^\top v\Big)^2\Big], \tag{15}$$

where

$$\begin{cases} X_1 & \sim & \mathcal{N}\Big(\sqrt{\frac{d}{2}} k^\star, \gamma^2 I_d\Big) \\ X_\ell & \sim & \mathcal{N}(0, I_d) \quad \text{for} \quad \ell \geqslant 2. \end{cases}$$

We rewrite this quantity in terms of multivariate standard Gaussian random variables. Using Assumption 1, we get

$$\mathcal{R}(k,v) = \mathbb{E}\Big[\Big(\gamma\widetilde{X}_1^\top v^\star + \xi - \mathrm{erf}\Big(\lambda\Big(\sqrt{\frac{d}{2}}k_*^\top k + \gamma\widetilde{X}_1^\top k\Big)\Big)\Big)\Big(\sqrt{\frac{d}{2}}k_*^\top v + \gamma\widetilde{X}_1^\top v\Big) - \sum_{\ell=2}^{L}\mathrm{erf}(\lambda X_\ell^\top k)X_\ell^\top v\Big)^2\Big],$$

where $\widetilde{X}_1, X_2, \dots, X_L \sim \mathcal{N}(0, I_d)$. This can be formulated in terms of the five scalar quantities $\kappa = k^\top k^\star$, $\nu = v^\top v^\star$, $\theta = v^\top k^\star$, $\eta = k^\top v^\star$, and $\rho = k^\top v$. Indeed, we have

$$\begin{aligned}\mathcal{R}(k,v) &= \mathcal{R}^<(\kappa,\nu,\theta,\eta,\rho)\\ &:= \mathbb{E}\Big[\Big(\gamma G_1^{v^\star} + \xi - \Big(\sqrt{\frac{d}{2}}\theta + \gamma G_1^v\Big)\mathrm{erf}\Big(\lambda\Big(\sqrt{\frac{d}{2}}\kappa + \gamma G_1^k\Big)\Big) - \sum_{\ell=2}^{L} G_\ell^v\mathrm{erf}(\lambda G_\ell^k)\Big)^2\Big],\end{aligned} \tag{16}$$

where

$$\begin{pmatrix} G_1^{v^\star}\\ G_1^v\\ G_1^k\end{pmatrix}, \dots, \begin{pmatrix} G_L^{v^\star}\\ G_L^v\\ G_L^k\end{pmatrix} \underset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, \begin{pmatrix} 1 & \nu & \eta\\ \nu & 1 & \rho\\ \eta & \rho & 1\end{pmatrix}\right). \tag{17}$$

This last expression only involves the five parameters $\kappa, \nu, \theta, \eta, \rho$, which play a role either explicitly in the function or as parameters of the covariance of the random variables. This proves the first statement of Theorem 1. A computation of a closed-form formula for this expectation is given in Appendix C.

On the manifold $\mathcal{M}$ defined by $\theta = \eta = \rho = 0$, we can simplify the expressions (16)–(17)

$$\mathcal{R}^<(\kappa,\nu,0,0,0) = \mathbb{E}\Big[\Big(\gamma G_1^{v^\star} + \xi - \gamma G_1^v\mathrm{erf}\Big(\lambda\Big(\sqrt{\frac{d}{2}}\kappa + \gamma G_1^k\Big)\Big) - \sum_{\ell=2}^{L} G_\ell^v\mathrm{erf}(\lambda G_\ell^k)\Big)^2\Big]$$

where $\begin{pmatrix} G_1^{v^\star}\\ G_1^v\end{pmatrix}, \dots, \begin{pmatrix} G_L^{v^\star}\\ G_L^v\end{pmatrix} \underset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, \begin{pmatrix} 1 & \nu\\ \nu & 1\end{pmatrix}\right)$, $G_1^k, \dots, G_L^k \underset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$, and $\xi \sim \mathcal{N}(0, \varepsilon^2)$ are independent.

We first expand in ξ and obtain

$$\mathcal{R}^<(\kappa,\nu,0,0,0) = \varepsilon^2 + \mathbb{E}\Big[\Big(\gamma G_1^{v^\star} - \gamma G_1^v\mathrm{erf}\Big(\lambda\Big(\sqrt{\frac{d}{2}}\kappa + \gamma G_1^k\Big)\Big) - \sum_{\ell=2}^{L} G_\ell^v\mathrm{erf}(\lambda G_\ell^k)\Big)^2\Big].$$

We now expand the square, as follows:

$$\begin{aligned}\mathcal{R}^<(\kappa,\nu,0,0,0) = \varepsilon^2 &+ \gamma^2\mathbb{E}\left[(G_1^{v^\star})^2\right] - 2\gamma^2\mathbb{E}\Big[G_1^{v^\star}G_1^v\mathrm{erf}\Big(\lambda\Big(\sqrt{\frac{d}{2}}\kappa + \gamma G_1^k\Big)\Big)\Big]\\ &+ \gamma^2\mathbb{E}\Big[(G_1^v)^2\mathrm{erf}^2\Big(\lambda\Big(\sqrt{\frac{d}{2}}\kappa + \gamma G_1^k\Big)\Big)\Big]\\ &- 2\sum_{\ell=2}^{L}\gamma\mathbb{E}\Big[\Big(G_1^{v^\star} - G_1^v\mathrm{erf}\Big(\lambda\Big(\sqrt{\frac{d}{2}}\kappa + \gamma G_1^k\Big)\Big)\Big)G_\ell^v\mathrm{erf}(\lambda G_\ell^k)\Big]\\ &+ \sum_{\ell,m=2}^{L}\mathbb{E}[G_\ell^v\mathrm{erf}(\lambda G_\ell^k)G_m^v\mathrm{erf}(\lambda G_m^k)].\end{aligned}$$

We address each term in this sum separately.

- Since $G_1^{v^\star} \sim \mathcal{N}(0,1)$, $\gamma^2\mathbb{E}\left[(G_1^{v^\star})^2\right] = \gamma^2$.
- Since $\begin{pmatrix} G_1^{v^\star}\\ G_1^v\end{pmatrix} \sim \mathcal{N}\left(0, \begin{pmatrix} 1 & \nu\\ \nu & 1\end{pmatrix}\right)$ is independent from $G_1^k \sim \mathcal{N}(0,1)$, we have

$$-2\gamma^2\mathbb{E}\Big[G_1^{v^\star}G_1^v\mathrm{erf}\Big(\lambda\Big(\sqrt{\frac{d}{2}}\kappa + \gamma G_1^k\Big)\Big)\Big] = -2\gamma^2\mathbb{E}\Big[G_1^{v^\star}G_1^v\Big]\mathbb{E}\Big[\mathrm{erf}\Big(\lambda\Big(\sqrt{\frac{d}{2}}\kappa + \gamma G_1^k\Big)\Big)\Big]$$

$$= -2\gamma^2\nu\mathbb{E}\Big[\mathrm{erf}\Big(\lambda\Big(\sqrt{\frac{d}{2}}\kappa + \gamma G_1^k\Big)\Big)\Big]\,.$$

Finally, using Lemma 18(ii), we obtain

$$-2\gamma^2\mathbb{E}\Big[G_1^{v^\star}G_1^v\mathrm{erf}\Big(\lambda\Big(\sqrt{\frac{d}{2}}\kappa + \gamma G_1^k\Big)\Big)\Big] = -2\gamma^2\nu\mathrm{erf}\Big(\lambda\sqrt{\frac{d}{2}}\frac{\kappa}{\sqrt{1+2\lambda^2\gamma^2}}\Big)\,.$$

- Since $G_1^v, G_1^k \sim_{\text{i.i.d.}} \mathcal{N}(0,1)$, we have

$$\gamma^2\mathbb{E}\Big[(G_1^v)^2\mathrm{erf}^2\Big(\lambda\Big(\sqrt{\frac{d}{2}}\kappa + \gamma G_1^k\Big)\Big)\Big] = \gamma^2\mathbb{E}\Big[(G_1^v)^2\Big]\mathbb{E}\Big[\mathrm{erf}^2\Big(\lambda\Big(\sqrt{\frac{d}{2}}\kappa + \gamma G_1^k\Big)\Big)\Big]$$
$$= \gamma^2\mathbb{E}\Big[\mathrm{erf}^2\Big(\lambda\Big(\sqrt{\frac{d}{2}}\kappa + \gamma G_1^k\Big)\Big)\Big]\,.$$

Using the definition of ζ in Eq. (7), we have

$$\gamma^2\mathbb{E}\Big[(G_1^v)^2\mathrm{erf}^2\Big(\lambda\Big(\sqrt{\frac{d}{2}}\kappa + \gamma G_1^k\Big)\Big)\Big] = \gamma^2\mathbb{E}\Big[(G_1^v)^2\Big]\mathbb{E}\Big[\mathrm{erf}^2\Big(\lambda\Big(\sqrt{\frac{d}{2}}\kappa + \gamma G_1^k\Big)\Big)\Big]$$
$$= \gamma^2\zeta\Big(\lambda\sqrt{\frac{d}{2}}\kappa, \lambda^2\gamma^2\Big)\,.$$

- For $\ell = 2, \dots, L$, $(G_1^{v^\star}, G_1^v, G_1^k)$, G_ℓ^v, and G_ℓ^k are independent. Thus

$$\mathbb{E}\Big[\Big(G_1^{v^\star} - G_1^v\mathrm{erf}\Big(\lambda\Big(\sqrt{\frac{d}{2}}\kappa + \gamma G_1^k\Big)\Big)\Big)G_\ell^v\mathrm{erf}(\lambda G_\ell^k)\Big]$$
$$= \mathbb{E}\Big[\Big(G_1^{v^\star} - G_1^v\mathrm{erf}\Big(\lambda\Big(\sqrt{\frac{d}{2}}\kappa + \gamma G_1^k\Big)\Big)\Big)\Big]\mathbb{E}\Big[G_\ell^v\Big]\mathbb{E}\Big[\mathrm{erf}(\lambda G_\ell^k)\Big] = 0\,,$$

where in the last step we use $\mathbb{E}[G_\ell^v] = 0$.

- Finally, to tackle the last term, we address the cases $\ell \neq m$ and $\ell = m$ separately. If $\ell \neq m$, as $G_\ell^v, G_\ell^k, G_m^v$, and G_m^k are independent, we have

$$\mathbb{E}[G_\ell^v\mathrm{erf}(\lambda G_\ell^k)G_m^v\mathrm{erf}(\lambda G_m^k)] = \mathbb{E}[G_\ell^v]\mathbb{E}[\mathrm{erf}(\lambda G_\ell^k)]\mathbb{E}[G_m^v]\mathbb{E}[\mathrm{erf}(\lambda G_m^k)] = 0\,.$$

If $\ell = m$, as $G_\ell^v, G_\ell^k \sim_{\text{i.i.d.}} \mathcal{N}(0,1)$, we have

$$\mathbb{E}[(G_\ell^v)^2\mathrm{erf}^2(\lambda G_\ell^k)] = \mathbb{E}[(G_\ell^v)^2]\mathbb{E}[\mathrm{erf}^2(\lambda G_\ell^k)] = \zeta(0,\lambda^2)\,.$$

Putting together these computations, we obtain

$$\mathcal{R}^<(\kappa,\nu,0,0,0) = \varepsilon^2 + \gamma^2 - 2\gamma^2\nu\mathrm{erf}\Big(\lambda\sqrt{\frac{d}{2}}\frac{\kappa}{\sqrt{1+2\lambda^2\gamma^2}}\Big) + \gamma^2\zeta\Big(\lambda\sqrt{\frac{d}{2}}\kappa, \lambda^2\gamma^2\Big)$$
$$+ (L-1)\zeta(0,\lambda^2)\,.$$

This proves Lemma 6. Taking $\kappa = \nu = 1$ proves Theorem 1.

B.2 Proof of Corollary 2

Recall that, according to Theorem 1,

$$\mathcal{R}_\lambda(k^\star, v^\star) = \gamma^2 - 2\gamma^2\,\mathrm{erf}\Big(\lambda\sqrt{\frac{d}{2(1+2\lambda^2\gamma^2)}}\Big) + \gamma^2\zeta\Big(\lambda\sqrt{\frac{d}{2}}, \lambda^2\gamma^2\Big) + (L-1)\zeta(0,\lambda^2) + \varepsilon^2\,,$$

where, for $t, \gamma \in \mathbb{R}$,

$$\zeta(t,\gamma^2) := \mathbb{E}\left[\mathrm{erf}^2(t + \gamma G)\right]\,, \qquad G \sim \mathcal{N}(0,1)\,.$$

We compute the limit of each term separately. First, we have

$$\lambda\sqrt{\frac{d}{2(1+2\lambda^2\gamma^2)}} \sim \frac{\lambda\sqrt{d}}{\sqrt{2}} \xrightarrow{d\to\infty} \infty\,. \tag{18}$$

Therefore, the second term of $\mathcal{R}_\lambda(k^\star, v^\star)$ tends to $-2\gamma^2$. To handle the third term, note by Jensen's inequality that

$$1 \geqslant \zeta\Big(\lambda\sqrt{\frac{d}{2}}, \lambda^2\gamma^2\Big) = \mathbb{E}\left[\mathrm{erf}^2\Big(\lambda\sqrt{\frac{d}{2}} + \lambda\gamma G\Big)\right] \geqslant \mathbb{E}\left[\mathrm{erf}\Big(\lambda\sqrt{\frac{d}{2}} + \lambda\gamma G\Big)\right]^2 .$$

Thus, by Lemma 18(ii),

$$1 \geqslant \zeta\Big(\lambda\sqrt{\frac{d}{2}}, \lambda^2\gamma^2\Big) \geqslant \mathrm{erf}^2\Big(\lambda\sqrt{\frac{d}{2(1+2\lambda^2\gamma^2)}}\Big) \to 1 ,$$

where we used (18). Thus the third term of $\mathcal{R}_\lambda(k^\star, v^\star)$ converges to γ^2. As for the fourth term, observe by Lemma 17 that

$$\mathrm{erf}^2(u) \leqslant \frac{4}{\pi}u^2 ,$$

hence

$$0 \leqslant \zeta(0, \lambda^2) \leqslant \frac{4}{\pi}\lambda^2\mathbb{E}[G^2] = \frac{4}{\pi}\lambda^2 .$$

Since $\lambda\sqrt{L} \to 0$, we get

$$(L-1)\zeta(0, \lambda^2) = \mathcal{O}(\lambda^2 L) = o(1) .$$

Putting everything together, we obtain

$$\mathcal{R}_\lambda(k^\star, v^\star) \xrightarrow{d\to\infty} \gamma^2 - 2\gamma^2 + \gamma^2 + 0 + \varepsilon^2 = \varepsilon^2 .$$

Since we already know by (8) that the Bayes risk is lower-bounded by ε^2, this proves that the Bayes risk is asymptotically *equal* to ε^2, and that the oracle predictor is asymptotically Bayes optimal.

B.3 Proof of Proposition 3

Let us first introduce a useful notation for the proof. If M is a block matrix, we denote by $M_{[ij]}$ its (i,j)-th block, and likewise, if u is a block vector, we denote by $u_{[j]}$ its j-th block. Next, note that

$$\begin{aligned}\mathbb{E}[Y^2] &= \varepsilon^2 + \mathbb{E}\left[((v^\star)^\top X_{J_0})^2\right] \\ &= \varepsilon^2 + \gamma^2\|v^\star\|_2^2 \\ &= \varepsilon^2 + \gamma^2,\end{aligned}$$

since $\|v^\star\|_2^2 = 1$. Recall that

$$\beta^\star \in \arg\min_{\beta\in\mathbb{R}^{dL}} \mathbb{E}\Big[(Y - (X_1^\top, \dots, X_L^\top)\beta)^2\Big]$$

is the optimal linear predictor. The classical formula for linear regression shows that

$$\beta^\star = \left(\mathbb{E}\left[\begin{pmatrix}X_1\\ \vdots \\ X_L\end{pmatrix}(X_1^\top, \dots, X_L^\top)\right]\right)^{-1} \mathbb{E}\left[((v^\star)^\top X_{J_0} + \xi)\begin{pmatrix}X_1\\ \vdots \\ X_L\end{pmatrix}\right] .$$

On the one hand, let

$$M = \begin{pmatrix}X_1\\ \vdots \\ X_L\end{pmatrix}(X_1^\top, \dots, X_L^\top) .$$

Then $\mathbb{E}[M] = \mathbb{E}[\mathbb{E}[M|J_0]]$, and $\mathbb{E}[M|J_0]$ is a block-diagonal matrix, where, for $j, j' \in \{1, \dots, L\}$,

$$\mathbb{E}[M|J_0 = j]_{[j',j']} = \delta_{j\neq j'} I_d + \delta_{j=j'}(\gamma^2 I_d + \frac{d}{2}k^\star(k^\star)^\top) .$$

Thus

$$\mathbb{E}[M]_{[j',j']} = (1-p_{j'})I_d + p_{j'}(\gamma^2 I_d + \frac{d}{2}k^\star(k^\star)^\top) = I_d + p_{j'}(\gamma^2 - 1)I_d + p_{j'}\frac{d}{2}k^\star(k^\star)^\top .$$

On the other hand, let

$$u = ((v^\star)^\top X_{J_0} + \xi) \begin{pmatrix} X_1 \\ \vdots \\ X_L \end{pmatrix} .$$

Then

$$\mathbb{E}[u] = \mathbb{E}\left[\begin{pmatrix} X_1 \\ \vdots \\ X_L \end{pmatrix} X_{J_0}^\top\right] v^\star = \mathbb{E}\left[\mathbb{E}\left[\begin{pmatrix} X_1 \\ \vdots \\ X_L \end{pmatrix} X_{J_0}^\top\right] \middle| J_0\right] v^\star$$
$$= \begin{pmatrix} p_1(\gamma^2 I_d & + & \frac{d}{2} k^\star (k^\star)^\top) \\ & \vdots & \\ p_L(\gamma^2 I_d & + & \frac{d}{2} k^\star (k^\star)^\top) \end{pmatrix} v^\star = \gamma^2 \begin{pmatrix} p_1 v^\star \\ \vdots \\ p_L v^\star \end{pmatrix},$$

since, by Assumption 1, $k^{\star\top} v^\star = 0$.

Since $\mathbb{E}[M]$ is a block-diagonal matrix and $\mathbb{E}[u]$ is a block vector, we get by standard computation rules for block matrices

$$\beta^\star_{[j]} = (\mathbb{E}[M]^{-1}\mathbb{E}[u])_{[j]} = \mathbb{E}[M]^{-1}_{[j,j]}\mathbb{E}[u]_{[j]} = \left(I_d + p_j(\gamma^2 - 1)I_d + p_j \frac{d}{2} k^\star (k^\star)^\top\right)^{-1} \gamma^2 p_j v^\star .$$

Recall the Sherman-Morrison formula (Press et al., 2007, Section 2.7.1), which states that for any vectors $u, v \in \mathbb{R}^d$, $(I_d + uu^\top)^{-1} v = \left(I_d - uu^\top/(1 + u^\top u)\right) v$. Applying this formula with orthogonal vectors, we obtain

$$\beta^\star_{[j]} = \left(1 + p_j(\gamma^2 - 1)\right)^{-1} \gamma^2 p_j v^\star = \frac{\gamma^2 p_j}{1 + p_j(\gamma^2 - 1)} v^\star ,$$

which shows the first formula of the proposition. Finally, the risk associated with the optimal linear predictor $(X_1^\top, \dots, X_L^\top) \mapsto (X_1^\top, \dots, X_L^\top)\beta^\star$ is given by

$$\begin{aligned} \mathcal{R}(\beta^\star) &= \mathbb{E}[Y^2] - \mathbb{E}[Y(X_1^\top \dots X_L^\top)\beta^\star] \\ &= \varepsilon^2 + \gamma^2 - \gamma^2 \cdot \begin{pmatrix} p_1(v^\star)^\top, & \dots, & p_L(v^\star)^\top \end{pmatrix} \begin{pmatrix} \beta^\star_{[1]} \\ \vdots \\ \beta^\star_{[L]} \end{pmatrix} \\ &= \varepsilon^2 + \gamma^2 - \gamma^4 \sum_{j=1}^L \frac{p_j^2}{1 + p_j(\gamma^2 - 1)} . \end{aligned} \tag{19}$$

This shows the formula for the risk given in the Proposition. To obtain the last bound, observe that, if $\gamma^2 \geqslant 1$, we have $1 + p_j(\gamma^2 - 1) \geqslant 1$. If $\gamma^2 \leqslant 1$, since $p_j \leqslant 1$, we have $1 + p_j(\gamma^2 - 1) \geqslant 1 + (\gamma^2 - 1) = \gamma^2$. Thus we obtain $1 + p_j(\gamma^2 - 1) \geqslant \min(1, \gamma^2)$. Therefore,

$$\begin{aligned} \mathcal{R}(\beta^\star) &\geqslant \varepsilon^2 + \gamma^2 - \max(\gamma^4, \gamma^2) \sum_{j=1}^L p_j^2 \\ &\geqslant \varepsilon^2 + \gamma^2 - \max(\gamma^4, \gamma^2) \sum_{j=1}^L p_j \cdot \max_{j=1,\dots,L} p_j \\ &\geqslant \varepsilon^2 + \gamma^2 - (\gamma^4 + \gamma^2) \max_{j=1,\dots,L} p_j \\ &\geqslant \varepsilon^2 + \gamma^2 - \gamma^2(\gamma^2 + 1) \max_{j=1,\dots,L} p_j . \end{aligned}$$

When all p_j are equal to $1/L$, all terms in the sum are equal, and Eq. (19) simplifies to

$$\mathcal{R}(\beta^\star) = \varepsilon^2 + \gamma^2 - L\gamma^4 \frac{\frac{1}{L^2}}{1 + \frac{1}{L}(\gamma^2 - 1)} = \varepsilon^2 + \gamma^2 - \frac{\gamma^4}{L + \gamma^2 - 1} .$$

B.4 PROOF OF LEMMA 4

As a first step in the proof, we prove the next lemma, which is the key towards the invariance property we are aiming at, in that it shows that, for a point on the manifold $\mathcal{M}$ (defined by $\theta = \eta = \rho = 0$), the gradient of the risk does not "push" the point outside of the manifold. Its proof leverages the expression of the risk as a function of five parameters derived in the previous section

Lemma 14. *At any point $(\kappa, \nu, \theta, \eta, \rho)$ such that $\theta = \eta = \rho = 0$, we have $\partial_\theta \mathcal{R}^< = \partial_\eta \mathcal{R}^< = \partial_\rho \mathcal{R}^< = 0$.*

Proof. We use Eq. (16)–(17) and change signs in the square function:

$$\begin{aligned}
&\mathcal{R}^<(\kappa, \nu, \theta, \eta, \rho) \\
&\quad = \mathbb{E}\Big[\Big(\gamma G_1^{v^\star} + \xi - \Big(\sqrt{\frac{d}{2}}\theta + \gamma G_1^v\Big)\mathrm{erf}\Big(\lambda\Big(\sqrt{\frac{d}{2}}\kappa + \gamma G_1^k\Big)\Big) - \sum_{\ell=2}^{L} G_\ell^v \mathrm{erf}(\lambda G_\ell^k)\Big)^2\Big] \\
&\quad = \mathbb{E}\Big[\Big(\gamma(-G_1^{v^\star}) - \xi - \Big(\sqrt{\frac{d}{2}}(-\theta) + \gamma(-G_1^v)\Big)\mathrm{erf}\Big(\lambda\Big(\sqrt{\frac{d}{2}}\kappa + \gamma G_1^k\Big)\Big) \\
&\qquad - \sum_{\ell=2}^{L}(-G_\ell^v)\mathrm{erf}(\lambda G_\ell^k)\Big)^2\Big]\,,
\end{aligned}$$

where

$$\begin{pmatrix} G_1^{v^\star} \\ G_1^v \\ G_1^k \end{pmatrix}, \dots, \begin{pmatrix} G_L^{v^\star} \\ G_L^v \\ G_L^k \end{pmatrix} \underset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, \begin{pmatrix} 1 & \nu & \eta \\ \nu & 1 & \rho \\ \eta & \rho & 1 \end{pmatrix}\right), \qquad \xi \sim \mathcal{N}(0, \varepsilon^2)\,.$$

Thus

$$\begin{pmatrix} -G_1^{v^\star} \\ -G_1^v \\ G_1^k \end{pmatrix}, \dots, \begin{pmatrix} -G_L^{v^\star} \\ -G_L^v \\ G_L^k \end{pmatrix} \underset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, \begin{pmatrix} 1 & \nu & -\eta \\ \nu & 1 & -\rho \\ -\eta & -\rho & 1 \end{pmatrix}\right), \qquad -\xi \sim \mathcal{N}(0, \varepsilon^2)\,.$$

As a consequence,

$$\mathcal{R}^<(\kappa, \nu, \theta, \eta, \rho) = \mathcal{R}^<(\kappa, \nu, -\theta, -\eta, -\rho)\,.$$

Taking the partial derivative in θ, we are led to

$$\partial_\theta \mathcal{R}^<(\kappa, \nu, \theta, \eta, \rho) = -\partial_\theta \mathcal{R}^<(\kappa, \nu, -\theta, -\eta, -\rho)\,.$$

At a point such that $\theta = \eta = \rho = 0$, this gives $\partial_\theta \mathcal{R}^<(\kappa, \nu, 0, 0, 0) = -\partial_\theta \mathcal{R}^<(\kappa, \nu, 0, 0, 0)$ and thus $\partial_\theta \mathcal{R}^<(\kappa, \nu, 0, 0, 0) = 0$. The proof for the other two derivatives $\partial_\eta \mathcal{R}$, $\partial_\rho \mathcal{R}$ is identical. □

We now complete the proof of Lemma 4. By the chain rule for total derivatives applied to $R(k, v) = \mathcal{R}^<(\kappa, \nu, \theta, \eta, \rho)$, and then by Lemma 14, on the manifold $\mathcal{M}$, we have

$$\nabla_k \mathcal{R} = (\partial_\kappa \mathcal{R}^<)k^\star + (\partial_\eta \mathcal{R}^<)v^\star + (\partial_\rho \mathcal{R}^<)v = (\partial_\kappa \mathcal{R}^<)k^\star\,, \tag{20}$$

and, similarly,

$$\nabla_v \mathcal{R} = (\partial_\nu \mathcal{R}^<)v^\star + (\partial_\theta \mathcal{R}^<)k^\star + (\partial_\rho \mathcal{R})k = (\partial_\nu \mathcal{R}^<)v^\star\,. \tag{21}$$

Recall the formulas for the PGD updates

$$\begin{aligned}
k_{t+1} &= \mathrm{Proj}_{\mathbb{S}^{d-1}}(k_t - \alpha(I - k_t k_t^\top)\nabla_k \mathcal{R}(k_t, v_t)) = \frac{k_t - \alpha(I - k_t k_t^\top)\nabla_k \mathcal{R}(k_t, v_t)}{\left\|k_t - \alpha(I - k_t k_t^\top)\nabla_k \mathcal{R}(k_t, v_t)\right\|_2}, \\
v_{t+1} &= \mathrm{Proj}_{\mathbb{S}^{d-1}}(v_t - \alpha(I - v_t v_t^\top)\nabla_v \mathcal{R}(k_t, v_t)) = \frac{v_t - \alpha(I - v_t v_t^\top)\nabla_v \mathcal{R}(k_t, v_t)}{\left\|v_t - \alpha(I - v_t v_t^\top)\nabla_v \mathcal{R}(k_t, v_t)\right\|_2}.
\end{aligned}$$

Let $c_k = \left\|k_t - \alpha(I - k_t k_t^\top)\nabla_k \mathcal{R}(k_t, v_t)\right\|_2$ and $c_v = \left\|v_t - \alpha(I - v_t v_t^\top)\nabla_v \mathcal{R}(k_t, v_t)\right\|_2$. Then, if $(k_t, v_t) \in \mathcal{M}$,

$$(v^\star)^\top k_{t+1} = \frac{(v^\star)^\top k_t - \alpha(v^\star)^\top (I - k_t k_t^\top)(\partial_\kappa \mathcal{R}^<(\kappa_t, \nu_t))k^\star}{c_k} = 0,$$

$$(k^\star)^\top v_{t+1} = \frac{(k^\star)^\top v_t - \alpha (k^\star)^\top (I - v_t v_t^\top)(\partial_\nu \mathcal{R}^<(\kappa_t, \nu_t)) v^\star}{c_v} = 0,$$

and

$$v_{t+1}^\top k_{t+1} = \frac{v_t^\top k_t - \alpha(\partial_\nu \mathcal{R}^<)((I - v_t v_t^\top)v^\star)^\top k_t - \alpha(\partial_\kappa \mathcal{R}^<)((I - k_t k_t^\top)k^\star)^\top v_t}{c_v c_k} + \frac{\alpha^2 (\partial_\kappa \mathcal{R}^<)(\partial_\nu \mathcal{R}^<)((I - k_t k_t^\top)k^\star)^\top (I - v_t v_t^\top) v^\star}{c_v c_k} = 0\,,$$

where we have omitted the dependence of $(\partial_\kappa \mathcal{R}^<)$ and $(\partial_\nu \mathcal{R}^<)$ in (κ_t, ν_t) in the last expression for the ease of readability. Note that the last term is equal to zero since

$$((I - k_t k_t^\top)k^\star)^\top (I - v_t v_t^\top) v^\star = (k^\star - \kappa_t k_t)^\top (v^\star - \nu_t v_t) = 0\,.$$

This shows that $(k_{t+1}, v_{t+1}) \in \mathcal{M}$.

B.5 Proof of Lemma 7

By definition of the PGD iterates and by (20)–(21), one has

$$\kappa_{t+1} = k_{t+1}^\top k^\star = \frac{\kappa_t - \alpha \partial_\kappa \mathcal{R}^<(\kappa_t, \nu_t)(k^\star)^\top (I - k_t k_t^\top) k^\star}{\sqrt{1 + \alpha^2 (\partial_\kappa \mathcal{R}^<)^2 \|(I - k_t k_t^\top)k^\star\|_2^2}} = \frac{\kappa_t - \alpha(\partial_\kappa \mathcal{R}^<)(1 - \kappa_t^2)}{\sqrt{1 + \alpha^2 (\partial_\kappa \mathcal{R}^<)^2 (1 - \kappa_t^2)}},$$
$$\nu_{t+1} = v_{t+1}^\top v^\star = \frac{\nu_t - \alpha \partial_\nu \mathcal{R}^<(\kappa_t, \nu_t)(v^\star)^\top (I - v_t v_t^\top) v^\star}{\sqrt{1 + \alpha^2 (\partial_\nu \mathcal{R}^<)^2 \|(I - v_t v_t^\top)v^\star\|_2^2}} = \frac{\nu_t - \alpha(\partial_\nu \mathcal{R}^<)(1 - \nu_t^2)}{\sqrt{1 + \alpha^2 (\partial_\nu \mathcal{R}^<)^2 (1 - \nu_t^2)}},$$

where we have used the Pythagorean theorem and the idempotent property of projection matrices for the denominator.

B.6 Proof of Proposition 8

In this proof, C denotes a constant that does not depend on the step t nor on the step size α, and which may vary from line to line. First note that the risk $\mathcal{R}^<$ is C^∞ on the compact set $[-1, 1]^2$. In particular, it is a Λ-smooth function for some $\Lambda > 0$, in the sense that its gradient is Λ-Lipschitz continuous. Thus

$$\mathcal{R}^<(\kappa_{t+1}, \nu_{t+1}) \leqslant \mathcal{R}^<(\kappa_t, \nu_t) + (\nabla \mathcal{R}^<(\kappa_t, \nu_t))^\top \begin{pmatrix} \kappa_{t+1} - \kappa_t \\ \nu_{t+1} - \nu_t \end{pmatrix} + \frac{\Lambda}{2} \left\| \begin{pmatrix} \kappa_{t+1} - \kappa_t \\ \nu_{t+1} - \nu_t \end{pmatrix} \right\|_2^2,$$

i.e.,

$$\begin{aligned} &\mathcal{R}^<(\kappa_{t+1}, \nu_{t+1}) - \mathcal{R}^<(\kappa_t, \nu_t) \\ &\quad \leqslant (\partial_\kappa \mathcal{R}^<)(\kappa_{t+1} - \kappa_t) + (\partial_\nu \mathcal{R}^<)(\nu_{t+1} - \nu_t) + \frac{\Lambda}{2}\left[(\kappa_{t+1} - \kappa_t)^2 + (\nu_{t+1} - \nu_t)^2\right]. \end{aligned} \tag{22}$$

Our goal in the following computations is to derive an inequality of the form

$$\begin{aligned} \mathcal{R}^<(\kappa_{t+1}, \nu_{t+1}) - \mathcal{R}^<(\kappa_t, \nu_t) \leqslant &-\alpha(\partial_\kappa \mathcal{R}^<)^2(1 - \kappa_t^2) - \alpha(\partial_\nu \mathcal{R}^<)^2(1 - \nu_t^2) \\ &+ C\alpha^2(\partial_\kappa \mathcal{R}^<)^2(1 - \kappa_t^2) + C\alpha^2(\partial_\nu \mathcal{R}^<)^2(1 - \nu_t^2)\,, \end{aligned}$$

which shall give us a descent lemma for α small enough. To this aim, observe that, by definition of the iterates (κ_t, ν_t) given by (12)–(13), one has

$$\begin{aligned} \kappa_{t+1} - \kappa_t &= \left[\frac{1}{\sqrt{1 + \alpha^2(\partial_\kappa \mathcal{R}^<)^2(1 - \kappa_t^2)}} - 1\right]\kappa_t - \frac{\alpha(\partial_\kappa \mathcal{R}^<)(1 - \kappa_t^2)}{\sqrt{1 + \alpha^2(\partial_\kappa \mathcal{R}^<)^2(1 - \kappa_t^2)}} \\ &= -\alpha(\partial_\kappa \mathcal{R}^<)(1 - \kappa_t^2) \\ &\quad + \left[\frac{1}{\sqrt{1 + \alpha^2(\partial_\kappa \mathcal{R}^<)^2(1 - \kappa_t^2)}} - 1\right](\kappa_t - \alpha(\partial_\kappa \mathcal{R}^<)(1 - \kappa_t^2))\,. \end{aligned} \tag{23}$$

As a consequence,

$$\begin{aligned}|\kappa_{t+1} - \kappa_t + \alpha(\partial_\kappa \mathcal{R}^<)(1-\kappa_t^2)| &\leqslant \Big|\frac{1}{\sqrt{1+\alpha^2(\partial_\kappa \mathcal{R}^<)^2(1-\kappa_t^2)}} - 1\Big||\kappa_t - \alpha(\partial_\kappa \mathcal{R}^<)(1-\kappa_t^2)| \\ &\leqslant \alpha^2(\partial_\kappa \mathcal{R}^<)^2(1-\kappa_t^2)|\kappa_t - \alpha(\partial_\kappa \mathcal{R}^<)(1-\kappa_t^2)| \\ &\leqslant C\alpha^2(\partial_\kappa \mathcal{R}^<)^2(1-\kappa_t^2)\,, \end{aligned} \tag{24}$$

where the second inequality holds by Lemma 16 and the last bound holds since the function $(\kappa, \nu) \mapsto |\kappa - \alpha(\partial_\kappa \mathcal{R}^<(\kappa,\nu))(1-\kappa^2)|$ is uniformly bounded for all $\alpha \leqslant 1$. This bound has two implications. First,

$$\begin{aligned}(\partial_\kappa \mathcal{R}^<)(\kappa_{t+1} - \kappa_t) + \alpha(\partial_\kappa \mathcal{R}^<)^2(1-\kappa_t^2) &= (\partial_\kappa \mathcal{R}^<)((\kappa_{t+1} - \kappa_t) + \alpha(\partial_\kappa \mathcal{R}^<)(1-\kappa_t^2)) \\ &\leqslant |\partial_\kappa \mathcal{R}^<||\kappa_{t+1} - \kappa_t + \alpha(\partial_\kappa \mathcal{R}^<)(1-\kappa_t^2)| \\ &\leqslant C\alpha^2(\partial_\kappa \mathcal{R}^<)^2(1-\kappa_t^2)\,, \end{aligned} \tag{25}$$

where we use the fact that $|\partial_\kappa \mathcal{R}^<|$ is bounded, and the bound (24). Second, since the square function is Lipschitz on compact sets, we have

$$|(\kappa_{t+1} - \kappa_t)^2 - (\alpha(\partial_\kappa \mathcal{R}^<)(1-\kappa_t^2))^2| \leqslant C\alpha^2(\partial_\kappa \mathcal{R}^<)^2(1-\kappa_t^2)\,.$$

Thus

$$\begin{aligned}(\kappa_{t+1} - \kappa_t)^2 &\leqslant \alpha^2(\partial_\kappa \mathcal{R}^<)^2(1-\kappa_t^2)^2 + C\alpha^2(\partial_\kappa \mathcal{R}^<)^2(1-\kappa_t^2) \\ &\leqslant C\alpha^2(\partial_\kappa \mathcal{R}^<)^2(1-\kappa_t^2)\,. \end{aligned} \tag{26}$$

We also obtain analogous bounds to (25)–(26) for ν, namely

$$(\partial_\nu \mathcal{R}^<)(\nu_{t+1} - \nu_t) + \alpha(\partial_\nu \mathcal{R}^<)(1-\nu_t^2) \leqslant C\alpha^2(\partial_\nu \mathcal{R}^<)^2(1-\nu_t^2)\,, \tag{27}$$

and

$$(\nu_{t+1} - \nu_t)^2 \leqslant C\alpha^2(\partial_\nu \mathcal{R}^<)^2(1-\nu_t^2)\,. \tag{28}$$

Plugging the bounds (25)–(28) into Eq. (22), we obtain the desired inequality

$$\begin{aligned}\mathcal{R}^<(\kappa_{t+1}, \nu_{t+1}) - \mathcal{R}^<(\kappa_t, \nu_t) \leqslant &-\alpha(\partial_\kappa \mathcal{R}^<)^2(1-\kappa_t^2) - \alpha(\partial_\nu \mathcal{R}^<)^2(1-\nu_t^2) \\ &+ C\alpha^2(\partial_\kappa \mathcal{R}^<)^2(1-\kappa_t^2) + C\alpha^2(\partial_\nu \mathcal{R}^<)^2(1-\nu_t^2)\,. \end{aligned}$$

By choosing the step size $\alpha \leqslant \frac{1}{2C}$, this ensures that

$$\mathcal{R}^<(\kappa_{t+1}, \nu_{t+1}) - \mathcal{R}^<(\kappa_t, \nu_t) \leqslant -\frac{\alpha}{2}(\partial_\kappa \mathcal{R}^<)^2(1-\kappa_t^2) - \frac{\alpha}{2}(\partial_\nu \mathcal{R}^<)^2(1-\nu_t^2).$$

This shows that the risk is decreasing along the PGD iterates. Next, introducing $\mathcal{R}^<_{\min} = \min_{(\kappa,\nu)\in[0,1]^2} \mathcal{R}^<(\kappa,\nu)$ and using a telescopic sum, we have, for all $T \geqslant 0$,

$$\begin{aligned}\mathcal{R}^<(\kappa_0, \nu_0) - \mathcal{R}^<_{\min} &\geqslant \mathcal{R}^<(\kappa_0, \nu_0) - \mathcal{R}^<(\kappa_T, \nu_T) \\ &\geqslant \frac{\alpha}{2}\sum_{t=0}^{T-1}\left[(\partial_\kappa \mathcal{R}^<)^2(1-\kappa_t^2) + (\partial_\nu \mathcal{R}^<)^2(1-\nu_t^2)\right]. \end{aligned}$$

Since the left-hand side is finite, and the terms of the sum are nonnegative, we conclude that the series converges as $T \to \infty$. In particular, the generic term $(\partial_\kappa \mathcal{R}^<)^2(1-\kappa_t^2) + (\partial_\nu \mathcal{R}^<)^2(1-\nu_t^2)$ of the series converges to 0 as $t \to \infty$. Therefore, the accumulation points $(\kappa_\infty, \nu_\infty)$ satisfy

$$\begin{cases} \partial_\kappa \mathcal{R}^<(\kappa_\infty, \nu_\infty) = 0 \quad \text{or} \quad \kappa_\infty^2 = 1 \\ \partial_\nu \mathcal{R}^<(\kappa_\infty, \nu_\infty) = 0 \quad \text{or} \quad \nu_\infty^2 = 1. \end{cases}$$

Inspecting identity (23), we observe that the convergence of the general term also implies $\kappa_{t+1} - \kappa_t \to 0$. We obtain similarly that $\nu_{t+1} - \nu_t \to 0$.

B.7 Proof of Proposition 9

Recall that the risk in terms of (κ, ν) is given by

$$\mathcal{R}^<(\kappa,\nu) = \gamma^2 - 2\gamma^2\nu\,\mathrm{erf}\left(\frac{\lambda\sqrt{d/2}\kappa}{\sqrt{1+2\lambda^2\gamma^2}}\right) + \gamma^2\zeta\Big(\lambda\sqrt{\frac{d}{2}}\,\kappa, \lambda^2\gamma^2\Big) + (L-1)\zeta(0,\lambda^2) + \varepsilon^2\,.$$

Then the gradients of $\mathcal{R}^<$ are given by

$$\partial_\kappa\mathcal{R}^<(\kappa,\nu) = -2\gamma^2\lambda\sqrt{\frac{d}{2(1+2\lambda^2\gamma^2)}}\mathrm{erf}'\left(\frac{\lambda\sqrt{d/2}\kappa}{\sqrt{1+2\lambda^2\gamma^2}}\right)\left(\nu - \mathrm{erf}\left(\frac{\lambda\sqrt{d/2}\kappa}{\sqrt{(1+2\lambda^2\gamma^2)(1+4\lambda^2\gamma^2)}}\right)\right)$$

and

$$\partial_\nu\mathcal{R}^<(\kappa,\nu) = -2\gamma^2\mathrm{erf}\left(\frac{\lambda\sqrt{d/2}\kappa}{\sqrt{1+2\lambda^2\gamma^2}}\right).$$

Therefore, the solutions of the system (14) satisfy

$$\begin{cases} -\nu + \mathrm{erf}(c_1\kappa) = 0 \text{ or } \kappa = \pm 1 \\ \kappa = 0 \text{ or } \nu = \pm 1\,, \end{cases}$$

with $c_1 = \frac{\lambda}{\sqrt{4\lambda^2\gamma^2+1}}\sqrt{\frac{d}{2(1+2\lambda^2\gamma^2)}}$. The solutions of this system are

$$(\kappa,\nu) = (0,0) \text{ or } (\kappa,\nu) = (\pm 1, \pm 1)\,.$$

B.8 Proof of Proposition 10

Since $\mathcal{R}^<$ is a smooth function, the extrema of this function on $[-1,1]^2$ are either critical points (admitting null derivatives) or points on the boundary of the square $[-1,1]^2$. Starting with critical points, the only critical point is $(0,0)$, and it is a saddle point. Indeed, the Hessian of $\mathcal{R}^<$ at (0,0) is

$$H_{\mathcal{R}^<}(0,0) = -\frac{4}{\sqrt{\pi}}\gamma^2\lambda\sqrt{\frac{d}{2(1+2\lambda^2\gamma^2)}}\underbrace{\begin{pmatrix} c & 1 \\ 1 & 0 \end{pmatrix}}_{:=M}$$

where $c = -\frac{2\lambda}{\sqrt{\pi(4\lambda^2\gamma^2+1)}}\sqrt{\frac{d}{2(1+2\lambda^2\gamma^2)}} < 0$. Then, as $\det(M) = -1$, the two eigenvalues of $H_{\mathcal{R}^<}(0,0)$ have opposite signs, $(0,0)$ is thus a saddle point. The extrema of $\mathcal{R}^<$ must therefore be on the boundary of the square, which we examine next.

For any $(\kappa,\nu) \in (-1,1)^2$, one has, by inspecting the signs of the gradients given in the proof of Proposition 9,

$$\mathcal{R}^<(1,1) < \mathcal{R}^<(\kappa,1) < \mathcal{R}^<(-1,1) \quad \text{and} \quad \mathcal{R}^<(1,1) < \mathcal{R}^<(1,\nu) < \mathcal{R}^<(1,-1)\,.$$

This shows that the minimum of $\mathcal{R}^<$ on $\{(\kappa,1), \kappa \in [-1,1]\} \cup \{(1,\nu), \nu \in [-1,1]\}$ is reached at $(1,1)$, and the maximum is reached both at $(1,-1)$ and $(-1,1)$, since $\mathcal{R}^<$ is even. Using again evenness of $\mathcal{R}^<$, we conclude that the extrema of $\mathcal{R}^<$ on the whole boundary of the square, and thus on the whole square, are the minimizers $(1,1)$ and $(-1,-1)$, and the maximizers $(1,-1)$ and $(-1,1)$.

B.9 Proof of Proposition 12

We prove the statements of the proposition one by one.

The mapping g is a local diffeomorphism around $(0,0)$, whose Jacobian matrix has one eigenvalue in $(0,1)$ and one eigenvalue in $(1,\infty)$. Consider the Taylor expansion of the first component $g(\kappa,\nu)_1$ of $g(\kappa,\nu)$. Since $\partial_\kappa \mathcal{R}^<(0,0) = 0$, and $\mathcal{R}^<$ is smooth, letting $x = (\kappa,\nu)$, we have $(\partial_\kappa \mathcal{R}^<(\kappa,\nu))^2 = O(\|x\|^2)$. Thus,

$$\begin{aligned} g(\kappa,\nu)_1 &= \frac{\kappa - \alpha(\partial_\kappa \mathcal{R}^<(\kappa,\nu))(1-\kappa^2)}{\sqrt{1+\alpha^2(\partial_\kappa \mathcal{R}^<(\kappa,\nu))^2(1-\kappa^2)}} \\ &= \frac{\kappa - \alpha(\partial_\kappa \mathcal{R}^<(\kappa,\nu))(1-\kappa^2)}{\sqrt{1+O(\|x\|^2)}} \\ &= (\kappa - \alpha(\partial_\kappa \mathcal{R}^<(\kappa,\nu))(1-\kappa^2))\left(1+O(\|x\|^2)\right) \\ &= \kappa - \alpha\partial_\kappa \mathcal{R}^<(\kappa,\nu) + O(\|x\|^2)\,. \end{aligned}$$

Proceeding similarly with the second component of g, we obtain that the Jacobian of g at $(0,0)$ is given by

$$J_g(0,0) = I_2 - \alpha H_{\mathcal{R}^<}(0,0) = I_2 + \alpha \cdot \frac{4}{\sqrt{\pi}}\gamma^2\lambda\sqrt{\frac{d}{2(1+2\lambda^2\gamma^2)}}\underbrace{\begin{pmatrix} c & 1 \\ 1 & 0 \end{pmatrix}}_{=:M},$$

where $c = -\frac{2\lambda}{\sqrt{\pi(4\lambda^2\gamma^2+1)}}\sqrt{\frac{d}{2(1+2\lambda^2\gamma^2)}} < 0$. Since $\det(M) = -1$, one can choose α small enough so that one eigenvalue of $J_g(0,0)$ is strictly between 0 and 1 and the other one is strictly larger than 1. Therefore, $J_g(0,0)$ is invertible, showing that g is a local diffeomorphism around $(0,0)$.

The mapping g is differentiable on $[-1,1]^2$, and its Jacobian is not degenerate. The mapping g is clearly differentiable as a composition of differentiable function. The more delicate part is to show that its Jacobian cannot be degenerate. To show this statement, observe first that, for $x \in [-1,1]^2$, we may write $g(x) = x + \alpha h(x)$, where the first component of h is given by

$$\begin{aligned} h(\kappa,\nu)_1 &= \frac{1}{\alpha}(g(\kappa,\nu)_1 - \kappa) \\ &= \frac{1}{\alpha}\Big(\frac{\kappa - \alpha(\partial_\kappa \mathcal{R}^<(\kappa,\nu))(1-\kappa^2)}{\sqrt{1+\alpha^2(\partial_\kappa \mathcal{R}^<(\kappa,\nu))^2(1-\kappa^2)}} - \kappa\Big) \\ &= \underbrace{\frac{\kappa}{\alpha}\Big(\frac{1}{\sqrt{1+\alpha^2(\partial_\kappa \mathcal{R}^<(\kappa,\nu))^2(1-\kappa^2)}} - 1\Big)}_{=:f_\alpha^{(1)}(\kappa,\nu)} - \underbrace{\frac{(\partial_\kappa \mathcal{R}^<(\kappa,\nu))(1-\kappa^2)}{\sqrt{1+\alpha^2(\partial_\kappa \mathcal{R}^<(\kappa,\nu))^2(1-\kappa^2)}}}_{=:f_\alpha^{(2)}(\kappa,\nu)}\,. \end{aligned}$$

Let us prove that the gradient of $h(\kappa,\nu)_1$ is bounded uniformly over $\alpha \leqslant 1$. The uniform boundedness is clear for the gradient of $f_\alpha^{(2)}$, which writes as a composition of functions with uniformly bounded gradients for $\alpha \leqslant 1$. Moving on to $f_\alpha^{(1)}$ and letting

$$g : \begin{cases} [-1,1]\times[0,B] & \to \mathbb{R} \\ (a,b) & \mapsto \frac{a}{\alpha}\big(\frac{1}{\sqrt{1+\alpha^2 b}} - 1\big) \end{cases}, \quad B = \sup_{(\kappa,\nu)\in[-1,1]^2} (\partial_\kappa \mathcal{R}^<(\kappa,\nu))^2(1-\kappa^2)\,,$$

we observe that $f_\alpha^{(1)}$ is the composition of g with a smooth function independent of α. In particular, it suffices to show the uniform boundedness of ∇g to deduce the one of $\nabla f_\alpha^{(1)}$. We further have, by Lemma 16, and for $\alpha \leqslant 1$,

$$\big|\partial_a g(a,b)\big| = \frac{1}{\alpha}\Big|\frac{1}{\sqrt{1+\alpha^2 b}} - 1\Big| \leqslant \alpha b \leqslant B$$

and

$$\big|\partial_b g(a,b)\big| = \Big|-\frac{\alpha a}{2(1+\alpha^2 b)^{3/2}}\Big| \leqslant \frac{\alpha}{2} \leqslant \frac{1}{2}\,.$$

Therefore, the gradient of $h(\kappa,\nu)_1$ is bounded uniformly over $\alpha \leqslant 1$. Proceeding similarly with the gradient of $h(\kappa,\nu)_2$, we obtain that the Jacobian of $h(\kappa,\nu)$ is uniformly bounded over $\alpha \leqslant 1$. Recall now that $J_g(\kappa,\nu) = I_2 + \alpha J_h(\kappa,\nu)$. Therefore, taking α small enough, we obtain that the eigenvalues of J_g have to be bounded away from zero.

The mapping g is injective. The computation above shows that h is β-Lipschitz continuous with β independent of α (for α small enough). In particular we can choose α such that $\alpha < 1/\beta$. Now, let $x \neq y \in [-1,1]^2$ be such that $g(x) = g(y)$. Then

$$\|x - y\| \leqslant \alpha \|h(x) - h(y)\| \leqslant \alpha\beta \|x - y\| < \|x - y\| \,.$$

This is a contradiction, showing that g is injective.

B.10 Proof of Proposition 13

Recall that $(1,-1)$ and $(-1,1)$ are maxima of the risk $\mathcal{R}^<$ on $[-1,1]^2$ by Proposition 10, and that the value of the risk decreases along the iterates of PGD by Proposition 8. Thus the only possible way to converge to these points is to start the dynamics from them.

The case of the point $(0,0)$ is more delicate. We apply the Center-Stable Manifold theorem (Shub, 1987, Theorem III.7) to g, which is a local diffeomorphism around $(0,0)$ by Proposition 12. This guarantees the existence of a local center-stable manifold $W^{\text{cs}}_{\text{loc}}$, which verifies the following properties. First, its codimension is equal to the number of eigenvalues of $J_g(0,0)$ of magnitude larger than 1, that is, 1, by Proposition 12. Hence it has Lebesgue measure zero. Second, there exists a neighborhood B of 0 such that $\bigcap_{t=0}^{\infty} g^{-t}(B) \subset W^{\text{cs}}_{\text{loc}}$. Then, let W^s be the set of all x which converge to $(0,0)$ under the gradient map g, and take $x \in W^s$. Then there exists a T such that $g^t(x) \in B$ for all $t \geqslant T$. This means that $g^T(x) \in \bigcap_{s=0}^{\infty} g^{-s}(B)$, and thus $g^T(x) \in W^{\text{cs}}_{\text{loc}}$. So, $x \in g^{-T}(W^{\text{cs}}_{\text{loc}})$. We have just shown that

$$W^s \subset \bigcup_{T \geqslant 0} g^{-T}(W^{\text{cs}}_{\text{loc}}) \,.$$

Finally, we prove that the pre-image of sets of measure zero by g^T has measure zero for any $T \geqslant 0$. This shall conclude the proof of the result since countable unions of sets of measure zero have measure zero. To show this, note that g is injective by Proposition 12, and therefore g^T is injective too. This allows to define an inverse g^{-T} of g^T defined on the image of g^T, and the pre-image by g^T of $W^{\text{cs}}_{\text{loc}}$ is exactly the image by g^{-T} of $W^{\text{cs}}_{\text{loc}}$ (intersected with the domain of definition of g^{-T}). Furthermore, by Proposition 12, the Jacobian of g^T is invertible. This guarantees that g^{-T} is differentiable by the inverse function theorem. The conclusion follows by recalling that differentiable functions map sets of measure zero to sets of measure zero.

C Expression of the risk beyond the invariant manifold

In this appendix, we provide an expression of the risk $\mathcal{R}$ beyond the manifold $\mathcal{M}$ that extends the one provided in Lemma 6. This result is not needed to prove Theorem 5, and its proof is more involved that the one of Lemma 6. However, we provide it since it might be relevant to follow-up works that would study the dynamics if not initialized on the invariant manifold $\mathcal{M}$. It is also useful for the numerical simulations (see Appendix E).

Proposition 15. *We have the closed-form expression*

$$\begin{aligned}
&\mathcal{R}^<_\lambda(\kappa,\nu,\theta,\eta,\rho) = \varepsilon^2 + \gamma^2 \\
&- 2\gamma^2\nu \operatorname{erf}\left(\frac{\lambda\sqrt{d/2}\kappa}{\sqrt{1+2\lambda^2\gamma^2}}\right) - 2\lambda\gamma^2\sqrt{\frac{d}{2}}\eta\theta\frac{1}{\sqrt{1+2\lambda^2\gamma^2}}\operatorname{erf}'\left(\frac{\lambda\sqrt{d/2}\kappa}{\sqrt{1+2\lambda^2\gamma^2}}\right) \\
&- \frac{2\lambda^2\gamma^4\eta\rho}{1+2\lambda^2\gamma^2}\operatorname{erf}''\left(\frac{\lambda\sqrt{d/2}\kappa}{\sqrt{1+2\lambda^2\gamma^2}}\right) + (\frac{d}{2}\theta^2+\gamma^2)\zeta\left(\lambda\sqrt{\frac{d}{2}}\,\kappa, \lambda^2\gamma^2\right) \\
&+ \sqrt{\frac{d}{2}}\left(\theta\rho - \frac{\lambda^2\gamma^2\rho^2\kappa}{1+2\lambda^2\gamma^2}\right)\frac{4\lambda\gamma^2}{\sqrt{1+2\lambda^2\gamma^2}}\operatorname{erf}\left(\frac{\lambda\sqrt{d/2}\,\kappa}{\sqrt{(1+4\lambda^2\gamma^2)(1+2\lambda^2\gamma^2)}}\right)\operatorname{erf}'\left(\frac{\lambda\sqrt{d/2}\kappa}{\sqrt{1+2\lambda^2\gamma^2}}\right) \\
&+ \frac{4\lambda^2\gamma^4\rho^2}{\sqrt{\pi}\sqrt{1+4\lambda^2\gamma^2(1+2\lambda^2\gamma^2)}}\operatorname{erf}'\left(-\frac{\lambda\sqrt{d}\kappa}{\sqrt{1+4\lambda^2\gamma^2}}\right) \\
&+ (L-1)\left[\zeta(0,\lambda^2) + \frac{8\lambda^2}{\pi\sqrt{1+4\lambda^2}(1+2\lambda^2)}\rho^2\right] + \frac{4\lambda^2}{(1+2\lambda^2)\pi}(L-1)(L-2)\rho^2
\end{aligned}$$

$$+\frac{4\lambda(L-1)\rho}{\sqrt{(1+2\lambda^2)\pi}}\left(\sqrt{\frac{d}{2}}\theta\,\mathrm{erf}\left(\frac{\lambda\sqrt{d/2}\kappa}{\sqrt{1+2\lambda^2\gamma^2}}\right)+\frac{\lambda\gamma^2\rho}{\sqrt{1+2\lambda^2\gamma^2}}\,\mathrm{erf}'\left(\frac{\lambda\sqrt{d/2}\kappa}{\sqrt{1+2\lambda^2\gamma^2}}\right)\right).$$

Proof. We first recall the notations for the five scalar products that are used throughout this proof.

$$\nu = v^\top v^\star\,,\quad \kappa = k^\top k^\star\,,\quad \theta = v^\top k^\star\,,\quad \eta = k^\top v^\star\,,\quad \rho = k^\top v\,.$$

A first decomposition. We start back from the expression (15) obtained for the risk. By expanding in ξ, then expanding the square, we obtain

$$\begin{aligned}
\mathcal{R}(k,v) &= \mathbb{E}\Big(\Big(X_1^\top v^\star - \sum_{\ell=1}^{L} X_\ell^\top v\,\mathrm{erf}(\lambda X_\ell^\top k)\Big)^2\Big) + \varepsilon^2 \\
&= \mathbb{E}\Big(\Big(X_1^\top v^\star - X_1^\top v\,\mathrm{erf}(\lambda X_1^\top k) - \sum_{\ell=2}^{L} X_\ell^\top v\,\mathrm{erf}(\lambda X_\ell^\top k)\Big)^2\Big) + \varepsilon^2 \\
&= \underbrace{\mathbb{E}\Big(\big(X_1^\top v^\star - X_1^\top v\,\mathrm{erf}(\lambda X_1^\top k)\big)^2\Big)}_{=:R_1} + \underbrace{\sum_{\ell=2}^{L}\mathbb{E}\Big(\big(X_\ell^\top v\,\mathrm{erf}(\lambda X_\ell^\top k)\big)^2\Big)}_{=:R_2} \\
&\quad + \underbrace{\sum_{\ell\neq j\geqslant 2}^{L}\mathbb{E}\Big(X_\ell^\top v\,\mathrm{erf}(\lambda X_\ell^\top k) X_j^\top v\,\mathrm{erf}(\lambda X_j^\top k)\Big)}_{=:R_3} \\
&\quad \underbrace{-2\sum_{\ell=2}^{L}\mathbb{E}\Big(\big(X_1^\top v^\star - X_1^\top v\,\mathrm{erf}(\lambda X_1^\top k)\big) X_\ell^\top v\,\mathrm{erf}(\lambda X_\ell^\top k)\Big)}_{=:R_4} + \varepsilon^2\,.
\end{aligned}$$

Computation of R_1. By expanding the square,

$$\begin{aligned}
&\mathbb{E}\Big(\big(X_1^\top v^\star - X_1^\top v\,\mathrm{erf}(\lambda X_1^\top k)\big)^2\Big) \\
&\qquad = \mathbb{E}\Big(\big(X_1^\top v^\star\big)^2\Big) - 2\mathbb{E}\Big(X_1^\top v^\star X_1^\top v\,\mathrm{erf}(\lambda X_1^\top k)\Big) + \mathbb{E}\Big(\big(X_1^\top v\,\mathrm{erf}(\lambda X_1^\top k)\big)^2\Big)\,.
\end{aligned}$$

These three terms are computed hereafter. First we have

$$\mathbb{E}\Big(\big(X_1^\top v^\star\big)^2\Big) = \Big(\mathbb{E}\big(X_1^\top v^\star\big)\Big)^2 + \mathrm{Var}\big(X_1^\top v^\star\big) = \big(\sqrt{\frac{d}{2}}(k^\star)^\top v^\star\big)^2 + \gamma^2 = \gamma^2\,.$$

Second,

$$\begin{aligned}
&\mathbb{E}\Big(X_1^\top v^\star X_1^\top v\,\mathrm{erf}(\lambda X_1^\top k)\Big) \\
&\quad = \mathbb{E}\left[\left(\sqrt{\frac{d}{2}}(k^\star)^\top v^\star + Z_1\right)\left(\sqrt{\frac{d}{2}}(k^\star)^\top v + Z_2\right)\,\mathrm{erf}\left(\lambda\sqrt{\frac{d}{2}}(k^\star)^\top k + \lambda Z_3\right)\right], \\
&\quad = \mathbb{E}\left[Z_1\left(\sqrt{\frac{d}{2}}\theta + Z_2\right)\,\mathrm{erf}\left(\lambda\sqrt{\frac{d}{2}}\kappa + \lambda Z_3\right)\right],
\end{aligned}$$

with

$$\begin{pmatrix} Z_1 \\ Z_2 \\ Z_3 \end{pmatrix} \sim \mathcal{N}\left(0, \gamma^2\begin{pmatrix} 1 & v^\top v^\star & k^\top v^\star \\ v^\top v^\star & 1 & v^\top k \\ k^\top v^\star & v^\top k & 1 \end{pmatrix}\right) = \mathcal{N}\left(0, \gamma^2\begin{pmatrix} 1 & \nu & \eta \\ \nu & 1 & \rho \\ \eta & \rho & 1 \end{pmatrix}\right).$$

Recall the multivariate version of Stein's lemma (Stein, 1981), which states that, when $Z, G_1, \ldots, G_p$ are centered and jointly Gaussian, and $\sigma : \mathbb{R}^p \to \mathbb{R}$,

$$\mathbb{E}\left[Z\sigma(G_1,\ldots,G_p)\right] = \sum_{i=1}^{p}\mathrm{Cov}(Z,G_i)\mathbb{E}\left[\partial_i\sigma(G_1,\ldots,G_p)\right]\,.$$

Therefore,

$$
\begin{aligned}
&\mathbb{E}\Big(X_1^\top v^\star X_1^\top v \operatorname{erf}(\lambda X_1^\top k)\Big) \\
&= \gamma^2\nu\mathbb{E}\left[\operatorname{erf}\left(\lambda\sqrt{\frac{d}{2}}\kappa + \lambda Z_3\right)\right] + \lambda\gamma^2\eta\mathbb{E}\left[\left(\sqrt{\frac{d}{2}}\theta + Z_2\right)\operatorname{erf}'\left(\lambda\sqrt{\frac{d}{2}}\kappa + \lambda Z_3\right)\right] \\
&= \gamma^2\nu\mathbb{E}\left[\operatorname{erf}\left(\lambda\sqrt{\frac{d}{2}}\kappa + \lambda Z_3\right)\right] + \lambda\gamma^2\sqrt{\frac{d}{2}}\eta\theta\mathbb{E}\left[\operatorname{erf}'\left(\lambda\sqrt{\frac{d}{2}}\kappa + \lambda Z_3\right)\right] \\
&\quad + \lambda^2\gamma^4\eta\rho\mathbb{E}\left[\operatorname{erf}''\left(\lambda\sqrt{\frac{d}{2}}\kappa + \lambda Z_3\right)\right] \\
&= \gamma^2\nu\operatorname{erf}\left(\frac{\lambda\sqrt{d/2}\kappa}{\sqrt{1+2\lambda^2\gamma^2}}\right) + \sqrt{\frac{d}{2}}\frac{\lambda\gamma^2\eta\theta}{\sqrt{1+2\lambda^2\gamma^2}}\operatorname{erf}'\left(\frac{\lambda\sqrt{d/2}\kappa}{\sqrt{1+2\lambda^2\gamma^2}}\right) \\
&\quad + \frac{\lambda^2\gamma^4\eta\rho}{1+2\gamma^2\lambda^2}\operatorname{erf}''\left(\frac{\lambda\sqrt{d/2}\kappa}{\sqrt{1+2\lambda^2\gamma^2}}\right)
\end{aligned}
$$

by using Lemma 18$(i)-(iii)$. Finally, using again Stein's lemma and Lemma 18$(iv)-(vi)$, the computation of the last term is as follows:

$$
\begin{aligned}
&\mathbb{E}\Big[\big(X_1^\top v \operatorname{erf}(\lambda X_1^\top k)\big)^2\Big] \\
&= \mathbb{E}\left[\left(\sqrt{\frac{d}{2}}(k^\star)^\top v + Z_2\right)^2 \operatorname{erf}\left(\lambda\sqrt{\frac{d}{2}}k^\top k^\star + \lambda Z_3\right)^2\right] \\
&= \mathbb{E}\left[\frac{d}{2}\theta^2\operatorname{erf}^2\left(\lambda\sqrt{\frac{d}{2}}\kappa + \lambda Z_3\right)\right] + 2\mathbb{E}\left[\sqrt{\frac{d}{2}}\theta Z_2\operatorname{erf}^2\left(\lambda\sqrt{\frac{d}{2}}\kappa + \lambda Z_3\right)\right] \\
&\quad + \mathbb{E}\left[Z_2^2\operatorname{erf}^2\left(\lambda\sqrt{\frac{d}{2}}\kappa + \lambda Z_3\right)\right] \\
&= \frac{d}{2}\theta^2\mathbb{E}\left[\operatorname{erf}^2\left(\lambda\sqrt{\frac{d}{2}}\kappa + \lambda Z_3\right)\right] \\
&\quad + 4\lambda\gamma^2\sqrt{\frac{d}{2}}\theta\rho\mathbb{E}\left[\operatorname{erf}\left(\lambda\sqrt{\frac{d}{2}}\kappa + \lambda Z_3\right)\operatorname{erf}'\left(\lambda\sqrt{\frac{d}{2}}\kappa + \lambda Z_3\right)\right] + \gamma^2\mathbb{E}\left[\operatorname{erf}^2\left(\lambda\sqrt{\frac{d}{2}}\kappa + \lambda Z_3\right)\right] \\
&\quad + 2\lambda\gamma^2\rho\mathbb{E}\left[Z_2\operatorname{erf}\left(\lambda\sqrt{\frac{d}{2}}\kappa + \lambda Z_3\right)\operatorname{erf}'\left(\lambda\sqrt{\frac{d}{2}}\kappa + \lambda Z_3\right)\right] \\
&= (\frac{d}{2}\theta^2 + \gamma^2)\mathbb{E}\left[\operatorname{erf}^2\left(\lambda\sqrt{\frac{d}{2}}\kappa + \lambda Z_3\right)\right] \\
&\quad + 4\lambda\gamma^2\sqrt{\frac{d}{2}}\theta\rho\mathbb{E}\left[\operatorname{erf}\left(\lambda\sqrt{\frac{d}{2}}\kappa + \lambda Z_3\right)\operatorname{erf}'\left(\lambda\sqrt{\frac{d}{2}}\kappa + \lambda Z_3\right)\right] \\
&\quad + 2\lambda^2\gamma^4\rho^2\left(\mathbb{E}\left[\operatorname{erf}\left(\lambda\sqrt{\frac{d}{2}}\kappa + \lambda Z_3\right)\operatorname{erf}''\left(\lambda\sqrt{\frac{d}{2}}\kappa + \lambda Z_3\right)\right] + \mathbb{E}\left[(\operatorname{erf}')^2\left(\lambda\sqrt{\frac{d}{2}}\kappa + \lambda Z_3\right)\right]\right) \\
&= (\frac{d}{2}\theta^2 + \gamma^2)\zeta\left(\lambda\sqrt{\frac{d}{2}}\,\kappa, \lambda^2\gamma^2\right) \\
&\quad + 4\lambda\gamma^2\sqrt{\frac{d}{2}}\theta\rho\mathbb{E}\left[\operatorname{erf}\left(\lambda\sqrt{\frac{d}{2}}\kappa + \lambda Z_3\right)\operatorname{erf}'\left(\lambda\sqrt{\frac{d}{2}}\kappa + \lambda Z_3\right)\right] \\
&\quad + \frac{2\lambda^2\gamma^4\rho^2}{1+2\lambda^2\gamma^2}\left(-2\lambda\sqrt{\frac{d}{2}}\kappa\,\mathbb{E}\left[\operatorname{erf}\left(\lambda\sqrt{\frac{d}{2}}\kappa + \lambda Z_3\right)\operatorname{erf}'\left(\lambda\sqrt{\frac{d}{2}}\kappa + \lambda Z_3\right)\right]\right.
\end{aligned}
$$

$$
\begin{aligned}
&+\mathbb{E}\left[(\mathrm{erf}')^2\left(\lambda\sqrt{\frac{d}{2}}\kappa+\lambda Z_3\right)\right]\Bigg)\\
&=(\frac{d}{2}\theta^2+\gamma^2)\zeta\Big(\lambda\sqrt{\frac{d}{2}}\,\kappa,\lambda^2\gamma^2\Big)\\
&+4\lambda\gamma^2\sqrt{\frac{d}{2}}\left(\theta\rho-\frac{\lambda^2\gamma^2\rho^2\kappa}{1+2\lambda^2\gamma^2}\right)\mathbb{E}\left[\mathrm{erf}\left(\lambda\sqrt{\frac{d}{2}}\kappa+\lambda Z_3\right)\mathrm{erf}'\left(\lambda\sqrt{\frac{d}{2}}\kappa+\lambda Z_3\right)\right]\\
&+\frac{2\lambda^2\gamma^4\rho^2}{1+2\lambda^2\gamma^2}\mathbb{E}\left[(\mathrm{erf}')^2\left(\lambda\sqrt{\frac{d}{2}}\kappa+\lambda Z_3\right)\right]\\
&=(\frac{d}{2}\theta^2+\gamma^2)\zeta\Big(\lambda\sqrt{\frac{d}{2}}\,\kappa,\lambda^2\gamma^2\Big)\\
&+\sqrt{\frac{d}{2}}\left(\theta\rho-\frac{\lambda^2\gamma^2\rho^2\kappa}{1+2\lambda^2\gamma^2}\right)\frac{4\lambda\gamma^2}{\sqrt{1+2\lambda^2\gamma^2}}\mathrm{erf}\Big(\frac{\lambda\sqrt{d/2}\,\kappa}{\sqrt{(1+4\lambda^2\gamma^2)(1+2\lambda^2\gamma^2)}}\Big)\mathrm{erf}'\Big(\frac{\lambda\sqrt{d/2}\,\kappa}{\sqrt{1+2\lambda^2\gamma^2}}\Big)\\
&+\frac{2\lambda^2\gamma^4\rho^2}{1+2\lambda^2\gamma^2}\left(\frac{2}{\sqrt{\pi}\sqrt{1+4\lambda^2\gamma^2}}\mathrm{erf}'\Big(-\frac{\lambda\sqrt{d}\kappa}{\sqrt{1+4\lambda^2\gamma^2}}\Big)\right)
\end{aligned}
$$

by Lemma 18$(iv)-(vi)$.

Computation of R_2. We have

$$
R_2=\sum_{\ell=2}^{L}\mathbb{E}\Big(\big(X_\ell^\top v\,\mathrm{erf}(\lambda X_\ell^\top k)\big)^2\Big)=(L-1)\mathbb{E}\Big(\big(X_2^\top v\,\mathrm{erf}(\lambda X_2^\top k)\big)^2\Big)\,.
$$

Thus, using previous calculations with $\gamma^2=1$, $\theta=0$, and $\kappa=0$, we obtain

$$
\begin{aligned}
R_2&=(L-1)\left[\zeta(0,\lambda^2)+\frac{4\lambda^2}{\sqrt{\pi}\sqrt{4\lambda^2+1}(1+2\lambda^2)}\rho^2\mathrm{erf}'(0)\right]\\
&=(L-1)\left[\zeta(0,\lambda^2)+\frac{8\lambda^2}{\pi\sqrt{4\lambda^2+1}(1+2\lambda^2)}\rho^2\right].
\end{aligned}
$$

Computation of R_3. Regarding the cross-product terms, by independence of the (X_ℓ)'s and Stein's lemma, one gets

$$
\mathbb{E}\Big(X_\ell^\top v\,\mathrm{erf}(\lambda X_\ell^\top k)X_j^\top v\,\mathrm{erf}(\lambda X_j^\top k)\Big)=\mathbb{E}\Big(X_\ell^\top v\,\mathrm{erf}(\lambda X_\ell^\top k)\Big)\mathbb{E}\Big(X_j^\top v\,\mathrm{erf}(\lambda X_j^\top k)\Big)=C^2\rho^2\,,
$$

with $C:=\lambda\mathbb{E}(\mathrm{erf}'(\lambda X_\ell^\top k))=2\lambda/\sqrt{(1+2\lambda^2)\pi}$ by Lemma 18(i). This leads to

$$
R_3=\frac{4\lambda^2}{(1+2\lambda^2)\pi}(L-1)(L-2)\rho^2.
$$

Computation of R_4. We have, again by independence and Stein's lemma,

$$
\begin{aligned}
&\mathbb{E}\Big(\big(X_1^\top v^\star-X_1^\top v\,\mathrm{erf}(\lambda X_1^\top k)\big)X_\ell^\top v\,\mathrm{erf}(\lambda X_\ell^\top k)\Big)\\
&\qquad=\mathbb{E}\Big(X_1^\top v^\star-X_1^\top v\,\mathrm{erf}(\lambda X_1^\top k)\Big)\mathbb{E}\Big(X_\ell^\top v\,\mathrm{erf}(\lambda X_\ell^\top k)\Big)\\
&\qquad=\Big(\sqrt{\frac{d}{2}}(k^\star)^\top v^\star-\mathbb{E}(X_1^\top v\,\mathrm{erf}(\lambda X_1^\top k))\Big)\mathbb{E}\Big(X_\ell^\top v\,\mathrm{erf}(\lambda X_\ell^\top k)\Big)\\
&\qquad=-\mathbb{E}(X_1^\top v\,\mathrm{erf}(\lambda X_1^\top k))\cdot C\rho\\
&\qquad=-\frac{2\lambda\rho}{\sqrt{(1+2\lambda^2)\pi}}\mathbb{E}(X_1^\top v\,\mathrm{erf}(\lambda X_1^\top k))\,.
\end{aligned}
$$

Note that, still using Stein's lemma,

$$
-\mathbb{E}(X_1^\top v\,\mathrm{erf}(\lambda X_1^\top k))
$$

$$
\begin{aligned}
&= -\mathbb{E}(\sqrt{\frac{d}{2}}(k^\star)^\top v \operatorname{erf}(\lambda X_1^\top k)) - \mathbb{E}((X_1^\top v - \sqrt{\frac{d}{2}}(k^\star)^\top v)\operatorname{erf}(\lambda X_1^\top k)) \\
&= -\mathbb{E}(\sqrt{\frac{d}{2}}\theta \operatorname{erf}(\lambda X_1^\top k)) - \operatorname{Cov}\Big(X_1^\top v,\, \operatorname{erf}(\lambda X_1^\top k)\Big) \\
&= -\sqrt{\frac{d}{2}}\theta\, \mathbb{E}(\operatorname{erf}(\lambda X_1^\top k)) - \lambda \operatorname{Cov}\Big(X_1^\top v, X_1^\top k\Big)\mathbb{E}\Big(\operatorname{erf}'(\lambda X_1^\top k)\Big) \\
&= -\sqrt{\frac{d}{2}}\theta \operatorname{erf}\left(\frac{\lambda\sqrt{d/2}\kappa}{\sqrt{1+2\lambda^2\gamma^2}}\right) - \lambda\gamma^2(k^\top v)\frac{1}{\sqrt{1+2\gamma^2\lambda^2}}\operatorname{erf}'\left(\frac{\lambda\sqrt{d/2}\kappa}{\sqrt{1+2\lambda^2\gamma^2}}\right),
\end{aligned}
$$

where we used that $\lambda X_1^\top k \overset{\mathcal{L}}{=} \lambda\sqrt{d/2}\kappa + G$ with $G \sim \mathcal{N}(0, \lambda^2\gamma^2)$, in combination with Lemma 18$(i)-(ii)$. Thus

$$
R_4 = \frac{4\lambda(L-1)\rho}{\sqrt{(1+2\lambda^2)\pi}}\left(\sqrt{\frac{d}{2}}\theta \operatorname{erf}\Big(\frac{\lambda\sqrt{d/2}\kappa}{\sqrt{1+2\lambda^2\gamma^2}}\Big) + \frac{\lambda\gamma^2\rho}{\sqrt{1+2\gamma^2\lambda^2}}\operatorname{erf}'\Big(\frac{\lambda\sqrt{d/2}\kappa}{\sqrt{1+2\lambda^2\gamma^2}}\Big)\right).
$$

All in all. Putting everything together, we obtain

$$
\begin{aligned}
&\mathcal{R}(k,v) = \varepsilon^2 \\
&+ \gamma^2 - 2\gamma^2\nu\operatorname{erf}\left(\frac{\lambda\sqrt{d/2}\kappa}{\sqrt{1+2\lambda^2\gamma^2}}\right) - 2\lambda\gamma^2\sqrt{\frac{d}{2}}\eta\theta\frac{1}{\sqrt{1+2\lambda^2\gamma^2}}\operatorname{erf}'\left(\frac{\lambda\sqrt{d/2}\kappa}{\sqrt{1+2\lambda^2\gamma^2}}\right) \\
&- \frac{2\lambda^2\gamma^4\eta\rho}{1+2\lambda^2\gamma^2}\operatorname{erf}''\left(\frac{\lambda\sqrt{d/2}\kappa}{\sqrt{1+2\lambda^2\gamma^2}}\right) + (\frac{d}{2}\theta^2+\gamma^2)\zeta\Big(\lambda\sqrt{\frac{d}{2}}\,\kappa, \lambda^2\gamma^2\Big) \\
&+ \sqrt{\frac{d}{2}}\left(\theta\rho - \frac{\lambda^2\gamma^2\rho^2\kappa}{1+2\lambda^2\gamma^2}\right)\frac{4\lambda\gamma^2}{\sqrt{1+2\lambda^2\gamma^2}}\operatorname{erf}\left(\frac{\lambda\sqrt{d/2}\,\kappa}{\sqrt{(1+4\lambda^2\gamma^2)(1+2\lambda^2\gamma^2)}}\right)\operatorname{erf}'\left(\frac{\lambda\sqrt{d/2}\,\kappa}{\sqrt{1+2\lambda^2\gamma^2}}\right) \\
&+ \frac{4\lambda^2\gamma^4\rho^2}{\sqrt{\pi}\sqrt{1+4\lambda^2\gamma^2(1+2\lambda^2\gamma^2)}}\operatorname{erf}'\Big(-\frac{\lambda\sqrt{d}\kappa}{\sqrt{1+4\lambda^2\gamma^2}}\Big) \\
&+ (L-1)\left[\zeta(0,\lambda^2) + \frac{8\lambda^2}{\pi\sqrt{1+4\lambda^2}(1+2\lambda^2)}\rho^2\right] + \frac{4\lambda^2}{(1+2\lambda^2)\pi}(L-1)(L-2)\rho^2 \\
&+ \frac{4\lambda(L-1)\rho}{\sqrt{(1+2\lambda^2)\pi}}\left(\sqrt{\frac{d}{2}}\theta \operatorname{erf}\Big(\frac{\lambda\sqrt{d/2}\kappa}{\sqrt{1+2\lambda^2\gamma^2}}\Big) + \frac{\lambda\gamma^2\rho}{\sqrt{1+2\lambda^2\gamma^2}}\operatorname{erf}'\Big(\frac{\lambda\sqrt{d/2}\kappa}{\sqrt{1+2\lambda^2\gamma^2}}\Big)\right).
\end{aligned}
$$

This concludes the proof. □

D TECHNICAL RESULTS

This section gathers formulas that are useful in the proofs, in particular regarding expectation of functions of Gaussian random variables involving erf.

Lemma 16. *For $u \geqslant 0$,*

$$
\Big|\frac{1}{\sqrt{1+u}} - 1\Big| \leqslant u\,.
$$

Proof. The argument of the absolute value is non-positive for $u \geqslant 0$, hence we need to show that

$$
f(u) := 1 - \frac{1}{\sqrt{1+u}} - u
$$

is non-positive for $u \geqslant 0$. Just note that

$$
f(0) = 0 \quad \text{and} \quad f'(u) = \frac{1}{(1+u)^{3+2}} - 1 \leqslant 0\,.
$$

□

Recall that the erf function is defined on $\mathbb{R}$ as

$$\mathrm{erf}(u) = \frac{2}{\sqrt{\pi}} \int_0^u e^{-t^2} dt \,.$$

Lemma 17 (Properties of the erf function)**.** *We have*

$$\begin{aligned}
\mathrm{erf}'(u) &= \frac{2}{\sqrt{\pi}} e^{-u^2} \,, \\
\mathrm{erf}''(u) &= -\frac{4}{\sqrt{\pi}} u e^{-u^2} = -2u\mathrm{erf}'(u) \,, \\
|\mathrm{erf}(u)| &\leqslant \frac{2}{\sqrt{\pi}} |u| \,.
\end{aligned}$$

Proof. The first two statements are clear by usual differentiation rules. Regarding the last statement, since erf is an odd function, it is sufficient to prove the statement for $u \geqslant 0$. Moreover, erf is concave on $[0, \infty)$, so we get, for $u \geqslant 0$,

$$|\mathrm{erf}(u)| = |\mathrm{erf}(u) - \mathrm{erf}(0)| \leqslant \mathrm{erf}'(0) u = \frac{2}{\sqrt{\pi}} u \,,$$

which concludes the proof. □

Lemma 18. *Let $G \sim \mathcal{N}(0, \gamma^2)$. For $t \in \mathbb{R}$,*

(i) $\mathbb{E}\left[\mathrm{erf}'(t+G)\right] = \frac{1}{\sqrt{1+2\gamma^2}} \mathrm{erf}'\Big(\frac{t}{\sqrt{1+2\gamma^2}}\Big)$.

(ii) $\mathbb{E}\left[\mathrm{erf}(t+G)\right] = \mathrm{erf}\Big(\frac{t}{\sqrt{1+2\gamma^2}}\Big)$.

(iii) $\mathbb{E}\left[\mathrm{erf}''(t+G)\right] = \frac{1}{1+2\gamma^2} \mathrm{erf}''\Big(\frac{t}{\sqrt{1+2\gamma^2}}\Big)$.

(iv) $\mathbb{E}\left[(\mathrm{erf}')^2(t+G)\right] = \frac{2}{\sqrt{\pi}\sqrt{1+4\gamma^2}} \mathrm{erf}'\Big(-\frac{\sqrt{2}t}{\sqrt{1+4\gamma^2}}\Big)$.

(v) $(1+2\gamma^2)\mathbb{E}[\mathrm{erf}(t+G)\mathrm{erf}''(t+G)] = -2t\,\mathbb{E}[\mathrm{erf}(t+G)\mathrm{erf}'(t+G)] - 2\gamma^2 \mathbb{E}[(\mathrm{erf}'(t+G))^2]$.

(vi) $\mathbb{E}\left[\mathrm{erf}(t+G)\mathrm{erf}'(t+G)\right] = \frac{1}{\sqrt{1+2\gamma^2}} \mathrm{erf}\Big(\frac{t}{\sqrt{(1+4\gamma^2)(1+2\gamma^2)}}\Big) \mathrm{erf}'\Big(\frac{t}{\sqrt{1+2\gamma^2}}\Big)$.

This lemma reveals the importance of choosing the erf function as the component-wise nonlinearity: there are closed-form formulas for the expectation of erf and its derivatives applied to Gaussian random variables. Extending the results to any nonlinear, bounded, increasing, equal to 0 at 0, and differentiable activation function is an interesting next step.

Proof. *(i)* By Lemma 17,

$$\begin{aligned}
\mathbb{E}\left[\mathrm{erf}'(t+G)\right] &= \frac{\sqrt{2}}{\pi\gamma} \int e^{-(t+g)^2} e^{-\frac{g^2}{2\gamma^2}} \mathrm{d}g \\
&= \frac{\sqrt{2}}{\pi\gamma} \int e^{-\frac{g^2}{c}} e^{-2gt} e^{-t^2} \mathrm{d}g \qquad \text{for} \quad c := \frac{2\gamma^2}{1+2\gamma^2} \\
&= \frac{\sqrt{2}}{\pi\gamma} \int e^{-\frac{(g+ct)^2}{c} + ct^2 - t^2} \mathrm{d}g \\
&= \frac{\sqrt{2}}{\pi\gamma} e^{-t^2(1-c)} \underbrace{\int e^{-\frac{(g+ct)^2}{c}} \mathrm{d}g}_{=\sqrt{\pi c}} \\
&= \frac{2}{\sqrt{\pi(1+2\gamma^2)}} \exp\left(-t^2\left(1 - \frac{2\gamma^2}{1+2\gamma^2}\right)\right)
\end{aligned}$$

$$= \frac{2}{\sqrt{\pi}\sqrt{1+2\gamma^2}} \exp\left(-\frac{t^2}{1+2\gamma^2}\right).$$

(ii) By (i),

$$\begin{aligned}
\mathbb{E}\left[\mathrm{erf}(t+G)\right] &= \int_{-\infty}^{t} \mathbb{E}\left[\mathrm{erf}'(s+G)\right] ds \\
&= \int_{-\infty}^{t} \frac{2}{\sqrt{\pi}\sqrt{1+2\gamma^2}} \exp\left(-\frac{s^2}{1+2\gamma^2}\right) ds \\
&= \int_{-\infty}^{t/\sqrt{1+2\gamma^2}} \frac{2}{\sqrt{\pi}} \exp\left(-u^2\right) ds \\
&= \mathrm{erf}\Big(\frac{t}{\sqrt{1+2\gamma^2}}\Big).
\end{aligned}$$

(iii) By Lemma 17, and following the same steps as in (i),

$$\begin{aligned}
\mathbb{E}\left[\mathrm{erf}''(t+G)\right] &= -\frac{2\sqrt{2}}{\sqrt{\pi}\gamma} \int (t+g) e^{-(t+g)^2} e^{-\frac{g^2}{2\gamma^2}} dg \\
&= -\frac{2\sqrt{2}}{\pi\gamma} e^{-t^2(1-c)} \int (t+g) e^{-\frac{(g+ct)^2}{c}} dg \\
&= -\frac{2\sqrt{2}}{\pi\gamma} e^{-t^2(1-c)} \Big(t\sqrt{\pi c} + \sqrt{\pi c}\mathbb{E}(\mathcal{N}(-ct, \frac{c}{2}))\Big) \\
&= -\frac{2\sqrt{2c}}{\sqrt{\pi}\gamma} e^{-t^2(1-c)} (t-ct) \\
&= -\frac{4}{\sqrt{\pi(1+2\gamma^2)}} e^{-t^2(1-c)} \frac{1}{1+2\gamma^2} t \\
&= -\frac{4t}{\sqrt{\pi}(1+2\gamma^2)^{3/2}} \exp\left(-\frac{t^2}{1+2\gamma^2}\right).
\end{aligned}$$

(iv) By Lemma 17,

$$\begin{aligned}
\mathbb{E}\left[(\mathrm{erf}')^2(t+G)\right] &= \frac{1}{\sqrt{2\pi}\gamma} \int (\mathrm{erf}')^2(t+g) e^{-\frac{g^2}{2\gamma^2}} \,\mathrm{d}g \\
&= \frac{2\sqrt{2}}{\gamma\pi^{3/2}} \int e^{-2(t+g)^2} e^{-\frac{g^2}{2\gamma^2}} \,\mathrm{d}g \\
&= \frac{2\sqrt{2}}{\gamma\pi^{3/2}} \int e^{-\frac{g^2}{2\Gamma^2}} e^{-4gt} e^{-2t^2} \,\mathrm{d}g \qquad \text{with} \quad \Gamma^2 := \gamma^2/(1+4\gamma^2) \\
&= \frac{2\sqrt{2}}{\gamma\pi^{3/2}} \int e^{-\frac{(g+4\Gamma^2 t)^2}{2\Gamma^2}} e^{8\Gamma^2 t^2} e^{-2t^2} \,\mathrm{d}g \\
&= \frac{2\sqrt{2}}{\gamma\pi^{3/2}} e^{-2t^2(1-4\Gamma^2)} \int e^{-\frac{(g+4\Gamma^2 t)^2}{2\Gamma^2}} \,\mathrm{d}g \\
&= \frac{2\sqrt{2}}{\gamma\pi^{3/2}} e^{-2t^2(1-4\Gamma^2)} \sqrt{2\pi}\Gamma \\
&= \frac{4}{\pi\sqrt{1+4\gamma^2}} \exp\left(-\frac{2t^2}{1+4\gamma^2}\right).
\end{aligned}$$

(v) We use Lemma 17 and then Stein's lemma:

$$\begin{aligned}
&\mathbb{E}[\mathrm{erf}(t+G)\mathrm{erf}''(t+G)] \\
&\qquad = -2\mathbb{E}\left[(t+G)\mathrm{erf}(t+G)\mathrm{erf}'(t+G)\right]
\end{aligned}$$

$$
\begin{aligned}
&= -2t\mathbb{E}\left[\mathrm{erf}(t+G)\mathrm{erf}'(t+G)\right] - 2\mathbb{E}\left[G\mathrm{erf}(t+G)\mathrm{erf}'(t+G)\right] \\
&= -2t\mathbb{E}\left[\mathrm{erf}(t+G)\mathrm{erf}'(t+G)\right] \\
&\quad - 2\gamma^2\left(\mathbb{E}\left[\mathrm{erf}'(t+G)^2\right] + \mathbb{E}\left[\mathrm{erf}(t+G)\mathrm{erf}''(t+G)\right]\right) .
\end{aligned}
$$

Reordering terms, this gives the desired equation.

(vi) We define the function

$$
f(t) = \mathbb{E}\left[\mathrm{erf}(t+G)\mathrm{erf}'(t+G)\right] .
$$

Then, using Lemma 18(v), we have

$$
\begin{aligned}
f'(t) &= \mathbb{E}\left[\mathrm{erf}'(t+G)^2\right] + \mathbb{E}\left[\mathrm{erf}(t+G)\mathrm{erf}''(t+G)\right] \\
&= \mathbb{E}\left[\mathrm{erf}'(t+G)^2\right] - \frac{2t}{1+2\gamma^2}\,\mathbb{E}\left[\mathrm{erf}(t+G)\mathrm{erf}'(t+G)\right] \\
&\quad - \frac{2\gamma^2}{1+2\gamma^2}\mathbb{E}\left[(\mathrm{erf}'(t+G))^2\right] \\
&= \frac{1}{1+2\gamma^2}\mathbb{E}\left[(\mathrm{erf}'(t+G))^2\right] - \frac{2t}{1+2\gamma^2} f(t) .
\end{aligned}
$$

We solve this differential equation by the method of variation of parameters: we have

$$
\frac{\mathrm{d}}{\mathrm{d}t}\left(f(t)e^{t^2/(1+2\gamma^2)}\right) = \frac{1}{1+2\gamma^2}\mathbb{E}\left[(\mathrm{erf}'(t+G))^2\right] e^{t^2/(1+2\gamma^2)} .
$$

We use Lemmas 17 and 18(iv):

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\left(f(t)e^{t^2/(1+2\gamma^2)}\right) &= \frac{2}{\sqrt{\pi}}\frac{1}{(1+2\gamma^2)\sqrt{1+4\gamma^2}}\mathrm{erf}'\left(-\frac{\sqrt{2}t}{\sqrt{1+4\gamma^2}}\right) e^{t^2/(1+2\gamma^2)} \\
&= \frac{4}{\pi}\frac{1}{(1+2\gamma^2)\sqrt{1+4\gamma^2}} e^{-2t^2/(1+4\gamma^2)} e^{t^2/(1+2\gamma^2)} \\
&= \frac{4}{\pi}\frac{1}{(1+2\gamma^2)\sqrt{1+4\gamma^2}} \exp\left(-\frac{t^2}{(1+2\gamma^2)(1+4\gamma^2)}\right) \\
&= \frac{2}{\sqrt{\pi}}\frac{1}{(1+2\gamma^2)\sqrt{1+4\gamma^2}}\mathrm{erf}'\left(\frac{t}{\sqrt{(1+2\gamma^2)(1+4\gamma^2)}}\right) .
\end{aligned}
$$

As the distribution of G is symmetric and erf is an odd function, we have that $f(0) = \mathbb{E}\left[\mathrm{erf}(G)\mathrm{erf}'(G)\right] = 0$. Thus integrating the above derivative, we obtain

$$
\begin{aligned}
f(t)e^{t^2/(1+2\gamma^2)} &= \frac{2}{\sqrt{\pi}}\frac{1}{(1+2\gamma^2)\sqrt{1+4\gamma^2}}\int_0^t \mathrm{d}s\,\mathrm{erf}'\left(\frac{s}{\sqrt{(1+2\gamma^2)(1+4\gamma^2)}}\right) \\
&= \frac{2}{\sqrt{\pi}}\frac{1}{\sqrt{1+2\gamma^2}}\mathrm{erf}\left(\frac{t}{\sqrt{(1+2\gamma^2)(1+4\gamma^2)}}\right) .
\end{aligned}
$$

Using again Lemma 17, we obtain the claimed result:

$$
f(t) = \frac{1}{\sqrt{1+2\gamma^2}}\mathrm{erf}'\left(\frac{t}{\sqrt{1+2\gamma^2}}\right)\mathrm{erf}\left(\frac{t}{\sqrt{(1+2\gamma^2)(1+4\gamma^2)}}\right) .
$$

□

E Experimental details and additional results

Our code is available at

https://github.com/PierreMarion23/single-location-regression

We use the Transformers (Wolf et al., 2020) and scikit-learn (Pedregosa et al., 2011) libraries for the experiment of Section 2, and JAX (Bradbury et al., 2018) for the experiment of Section 5. All experiments run in a short time (less than one hour) on a standard laptop.

E.1 EXPERIMENT OF SECTION 2 (NLP MOTIVATIONS)

Data generation. We use synthetically-generated data for this experiment. To create our train set, we generate sentences according to the patterns

```
The city is [SENTIMENT ADJ]. [PRONOUN] [COLOR ADJ] [ANIMAL] is
[ADV] [SENTIMENT ADJ].
```

and

```
The city is [SENTIMENT ADJ]. [PRONOUN] [SENTIMENT ADJ] [ANIMAL] is
[ADV] [COLOR ADJ].
```

where `ADJ` stands for adjective and `ADV` for adverb. Note that the difference between the two patterns is that the locations of the sentiment and of the color adjectives are swapped. Each element between brackets corresponds to a word, which can take a few different values that are chosen manually. For instance, some possible sentiment adjectives are `nice`, `clean`, `cute`, `delightful`, `mean`, `dirty`, or `nasty`. A possible value for some words is $\emptyset$, meaning that we remove the word from the sentence, which creates more variety in sentence length. By doing the Cartesian product over the possible values of each word in brackets, we generate in this way a large number of examples. Then, the label associated to each example depends solely on the sentiment adjective appearing in the *second* sentence. For instance, the words `nice`, `clean`, `cute`, or `delightful` are associated to a label $+1$, while the words `delightful`, `mean`, and `dirty` are associated to a label -1.

We now explain how the test sets are generated. We generate four test sets in order to assess the robustness of the model to various out-of-distribution changes. The baseline test set uses the same sentence patterns and the same sentiment adjectives as in the training set, but other words in the example (e.g., animals, adverb) are different. In particular, a given sentence cannot appear both in the train set and in the test set. Then, we generate another test set by using sentiment adjectives that are not present in the training set. We emphasize that the sentiment adjective fully determines the label, so using unseen adjectives at test time makes the task significantly harder. The third test set uses the same adjectives as in the train set, but another sentence pattern, namely

```
Hello, how are you?  Good evening, [PRONOUN] [COLOR ADJ] [ANIMAL]
is [ADV] [SENTIMENT ADJ].
```

Finally, the fourth test set combines a different sentence pattern and unseen adjectives. The size of the datasets is given in the table below. All datasets have the same number of $+1$ and -1 labels.

Name	Number of examples
Train set	15552
Test set	4608
Test w. OOD tokens	3072
Test w. OOD structure	144
Test w. OOD structure+tokens	96

Table 1: Size of the generated datasets.

Model. We recall that there exists several families of Transformer architectures, which in particular are not all best suited for sequence classification. An appropriate family is called encoder-only Transformer, and a foremost example is BERT (Devlin et al., 2019). We refer to Phuong & Hutter (2022) for an introductory discussion of Transformer architectures and associated algorithms. Here, we use a pretrained BERT model from the Hugging Face Transformers library (Wolf et al., 2020), with the default configuration, namely `bert-base-uncased`. The model has 110M parameters, 12 layers, the tokens have dimension $d = 768$, and each attention layer has 12 heads. It was pretrained by masked language modeling, namely some tokens in the input are hidden, and the model learns to predict the missing tokens. We refer to Devlin et al. (2019) for details on the architecture and pretraining procedure. We do not perform any fine-tuning on the model.

Experiment design. Our experiment consists in performing logistic regression on embeddings of [CLS] tokens in the hidden layers of the pretrained BERT model, where we recall that the [CLS] token is a special token added to the beginning of each input sequence. This is a particular case of the so-called *linear probing*, which is a common technique in the field of LLMs interpretability. More precisely, let ℓ denote a layer index between 0 and 12, where the index 0 corresponds to the input to the model (after tokenization and embedding in $\mathbb{R}^d$). Then, for each value of $\ell \in \{0, \dots, 12\}$, we train a logistic regression classifier, where, for each example, the input to the classifier is the embedding of the [CLS] token at layer ℓ (that is, a d-dimensional vector), and the label is simply the label of the sentence as described above.

Results. For $\ell = 0$ (blue bar in Figure 1b), the embedding of [CLS] is a fixed vector that does not depend on the rest of the sequence, so the classifier has a pure-chance accuracy of 50%. However, as soon as $\ell > 0$, thanks to the attention mechanism, the [CLS] token contains information about the sequence. We report in Figure 1b the average accuracy over $\ell \in \{1, \dots, 12\}$ for the train set (in orange) and the test sets (in green). We observe that the information contained in the [CLS] token is actually very rich, since logistic regression achieves a perfect accuracy of 100% in the train set. In other words, the data fed to the classifier is linearly separable. We emphasize that the size of the train set is significantly larger than the ambient dimension d, so it is far from trivial that this procedure would yield a linearly-separable dataset. Therefore, obtaining linearly-separable data demonstrates that *the model constructs a linear representations of the input inside the [CLS] token*. Moving on to the test sets, the accuracy on the baseline test set is very good (95%), which suggests some generalization abilities of the model. The accuracy on the out-of-distribution test sets degrades (between 64% and 75%), but remains largely superior to pure-chance performance. This suggests that the internal representation built by the Transformer model is to some extent universal, in the sense that it is robust to the specifics of the sentence structure and of the word choice.

E.2 Experiment of Section 5 (Gradient descent recovers the oracle predictor)

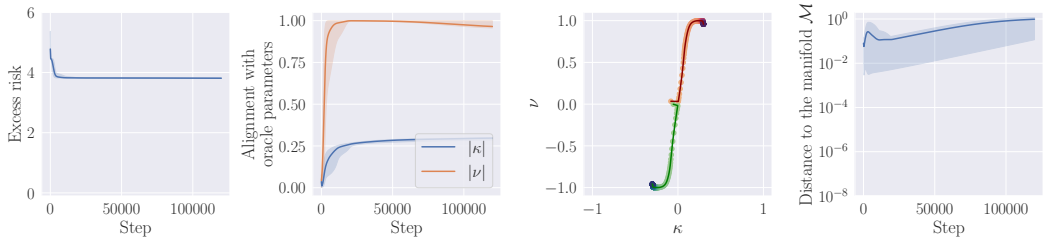
We begin by providing additional results before giving experimental details.

PGD with an initialization on the sphere and constant inverse temperature schedule. As emphasized in Section 5, the dynamics of PGD with a general initialization on $(\mathbb{S}^{d-1})^2$ depend on the choice of the inverse temperature schedule λ_t. The experiment presented in the main text in Figure 4a is for a decreasing schedule $\lambda_t = 1/(1+10^{-4}t)$. We report in Figure 7 results when taking a constant inverse temperature. We observe distinct patterns depending on the value of this parameter. With a large inverse temperature (Figure 7a), we observe that the dynamics in (κ, ν) always escape the neighborhood of 0. Furthermore, the direction $v^\star$ is almost perfectly recovered, i.e., $\nu \approx 1$. However, the value of $k^\star$ is only partially recovered: the dynamics stabilize around $\kappa \approx 0.3$. Moreover, the excess risk plateaus at a high value, while the dynamics stay far away from the manifold $\mathcal{M}$. In the case of a smaller inverse temperature (Figure 7b), the situation is different. We observe that some initializations lead to a convergence to the point $(\kappa, \nu) = (0, 0)$, in which case the dynamics stay far from the manifold $\mathcal{M}$. In other words, there is no recovery of $k^\star$ and $v^\star$. Other initializations lead to perfect recovery of $k^\star$ and $v^\star$. In all cases, the final excess risk is low. Theoretical study of these observations is left for future work.

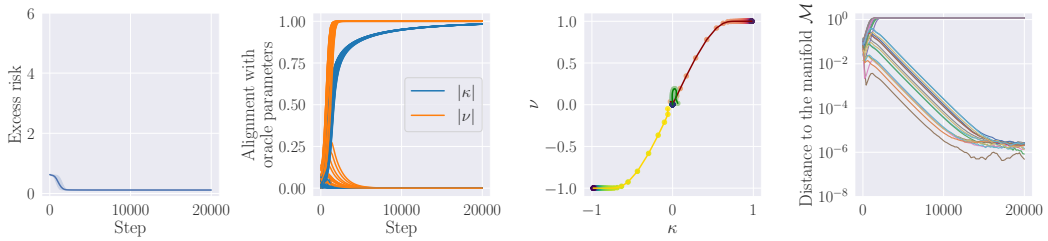
Implementation details. The implementation of the PGD algorithm (10) requires to compute the gradient of the risk. To this aim, we use the formula for the risk given by Proposition 15. Note that all quantities appearing in this expression have explicit derivatives. The only quantity for which this is not directly clear is the function ζ, which needs to be differentiated with respect to its first variable to compute the derivative of the risk with respect to κ. However, recall that $\zeta(t, \gamma^2) := \mathbb{E}\left[\mathrm{erf}^2(t + G)\right]$. Then, by Lemma (18),

$$\begin{aligned}\partial_t \zeta(t, \gamma^2) &= 2\mathbb{E}\left[(\mathrm{erf}\,\mathrm{erf}')(t + G)\right] \\ &= \frac{2}{\sqrt{1 + 2\gamma^2}} \mathrm{erf}\left(\frac{t}{\sqrt{(1 + 4\gamma^2)(1 + 2\gamma^2)}}\right) \mathrm{erf}'\left(\frac{t}{\sqrt{1 + 2\gamma^2}}\right) .\end{aligned}$$

Evaluating ζ itself (and not its derivative) is not required to simulate the dynamics, but is useful for reporting the value of the risk. For this, we also use the formula above, and use numerical quadrature



(a) For $\lambda_t = 0.9$.



(b) For $\lambda_t = 0.1$.

Figure 7: Dynamics of PGD from a random initialization on $(\mathbb{S}^{d-1})^2$, for two iteration-independent values of λ_t. **Left:** Excess risk as a function of the number of steps. **Middle left:** Alignment $|\kappa| = |k^\top k^\star|$ and $|\nu| = |v^\top v^\star|$ with the oracle parameters. **Middle right:** Trajectories of κ and ν in a few repetitions of the experiments. Each repetition corresponds to a color, the end point of each trajectory is in blue. **Right:** Distance to the invariant manifold $\mathcal{M}$. In all plots except the middle right ones, the experiment is repeated 30 times with independent random initializations, and either 95% percentile intervals are plotted or all the curves are plotted. Parameters are $d = 400$, $L = 10$, and $\gamma = \sqrt{1/2}$.

to compute the value of

$$\zeta(t, \gamma^2) = \int_{-\infty}^{t} \partial_s \zeta(s, \gamma^2) ds \,.$$

We report in the figures the value of the excess risk, i.e., the risk $\mathcal{R}_\lambda(k, v) - \varepsilon^2$. To compute the distance to the manifold $\mathcal{M}$, recall that it is defined by

$$\mathcal{M} = \{(k, v) \in \mathbb{S}^{d-1} \times \mathbb{S}^{d-1}, k^\top v^\star = 0, v^\top k^\star = 0, k^\top v = 0\} \,.$$

For a point $(k, v) \in \mathbb{S}^{d-1} \times \mathbb{S}^{d-1}$, its distance to $\mathcal{M}$ is therefore computed as

$$d_{\mathcal{M}}((k, v)) = \sqrt{(k^\top v^\star)^2 + (v^\top k^\star)^2 + (k^\top v)^2} \,.$$

Parameter values. The following table summarizes the value of the parameters in our experiments.

Name	Figure 4a	Figure 4b	Figure 5	Figure 7a	Figure 7b
d	400	400	80	400	400
L	10	10	10	10	10
γ	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$
λ_t	$1/(1 + 10^{-4}t)$	0.1	$2/(1 + 10^{-4}t)$	0.9	0.1
α	$4 \cdot 10^{-3}$	$4 \cdot 10^{-3}$	10^{-3}	10^{-3}	$4 \cdot 10^{-3}$
Number of steps	120k	20k	200k	120k	20k
N. of repetitions	30	30	30	30	30
Batch size	-	-	5	-	-
ε	0	0	0.1	0	0

Table 2: Parameter values for the experiments on recovery of the oracle predictor by gradient descent.

E.3 ADDITIONAL EXPERIMENTS

Transformer layer. The most general formulation of the Transformer layer we consider writes, for $\mathbb{X} \in \mathbb{R}^{L\times d}$,

$$\begin{aligned}\tilde{\mathbb{X}} &= \mathrm{concat}(r, \mathbb{X}) \\ \hat{\mathbb{X}} &= \tilde{\mathbb{X}} + \sum_{h=1}^{H} \mathrm{softmax}\left(\frac{1}{\sqrt{p}} \underbrace{\mathrm{LN}(\tilde{\mathbb{X}})Q_h}_{(L+1)\times p} \underbrace{K_h^\top \mathrm{LN}(\tilde{\mathbb{X}})^\top}_{p\times(L+1)} \right) \underbrace{\mathrm{LN}(\tilde{\mathbb{X}})V_h}_{(L+1)\times p} \underbrace{O_h^\top}_{p\times d}, \\ T(\mathbb{X}) &= \hat{\mathbb{X}} + \mathrm{ReLU}(\hat{\mathbb{X}}W_1^\top + \mathbf{1}b_1^\top)W_2^\top + \mathbf{1}b_2^\top ,\end{aligned} \tag{29}$$

where

- $\mathrm{concat}(r, \mathbb{X}) \in \mathbb{R}^{(L+1)\times d}$ adds a new token at the beginning of the sequence by concatenating $r \in \mathbb{R}^d$ to $X \in \mathbb{R}^{L\times d}$. This token corresponds to the [CLS] or register token (see Section 3 for discussion and references). In all our experiments, $r \in \mathbb{R}^d$ is a vector with i.i.d. Gaussian entries of variance $1/d$, which is not trained;
- LN denotes layer normalization, $\mathrm{softmax}$ denotes row-wise softmax, and $\mathbf{1} \in \mathbb{R}^{L+1}$ is the vector filled with 1;
- the parameters are $Q_h, K_h, V_h, O_h \in \mathbb{R}^{d\times p}$, $W_1 \in \mathbb{R}^{d\times m}$, $b_1 \in \mathbb{R}^m$, $W_2 \in \mathbb{R}^{m\times d}$, and $b_2 \in \mathbb{R}^d$.

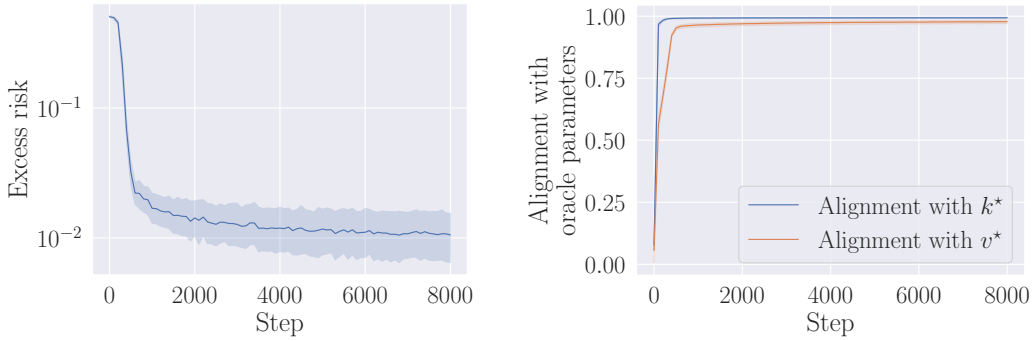
Experiment with single-head Transformer layer on single-location regression. We first consider the case of single-head attention, where $H = 1$ and $p = d$. For ease of notation, we drop the subscripts h in the parameters of the attention layer. We also set O to be the identity matrix. We aim at training the Transformer layer on the single-location regression task, to check that our simplified model is a good description of the Transformer layer. First note that the output of the Transformer layer (29) is a matrix in $\mathbb{R}^{(L+1)\times d}$ while the target of single-location regression is a scalar. Thus, we consider only the first row of $T(\mathbb{X})$, corresponding to the register token, and learn a linear projection of this row to $\mathbb{R}$. In other words, the Transformer layer should learn to store in the register token global information about the sequence, as described in Sections 2 and 3. Overall, letting $\theta \in \mathbb{R}^d$, our risk writes

$$\mathcal{R}(Q, K, V, W_1, b_1, W_2, b_2, \theta) = \mathbb{E}\Big[\big(Y - T(\mathbb{X})_1\theta\big)^2\Big] ,$$

where $(\mathbb{X}, Y)$ are distributed according to the single-location task as described in Section 2. We train using single-pass stochastic gradient descent (meaning that fresh samples are used at each step), for $8,000$ steps with a batch size of 128 and a learning rate of 0.01. The experiment is repeated 20 times with independent random initializations, and 95% percentile intervals are plotted (but are not visible when the variance is too small). Parameters K, V, W_1, W_2 are initialized with Gaussian entries of variance $2/(d_{\text{in}} + d_{\text{out}})$. The bias terms are initialized to 0, as well as the query matrix Q, following a standard recommendation in the literature on signal propagation in Transformer (Yang et al., 2021; He et al., 2023; He & Hofmann, 2024). The output weights θ are initialized with Gaussian entries of variance $1/d^2$, following the mean-field regime (Chizat et al., 2019). Parameters are $L = 10$, $d = p = 80$, $m = 200$, $\varepsilon^2 = 0.01$, $\gamma^2 = 0.5$.

Results are given in Figure 8. We observe in Figure 8a that the Transformer layer is able to solve single-layer regression. Furthermore, as shown by Figure 8b, it does so by encoding in its weights the underlying structure of the problem, namely the oracle parameters $k^\star$ and $v^\star$, as in our simplified model (see Section 5). More precisely, in the case of our model, we showed that the two parameters $k, v \in (\mathbb{R}^d)^2$ converge to $(k^\star, v^\star)$. To make appear the equivalent of k and v in the more complex parametrization (29), we let k be the first left singular vector of K, and $v = V(I + W_1W_2)\theta/\|V(I + W_1W_2)\theta\|$. We check numerically that the weight matrix $QK^\top$ is nearly rank-one after training[3], which validates taking k as the first singular vector of K in the present experiment. It also validates considering vector-valued parameters in our simplified model. The role of the vector k is to select the relevant token among all input tokens, while the vector v describes how successive transformations (the value matrix of the attention layer, the MLP with skip connection,

[3]The ratio between its first and second singular value is of the order of 10^6 at the end of training.



(a) Excess test risk as a function of the number of steps.

(b) Alignment between Transformer parameters and oracle parameters $k^\star$ and $v^\star$. We plot $|k^\top k^\star|$ and $v^\top v^\star$ as a function of the number of steps, where k is the first left singular vector of K, and $v := V(I + W_1W_2)\theta / \|V(I + W_1W_2)\theta\|$.

Figure 8: Training a full Transformer layer on single-location regression. The Transformer layer solves the task, and encodes the structure of the problem in its weights.

and the final linear projection) map this token to the output of the model. We observe that these two vectors align perfectly with $k^\star$ and $v^\star$. This confirms that our simplified model is a good description of how the Transformer layer solves single-location regression.

Multiple-location regression. A natural extension of single-location regression is when the output depends on $s > 1$ tokens instead of just one. This task, which we name multiple-location regression, can be written as

$$Y = \sum_{h=1}^{s} X_{J(h)}^\top v_h^\star + \xi, \tag{30}$$

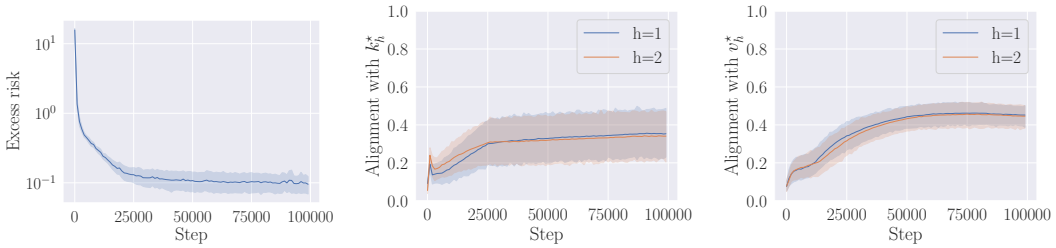
where $J(1), \dots, J(s)$ are latent discrete random variables on $\{1, \dots, L\}$, all different, and such that, conditionally on $J(1), \dots, J(s)$,

$$\begin{cases} X_{J(h)} & \sim & \mathcal{N}\Big(\sqrt{\frac{d}{2}} k_h^\star, \gamma^2 I_d\Big) \\ X_\ell & \sim & \mathcal{N}(0, I_d) \quad \text{for} \quad \ell \notin \{J(1), \dots, J(s)\}\,. \end{cases}$$

Experiment with simplified predictor on multiple-location regression. In accordance with the above, a natural extension of the model presented in the main text is the multi-head predictor

$$T_\lambda^{(k_1, v_1, \dots, k_h, v_h)}(\mathbb{X}) = \sum_{h=1}^{s} \mathrm{erf}\big(\lambda \mathbb{X} k_h\big)^\top \mathbb{X} v_h\,. \tag{31}$$

The hope is that each head (k_h, v_h) should align with one of the oracle directions $(k_h^\star, v_h^\star)$. As a first attempt in investigating this question, we run stochastic PGD in a setup similar to the one presented in Figure 5. We take $s = 2$, the pair $(J(1), J(2))$ takes uniform values among disjoint pairs of indices in $\{1, \dots, L\}$. The directions $(k_1^\star, v_1^\star)$ and $(k_2^\star, v_2^\star)$ are sampled independently uniformly on the sphere, such that $(k_i^\star)^\top v_i^\star = 0$. Parameter values are the same as in Figure 5, except that the number of steps is set to 10^5, the number of repetitions is set to 20, and the inverse temperature λ_t is constant after $2.5 \cdot 10^4$ steps. Results are given in Figure 9. We observe (Figure 9a) that our predictor is able to solve the task. However, the recovery of oracle parameters is only partial, as shown in Figures 9b and 9c: each head partially aligns with the oracle parameters, but the alignment is not perfect. In other words, the model is not well able to separate the signal coming from the different $X_{P(h)}$. This calls for additional research in understanding how attention heads differentiate from each other in order to attend to various signals, and why in our setup the heads are not well-separated.



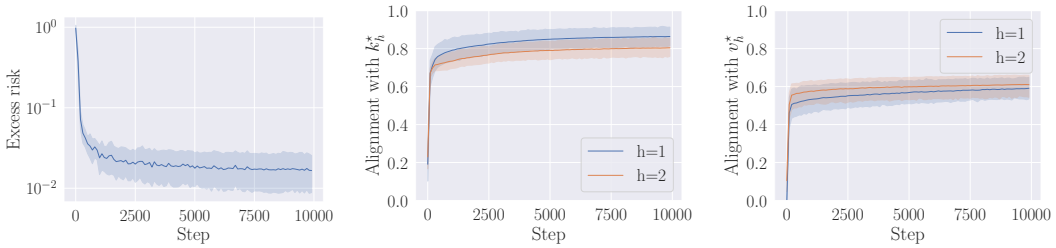
(a) Excess test risk as a function of the number of steps.

(b) Alignment $|k^\top k_h^\star|$ with the oracle parameters.

(c) Alignment $|v^\top v_h^\star|$ with the oracle parameters.

Figure 9: Training the multi-head predictor (31) on the multiple-location regression task (30). The predictor is able to reach a low-risk region. The recovery of oracle parameters by the predictor is partial. In the middle plot, for each repetition and each oracle parameter $k_h^\star$, we look at the end of training which head among k_1 and k_2 is closer to $k_h^\star$, and report the alignment between $k_h^\star$ and that head along training. If the alignment were perfect, this quantity would be close to 1. The same holds for the right plot.

Experiment with multi-head Transformer layer on multiple-location regression. We train a multi-head Transformer layer on the multiple-location regression task (30), taking $H = s = 2$. The data is generated as in the previous experiment. Parameters are as in the experiment for single-head Transformer, except the dimension $p = d/H = 40$, the number of repetitions set to 10, and the learning rate set to 0.02. Mimicking the single-head experiment, we let k_h be the first left singular vector of K_h, and $v_h = V_h O_h^\top (I + W_1 W_2)\theta / \|V_h O_h^\top (I + W_1 W_2)\theta\|$. We also check numerically that all weight matrices $Q_h K_h^\top$ are nearly rank-one after training. Results are reported in Figure 10. The conclusions are similar to the previous experiment: the excess risk is low at the end of training, but we observe partial recovery of the oracle parameters (although the recovery is somewhat better than with the simplified predictor, especially for $k_h^\star$). This suggests that our simplified predictor might be a first good testbed to understand the training dynamics of multi-head Transformer for this task.



(a) Excess test risk as a function of the number of steps.

(b) Alignment $|k^\top k_h^\star|$ with the oracle parameters.

(c) Alignment $v^\top v_h^\star$ with the oracle parameters.

Figure 10: Training the multi-head Transformer layer (29) on the multiple-location regression task (30). The predictor is able to reach a low-risk region. The recovery of oracle parameters by the predictor is partial. For each $h \in \{1, 2\}$, we let k_h be the first left singular vector of K_h, and $v_h = V_h O_h^\top (I + W_1 W_2)\theta / \|V_h O_h^\top (I + W_1 W_2)\theta\|$. In the middle plot, for each repetition and each oracle parameter $k_h^\star$, we look at the end of training which head among k_1 and k_2 is closer to $k_h^\star$, and report the alignment between $k_h^\star$ and that head along training. If the alignment were perfect, this quantity would be close to 1. The same holds for the right plot.

F Further discussion of related models

We begin by discussing some related works on training dynamics of Transformers (Jelassi et al., 2022; Nichani et al., 2024; Wang et al., 2024), to illustrate the originality of our task and predictor. Jelassi et al. (2022) study how (vision) Transformers learn spatial patterns in the data by relying on positional

encodings. This differs significantly from our task that is invariant by token permutation. Further, in their model, the argument of softmax (i.e., a matrix $A \in \mathbb{R}^{L\times L}$) is directly a parameter of the model. This is a radically different structure from the usual attention, and from our setup, where the data appear in the nonlinearity $\sigma(X_\ell^\top k)$. Next, Nichani et al. (2024) explore a task involving a fixed latent causal graph over the positions of the tokens. Here again, positional encodings play a critical role in their analysis, whereas our task is invariant under permutations of the tokens. Moreover, in Nichani et al. (2024), the output is expressed as a function of the last token, with the previous tokens providing the necessary context for this computation. In our setup, however, the output depends on a token whose position varies and must be identified within the context. Closer to our approach is the recent paper by Wang et al. (2024), which also incorporates a notion of token-wise sparsity: the output is computed as the average of a small subset of tokens, where the subset is identified by comparing the positional encodings of each token with that of a reference token. We outline two key differences with our setting. First, we do not make use of a reference token, but instead learn the latent direction $k^\star$ to identify the informative token. Second, in our setting, the tokens also encode an output projection direction $v^\star$ on top of $k^\star$. In other words, our task involves learning a linear regression in addition to identifying the relevant token, which is not the case in Wang et al. (2024).

Besides, we also note that our task shares similarities with multi-index models (McCullagh & Nelder, 1983) and mixtures of linear regressions (De Veaux, 1989). However, our task (P_{learn}) has a more structured nature, involving sequence-valued inputs and incorporating a single-location pattern.

Finally, one could imagine a multi-layer perceptron (MLP) designed specifically for single-location regression, where the weights have a diagonal structure with respect to the sequence index, namely

$$\mathrm{MLP}(X_1, \dots, X_L) = \sum_{\ell=1}^{L} W_2\sigma(W_1 X_\ell + b_1) + b_2.$$

In such a setup, the first layer could learn the projections along k^* and v^*, while the subsequent layer could learn to map these projections to the ouput Y (in a somewhat similar spirit to multi-index models). However, this architecture is far from resembling those used in practice. If we do not assume a diagonal structure and instead use traditional MLPs, the number of parameters must scale at least linearly with the sequence length, which is highly suboptimal and may lead to very slow training. This highlights the efficiency of attention layers, which perform single-location regression with a fixed number of learnable parameters, independent of the input length. We leave a rigorous study of the learning abilities of MLPs in single-location regression for future work.