Rethinking Memorization–Generalization Trade-Off in Generative Models

Jiseok Chae*
Natural Science Research Institute
KAIST
Daejeon, Republic of Korea

Kyuwon Kim*
Department of Mathematical Sciences
KAIST
Daejeon, Republic of Korea

Donghwan Kim
Department of Mathematical Sciences
KAIST
Daejeon, Republic of Korea

{jsch,kkw4053,donghwankim}@kaist.ac.kr

Abstract

Standard generative models face a memorization–generalization trade-off, that is avoiding memorization is considered necessary for generalization. In supervised learning, however, recent studies show that highly overparametrized models can memorize (interpolate) training data while still generalizing well, a phenomenon known as benign overfitting. Motivated by this, we investigate whether generative models can similarly bypass the trade-off and generalize while interpolating. Generative models should minimize the distance between the true data distribution and the distribution induced by mapping the full latent distribution through the generator. But since the true distribution is inaccessible, existing models instead minimize empirical risk with respect to the training distribution. Under this standard formulation, exact empirical-risk minimization forces the generator to produce only training samples. To address this issue, we consider an alternative empirical risk based on presampled latent variables. Our experiments demonstrate that the resulting presampled-latent version of the flow matching model exhibits benign overfitting on standard image benchmarks, such as MNIST and CIFAR-10. As a theoretical proof of concept, we recast generative modeling as a regression problem and extend existing benign overfitting theory to our setting. Together, these results establish, for the first time, that benign overfitting can occur in generative models.

1 Introduction

Generative models, such as generative adversarial networks (GANs) [22], diffusion models [27, 58], and flow matching models [39, 41], have become increasingly influential across various domains, from image and video synthesis [12, 28, 30, 52] to protein and novel material generation [3, 29, 67]. Accordingly, interest in understanding their *generalizability*, the ability to generate diverse realistic data unseen during training, has grown significantly in recent years. A conventional wisdom in generative models is that there exists a trade-off between memorization and generalization [46, 65, 68]. For this reason, regularization techniques, such as noise schedule tuning, weight decay, early stopping, and underparametrization, are typically incorporated in the existing works [7, 17, 26] to suppress memorization and thereby improve generalization.

*Equal contribution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Comparably, in the context of supervised learning, the *memorization–generalization trade-off*, based on the bias–variance trade-off [25], was generally accepted as true, until just a few years ago. Hence, regularization techniques were commonly employed to prevent memorization [34, 62]. However, recent studies have shown that overparametrized supervised learning models trained to achieve zero empirical risk, which necessarily memorize (*i.e.,* interpolate) the training data, can exhibit superior generalization [9, 47]. This empirically observed and theoretically supported phenomenon is now known as *benign overfitting* [8, 44, 61].

In contrast, to the best of our knowledge, there is currently neither theoretical nor even experimental evidence showing that generative models can achieve benign overfitting beyond the conventional trade-off. This motivates the following central question of our study:

Can generative models bypass the memorization-generalization trade-off and exhibit benign overfitting, similar to supervised learning?

We clarify our use of the term *memorization*. While some works in generative modeling literature restrict the term to cases where the model merely produce training samples, we adopt a broader notion: a model *memorizes* the training data if it *interpolates*, or has the ability to reproduce, them. Under this definition, memorization does not preclude generating novel and realistic samples.

To explore our main question, it is first necessary to understand how generative models are trained. Ideally, they should be trained to minimize a distance between the true data distribution and the model distribution obtained by pushing the latent distribution through the generator. In practice, however, the true data distribution is unavailable, so existing methods instead minimize an empirical risk based on the empirical data distribution [36, 69]. This standard approach gives rise to a trade-off between memorization and generalization. In particular, driving the empirical risk to its minimum forces the generated outputs to coincide with training samples, which leads to poor generalization [13, 42, 49].

In view of benign overfitting, one would like the model to generate outputs that are not identical to the training data while still achieving near-zero empirical risk. This suggests that it may not be necessary to use the entire latent distribution during training. Therefore, we consider minimizing an alternative empirical risk using only a fixed subset of vectors presampled from the latent distribution.

While this alternative empirical risk opens the possibility to overcoming the conventional memorization-generalization trade-off, it remains unclear whether it can lead to benign overfitting. As empirical evidence, we train flow matching models using presampled latent variables. Across datasets ranging from synthetic data to standard image benchmarks, we observe that models trained to interpolation can still generalize competitively without merely reproducing the training samples.

As a theoretical proof of concept, we further recast generative modeling as a regression problem enabling us to leverage the theories of benign overfitting in regression, *e.g.,* [8, 57, 61]. Specifically, we study learning the optimal transport (OT) map from latent variables to training data, and investigate how the benign overfitting theory can be extended to this setting. Under the Gaussian universality ansatz [56, 57], we theoretically demonstrate, for the first time in this setting, that generative models can exhibit benign overfitting, beyond the conventional trade-off. Our contributions are as follows.

- In Section 3, as empirical evidence, we train flow matching models using presampled latent variables. Across datasets ranging from synthetic data to standard image benchmarks, we observe that models trained to interpolation can still achieve competitive generalization performance without merely reproducing the training samples.
- In Section 4, as a theoretical proof of concept, we recast generative modeling as a regression task of learning the optimal transport map from the latent distribution to the true data distribution, using presampled latent variables and training data. This reformulation introduces non-i.i.d. noise; nonetheless, we show that the noise remains uniformly bounded with respect to the sample size, which is sufficient for our benign overfitting analysis.
- In Section 5, under the Gaussian universality ansatz, we show that optimal transport-based generative modeling framework introduced in Section 4 can exhibit benign overfitting. Our analysis builds upon existing theories of benign overfitting [8, 57, 61]. In particular, we leverage the theory of [57], which supports the *more is better* principle, where increasing the model size can improve generalization performance. This suggests that this phenomenon may also occur in generative modeling.

2 Background

2.1 Regression: Population and empirical risks

Consider a standard regression setting with a training dataset $\mathcal{D} = \{(\boldsymbol{z}_i, \boldsymbol{x}_i)\}_{i=1,\dots,n} \subset \mathbb{R}^d \times \mathbb{R}^m$. That is, $\{\boldsymbol{z}_i\}_{i=1,\dots,n}$ are i.i.d. samples from a probability distribution ξ over $\mathbb{R}^d$, and

$$\boldsymbol{x}_i = f^*(\boldsymbol{z}_i) + \epsilon_i \tag{1}$$

for some target function $f^* : \mathbb{R}^d \to \mathbb{R}^m$, where ϵ_i denotes the additive noise component. The ideal goal of a regression task is to minimize the *population risk*

$$\mathcal{R}(f) \coloneqq \mathbb{E}_{\boldsymbol{z}\sim\xi}[\|f(\boldsymbol{z}) - f^*(\boldsymbol{z})\|^2]. \tag{2}$$

However, because the target function f^* is typically inaccessible, the population risk cannot be evaluated directly in practice. Hence, the *empirical risk*

$$\hat{\mathcal{R}}(f) \coloneqq \frac{1}{n}\sum_{i=1}^{n}\|f(\boldsymbol{z}_i) - \boldsymbol{x}_i\|^2$$

is used as a proxy to be minimized. The generalization ability of the learned model $\hat{f}$ given the training dataset $\mathcal{D}$, possibly obtained by minimizing the empirical risk $\hat{\mathcal{R}}(f)$, is nonetheless evaluated by the population risk $\mathcal{R}(\hat{f})$. As a measure of average performance, one can consider the expected population risk over all possible training datasets $\mathbb{E}_{\mathcal{D}}[\mathcal{R}(\hat{f})]$.

2.2 Generative modeling: Population and empirical risks

Consider now a generative model setting, where the dataset $\{\boldsymbol{x}_i\}_{i=1,\dots,n}$ consists of n i.i.d. samples from a target distribution ν on $\mathbb{R}^m$. Many modern generative models, such as GANs, diffusion models, and flow matching models, learn a transformation, or a process, that maps a latent distribution ξ to approximate the target distribution ν. Although the subsequent discussion can be extended to stochastic generative processes, we restrict our attention in this section to deterministic generative processes, for which generation is represented by a (deterministic) map G. In this setting, ν is approximated by $G_\sharp\xi$, the *pushforward* of ξ by G.

The *population risk* of a generative model is defined by a divergence D, whose common choices include the Jensen-Shannon divergence [22] and the Wasserstein distances [5, 32], between the generated distribution and the target data distribution

$$\mathcal{R}(G) \coloneqq \mathsf{D}(G_\sharp\xi, \nu). \tag{3}$$

However, ν is not directly accessible in practice. Many approaches thus instead minimize what we refer to as the *semi-discrete empirical risk*

$$\hat{\mathcal{R}}_{\text{semi}}(G) \coloneqq \mathsf{D}(G_\sharp\xi, \hat{\nu}),$$

where $\hat{\nu} \coloneqq \frac{1}{n}\sum_{i=1}^{n}\delta_{\boldsymbol{x}_i}$ is the empirical target distribution with δ denoting the Dirac measure. A few works, *e.g.,* [6, 59], have also explored an alternative, which we call the *fully-discrete empirical risk*

$$\hat{\mathcal{R}}_{\text{fully}}(G) \coloneqq \mathsf{D}(G_\sharp\hat{\xi}, \hat{\nu}) \tag{4}$$

where $\hat{\xi} = \frac{1}{\ell}\sum_{i=1}^{\ell}\delta_{\boldsymbol{z}_i}$ is the empirical latent distribution on the i.i.d. samples $\boldsymbol{z}_1, \dots, \boldsymbol{z}_\ell \sim \xi$. Here, ℓ is the number of presampled latent varaibles, which is typically set to be $\Theta(n)$.

At first glance, the distinction between these two empirical risks may appear subtle. However, the difference is substantial. Notice that achieving near-optimal semi-discrete empirical risk forces $G_\sharp\xi \approx \hat{\nu}$, leading to the model only producing training samples, exhibiting the memorization–generalization trade-off [42]. On the other hand, near-optimal fully-discrete empirical risk does not necessarily induce the trade-off, as we soon discuss.

We note that training generative models such as GANs and flow matching models can be readily cast onto a presampled set of latent variables $\{\boldsymbol{z}_i\}_{i=1,\dots,\ell}$.

3 Empirical Evidence of Benign Overfitting in Flow Matching with Presampled Latent Variables

In this section, we provide empirical evidence demonstrating that benign overfitting can occur in generative models trained with presampled latent variables.

3.1 Flow Matching with Presampled Latent Variables

Flow matching [2, 39, 41] learns a time-dependent vector field $\boldsymbol{v}$ whose induced flow transports samples from the latent distribution ξ to the target distribution ν, ideally by minimizing the objective

$$\mathcal{R}(\boldsymbol{v}) \coloneqq \mathbb{E}_{\boldsymbol{z}\sim\xi,\boldsymbol{x}\sim\nu,t}\big[\|\boldsymbol{v}((1-t)\boldsymbol{z}+t\boldsymbol{x},t)-(\boldsymbol{x}-\boldsymbol{z})\|^2\big]. \tag{5}$$

Unlike the population risk in (3), this risk (5) implicitly measures the discrepancy between the generated distribution and the target data distribution. However, since ν is inaccessible in practice, the standard approach is to minimize the following empirical risk

$$\hat{\mathcal{R}}_{\text{semi}}(\boldsymbol{v}) \coloneqq \frac{1}{n}\sum_{j=1}^{n}\mathbb{E}_{\boldsymbol{z}\sim\xi,t}\big[\|\boldsymbol{v}((1-t)\boldsymbol{z}+t\boldsymbol{x}_j,t)-(\boldsymbol{x}_j-\boldsymbol{z})\|^2\big]. \tag{6}$$

Since minimizing (6) exactly leads to mere memorization [11], we propose to minimize the following fully-discrete empirical risk using presampled latent variables

$$\hat{\mathcal{R}}_{\text{fully}}(\boldsymbol{v}) \coloneqq \frac{1}{n\ell}\sum_{i=1}^{\ell}\sum_{j=1}^{n}\mathbb{E}_{t}\big[\big\|\boldsymbol{v}\big((1-t)\boldsymbol{z}_i+t\boldsymbol{x}_j,t\big)-\big(\boldsymbol{x}_j-\boldsymbol{z}_i\big)\big\|^2\big]. \tag{7}$$

3.2 Measuring Benign Overfitting: Interpolation Error and Reproduction Rate

In order to verify that the approach of presampling the latent variables bypasses the memorization-generalization trade-off and exhibits benign overfitting, we must establish two properties: (i) the trained model should interpolate the training data, but (ii) without merely memorizing (that is, reproducing) them and thus still generalizing well. We therefore introduce metrics for measuring the interpolation error and the reproduction, or mere memorization, rate.

A natural way to verify training data interpolation is to check whether, for each training sample $\boldsymbol{x}_j$, there exists a latent variable whose generated sample is close to $\boldsymbol{x}_j$. In supervised learning, training loss directly measures interpolation, since it quantifies the discrepancy between predictions and targets at given inputs. In flow matching, however, the training loss measures velocity errors along the straight line paths connecting latent–data pairs $(\boldsymbol{z},\boldsymbol{x})$ rather than endpoint discrepancies, and therefore may not faithfully reflect the interpolation error.

Indeed, both objectives (6) and (7) can exhibit nonzero residuals even when the model perfectly interpolates the training data. In particular, straight line paths connecting two distinct latent–data pairs $(\boldsymbol{z},\boldsymbol{x})$ and $(\boldsymbol{z}',\boldsymbol{x}')$ may intersect at time t; *i.e.,* $(1-t)\boldsymbol{z}+t\boldsymbol{x}=(1-t)\boldsymbol{z}'+t\boldsymbol{x}'$. In such cases, the optimal vector field at the intersection point becomes the average of the target velocities $\boldsymbol{x}-\boldsymbol{z}$ and $\boldsymbol{x}'-\boldsymbol{z}'$, thereby leaving an irreducible residual.

We therefore evaluate interpolation by directly measuring endpoint recovery, which itself requires some care. In the fully discrete objective (7), each presampled latent variable is paired with every training data. Consequently, near $t=0$, the vector field around $\boldsymbol{z}_i$ may point toward an averaged direction over multiple target samples, and the learned trajectory from $\boldsymbol{z}_i$ need not follow any particular straight-line path $(1-t)\boldsymbol{z}_i+t\boldsymbol{x}_j$. So, we trace backward from the training data $\boldsymbol{x}_j$ under the learned flow, and use its preimage as the corresponding latent variable for generating $\boldsymbol{x}_j$. We then measure the interpolation error by its mean squared error.

For the reproduction rate, we define it as the proportion of generated samples that reproduce training data under a nearest-neighbor criterion adopted from prior work [23, 65]. Here, a generated sample $\hat{\boldsymbol{x}}$ is counted as reproducing the training data if its nearest training sample is much closer than its second-nearest training sample. Formally, this amounts to $\frac{\|\hat{\boldsymbol{x}}-\boldsymbol{x}_1'\|_2}{\|\hat{\boldsymbol{x}}-\boldsymbol{x}_2'\|_2} < \frac{1}{k}$ for some predefined k, where $\boldsymbol{x}_1'$ and $\boldsymbol{x}_2'$ denote the nearest and second-nearest training samples to $\hat{\boldsymbol{x}}$, respectively. We take

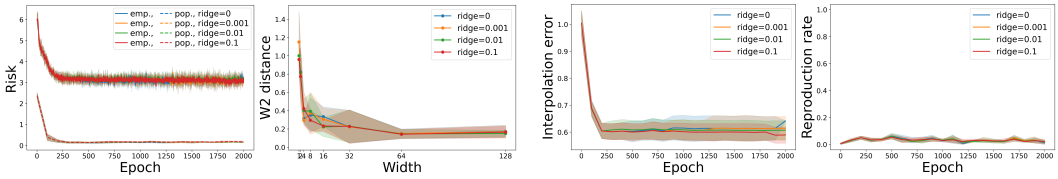
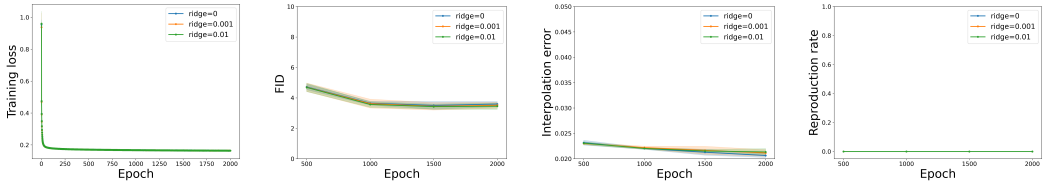
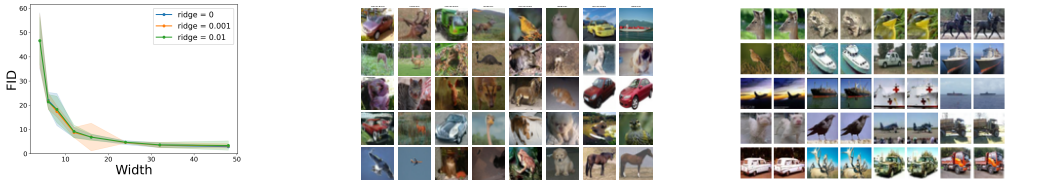


Figure 1: **Flow matching model with presampled latent variables on Gaussian dataset.** We learn a flow from a three-dimensional standard Gaussian latent distribution to a Gaussian target distribution. The first and second figures show the training loss and test $\mathcal{W}_2$ distance over epochs, as well as the test $\mathcal{W}_2$ distance as a function of model size. The third and fourth report the interpolation error and the reproduction rate, both over epochs. See Section B.2 for details.



(a) Training loss (b) FID vs. epochs (c) Interpolation error (d) Reproduction rate



(e) FID vs. model size (f) Generated images (g) Interpolated images

Figure 2: **Flow matching model with presampled latent samples on CIFAR-10.** (a)-(d) show that the training loss converges to a plateau while FID improves, interpolation error becomes small, and reproduction rate remains low. (e) shows a "more is better" trend. (f) shows fresh generations (in the odd-numbered columns) and nearest training samples (in the even-numbered column). (g) shows interpolation, comparing recovered samples with their corresponding target training data. Together, these panels illustrate that the trained flow interpolates the training data while fresh generations do not merely reproduce training data. See Section B.3 for details.

$k = 3$, following [65], as such a choice is reported therein to be empirically aligned with human perception of reproduction.

The next two subsections demonstrate that, when the flow matching with presampled latent variables is trained to almost exactly minimize the fully discrete empirical risk (7), the interpolation error is near zero with near zero reproduction rate, while generalizing well.

3.3 Experiments on low-dimensional data

We first conduct experiments with a low-dimensional Gaussian target distribution ν. Since the target distribution is known a priori, this setting allows us to directly evaluate population risk through the Wasserstein distance. Figure 1 shows that the fully discrete training loss converges to a positive plateau while the model interpolates the training data. Meanwhile, both the Wasserstein distance and the reproduction rate remain small, and generalization improves as the model size increases. These results suggest that benign overfitting, together with the "more is better" phenomenon, can arise even in this low-dimensional generative modeling task.

3.4 Experiments on image data

We next consider the case where ν represents a high dimensional image distribution, such as those of MNIST [14] and CIFAR-10 [35] datasets. In this setting, since the true target distribution is

Table 1: FID, interpolation error, and reproduction rate for flow matching with presampled latent variables. The mean over random seeds are reported.

Dataset	ℓ	Metric	Weight decay		
			0.0	0.001	0.01
MNIST	n	FID ↓	3.47	3.34	2.71
		Interp. error	0.0169	0.0170	0.0169
		Reprod. rate ↓	0.0%	0.0%	0.0%
	$4n$	FID ↓	3.57	3.45	2.85
		Interp. error	0.0170	0.0170	0.0170
		Reprod. rate ↓	0.0%	0.0%	0.0%
CIFAR-10	n	FID ↓	5.65	5.32	4.70
		Interp. error	0.0223	0.0221	0.0225
		Reprod. rate ↓	0.0%	0.0%	0.0%
	$4n$	FID ↓	3.58	3.51	3.45
		Interp. error	0.0220	0.0224	0.0221
		Reprod. rate ↓	0.0%	0.0%	0.0%

unknown, we report the FID as a practical proxy for population risk. The results are summarized in Table 1 and Figure 2. Experimental details are deferred to Section B.3. We observe that flow matching models trained with the objective (7) exhibit benign overfitting: they achieve strong FID scores and interpolate the training data, while the reproduction rate remains low. We further observe that increasing the number of intermediate channels improves FID, even though the training loss reaches essentially the same plateau once the number of channels exceeds 16. Thus, within the tested range, larger models improve generalization without mere memorization.

4 Recasting Generative Modeling with Presampled Latent Variables as Regression via Optimal Transport

Let us now develop a theoretical testbed for demonstrating how generative models can overcome the conventional memorization–generalization trade-off. Specifically, we propose a framework where generative modeling with presampled latent variables is recast as a regression problem via optimal transport. This regression formulation provides the basis for the benign overfitting analysis in the next section. For clarity, we focus on the balanced case $\ell = n$, which captures the essential structure while keeping the presentation simple.

We work under the following mild conditions on the latent and target distributions.

Assumption 4.1. The target distribution ν on $\mathbb{R}^m$ and the latent distribution ξ on $\mathbb{R}^d$ have finite second moments. That is, both $\mathbb{E}_{\boldsymbol{x}\sim\nu}[\boldsymbol{x}\boldsymbol{x}^\top] \coloneqq \boldsymbol{N}$ and $\mathbb{E}_{\boldsymbol{z}\sim\xi}[\boldsymbol{z}\boldsymbol{z}^\top] \coloneqq \boldsymbol{\Xi}$ are finite. In addition, ξ is absolutely continuous with respect to the Lebesgue measure.

We remark that the absolute continuity condition on ξ is not a restrictive one, as the Gaussian distribution is commonly used as latent distributions.

Assumption 4.1 implies that ν has a finite first moment and $\mathbb{E}_{\boldsymbol{x}\sim\nu}\left[\|\boldsymbol{x}\|^2\right] = \mathrm{tr}(\boldsymbol{N}) < \infty$. Analogous statements also hold for ξ. For the proofs, see Section D.1.

4.1 The first step toward bridging generative modeling and regression

As noted in Section 2.1, the goal of regression is to learn a certain target function f from the given data. On the other hand, generative modeling in general admits infinitely many generative mappings that minimize the population risk (3). Therefore, to link generative modeling with regression, a particular choice G^* of the generator should be set. In this paper, we take G^* to be the optimal transport map. To this end, in the rest of this paper we consider the equal-dimensional case, $d = m$, leaving extensions to unequal dimensions for future work, possibly following an embedding approach as in [32].

Given a cost function c, an *optimal transport (OT) map* G^* from ξ to ν is an optimal solution to

$$\inf_{G\,:\,G_\sharp\xi=\nu} \int c(\boldsymbol{z}, G(\boldsymbol{z}))\, d\xi(\boldsymbol{z}). \tag{8}$$

For the quadratic cost $c(\boldsymbol{z}, \boldsymbol{x}) = \|\boldsymbol{z} - \boldsymbol{x}\|^2$, if Assumption 4.1 holds, then the celebrated Brenier's theorem [15] ensures that there uniquely exists an OT map G^*.

Learning the OT map for generative modeling is not new; see, *e.g.*, [32, 43, 53]. Still, although prior works have suggested that such maps can improve practical performance, there has been no theoretical discussion on their generalization behavior when learning from finite training data $\{\boldsymbol{x}_i\}_{i=1,\dots,n}$. In contrast, we systemically analyze how an OT map can be learned from training data $\{\boldsymbol{x}_i\}_{i=1,\dots,n}$ and presampled latent variables $\{\boldsymbol{z}_i\}_{i=1,\dots,n}$, by casting the problem as a regression task.

4.2 Optimal transport map based regression model for generative modeling

Our next step is to formulate a regression model for generative modeling based on the optimal transport map, analogous to (1). This requires first pairing the finite training data with presampled latent variables, mirroring the relationship between training inputs and labels in regression.

Let S_n denote the set of all permutations on $\{1, \dots, n\}$. We naturally choose the discrete optimal transport map $\tilde{\pi} \in S_n$ between them, which is a solution to

$$\min_{\pi \in S_n} \sum_{i=1}^{n} \|\boldsymbol{z}_i - \boldsymbol{x}_{\pi(i)}\|^2.$$

Having paired the dataset as $\{(\boldsymbol{z}_i, \boldsymbol{x}_{\tilde{\pi}(i)})\}_{i=1,\dots,n}$, we can construct a regression model. Intuitively, the construction below can be viewed as a multidimensional analogue of the 1D case, where the discrete OT map between $\hat{\xi}$ and $\hat{\nu}$ pairs the latent samples and training data in nondecreasing order, thereby giving a discrete approximation of the OT map between ξ and ν. Deferring a more detailed discussion of this motivation to Section D.2, we now describe the construction of the regression model.

Given the samples $\boldsymbol{z}_1, \dots, \boldsymbol{z}_n$, it is possible to impose an order on them, for example, by lexicographical ordering. Let $\boldsymbol{z}_{1:n}, \dots, \boldsymbol{z}_{n:n}$ denote such ordered samples. Similarly, we can relabel $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}$ as $\{\boldsymbol{x}_{1:n}, \dots, \boldsymbol{x}_{n:n}\}$ so that $\{(\boldsymbol{z}_{i:n}, \boldsymbol{x}_{i:n})\}_{i=1,\dots,n}$ forms the pairing induced by the discrete OT coupling. This leads to a regression model

$$\boldsymbol{x}_{i:n} = G^*(\boldsymbol{z}_{i:n}) + \check{\boldsymbol{\epsilon}}_i. \tag{9}$$

Notice that the noise term $\check{\boldsymbol{\epsilon}}_i = \boldsymbol{x}_{i:n} - G^*(\boldsymbol{z}_{i:n})$ arises from the sampling process used to obtain $\{\boldsymbol{z}_1, \dots, \boldsymbol{z}_n\}$ and $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}$, rather than from additive label noise on $G^*(\boldsymbol{z}_{i:n})$ as is commonly assumed in regression tasks.

Here, since $\{\boldsymbol{z}_1, \dots, \boldsymbol{z}_n\}$ are i.i.d. samples from ξ, the joint distribution of $\{\boldsymbol{z}_1, \dots, \boldsymbol{z}_n\}$ is invariant under any permutation of the indices. Consequently, each $\boldsymbol{z}_{i:n}$ has probability $\frac{1}{n}$ of coinciding with any particular sample $\boldsymbol{z}_j$, for $j \in \{1, \dots, n\}$. We may thus regard the triplets $\{(\boldsymbol{z}_i, \boldsymbol{x}_{\tilde{\pi}(i)}, \boldsymbol{\epsilon}_i)\}_{i=1,\dots,n}$ as a random permutation of the triplets $\{(\boldsymbol{z}_{i:n}, \boldsymbol{x}_{i:n}, \check{\boldsymbol{\epsilon}}_i)\}_{i=1,\dots,n}$. Exploiting this observation, we can convert (9) into a regression model expressed in terms of $\{\boldsymbol{z}_1, \dots, \boldsymbol{z}_n\}$ and $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}$, to get

$$\boldsymbol{x}_{\tilde{\pi}(i)} = G^*(\boldsymbol{z}_i) + \boldsymbol{\epsilon}_i. \tag{10}$$

We remark that, unlike in standard regression settings, the noise terms $\boldsymbol{\epsilon}_i$ are not independent. As a result, existing benign overfitting theories cannot be applied directly. The next section shows how these theories can nevertheless be adapted to handle this dependence.

5 Benign Overfitting Theory in Generative Modeling via Optimal Transport

In this section, we theoretically demonstrate the benign overfitting of generative models from two perspectives. On one hand, built upon the results by Bartlett et al. [8] and Tsigler and Bartlett [61], we show that benign overfitting can occur in generative models. Specifically, we prove that the expected population risk vanishes asymptotically as the number of data points increases. In addition, based on the work by Simon et al. [57], we show the *more is better* principle also holds in generative models, where overparametrization results in models that achieve a lower population risk.

5.1 Regression with feature maps and the Gaussian universality ansatz

Let $\{\boldsymbol{z}_i\}_{i=1}^n \subset \mathbb{R}^d$ be the presampled latent vectors from ξ, and $\{\boldsymbol{x}_i\}_{i=1}^n \subset \mathbb{R}^m$ be the given dataset, which we assume to be i.i.d. samples from ν. In this section, we consider the OT map-based regression formulation of generative modeling, namely (10). Hence, the OT map G^* from ξ to ν is now our target function. Let $\tilde{\pi}$ be the optimal pairing between $\{\boldsymbol{z}_i\}_{i=1,\dots,n}$ and $\{\boldsymbol{x}_i\}_{i=1,\dots,n}$.

To facilitate theoretical analyses, we consider regression with *random feature* linear models. For some function $\boldsymbol{g} : \mathbb{R}^h \times \mathbb{R}^d \to \mathbb{R}^m$ and a probability distribution ρ on $\mathbb{R}^h$ from which $\boldsymbol{w}_1, \dots, \boldsymbol{w}_p$ are i.i.d. samples, the *random feature map* $\boldsymbol{\psi} : \mathbb{R}^d \to \mathbb{R}^{p\times m}$ is given by

$$\boldsymbol{\psi}(\boldsymbol{z}) = \begin{bmatrix} \boldsymbol{g}(\boldsymbol{w}_1, \boldsymbol{z}) & \cdots & \boldsymbol{g}(\boldsymbol{w}_p, \boldsymbol{z}) \end{bmatrix}^\top. \tag{11}$$

We wish to find a parameter $\boldsymbol{\beta} \in \mathbb{R}^p$, so that the regressor of the form $\hat{G}(\cdot) = \boldsymbol{\psi}(\cdot)^\top \boldsymbol{\beta}$ satisfies $G^* \approx \hat{G}$. Notice that, if $\boldsymbol{g}$ is of the form $\boldsymbol{g}(\boldsymbol{w}, \boldsymbol{z}) = \boldsymbol{\sigma}(\boldsymbol{w}^\top \boldsymbol{z})$ for some coordinatewise nonlinear $\boldsymbol{\sigma}$, then this model is exactly a two-layer neural network with only the last layer being trained.

The parameter $\boldsymbol{\beta}$ is found so that it minimizes the ℓ_2-regularized least squares objective

$$\min_{\boldsymbol{\beta}\,:\, G(\,\cdot\,)=\boldsymbol{\psi}(\,\cdot\,)^\top\boldsymbol{\beta}} \ \sum_{i=1}^n \frac{1}{2}\|G(\boldsymbol{z}_i) - \boldsymbol{x}_{\tilde{\pi}(i)}\|^2 + \frac{\delta}{2}\,\|\boldsymbol{\beta}\|^2. \tag{12}$$

This objective, for $\delta > 0$, yields an optimal solution

$$\hat{\boldsymbol{\beta}}(\delta) = \boldsymbol{\Psi}^\top(\boldsymbol{\Psi}\boldsymbol{\Psi}^\top + \delta \boldsymbol{I})^{-1}\mathbb{x}, \tag{13}$$

where $\boldsymbol{\Psi} \in \mathbb{R}^{nm\times p}$ is a block matrix whose ith block row is $\boldsymbol{\psi}(\boldsymbol{z}_i)^\top$ and $\mathbb{x} \in \mathbb{R}^{nm}$ is a concatenation of $\boldsymbol{x}_{\tilde{\pi}(1)}, \dots, \boldsymbol{x}_{\tilde{\pi}(n)}$. Accordingly, the learned model to be studied is $\hat{G}(\,\cdot\,) = \boldsymbol{\psi}(\,\cdot\,)^\top \hat{\boldsymbol{\beta}}(\delta)$. The definition of the population risk in (2) still applies verbatim, with $\boldsymbol{\psi}$ now understood as the matrix-valued function in (11). In particular, the population risk when using the regressor $\hat{G}$ reads

$$\mathcal{R}(\hat{\boldsymbol{\beta}}) = \mathbb{E}_{\boldsymbol{z}\sim\xi}\big[\big\|\boldsymbol{\psi}(\boldsymbol{z})^\top\hat{\boldsymbol{\beta}} - G^*(\boldsymbol{z})\big\|^2\big]. \tag{14}$$

This population risk provides an upper bound on the squared 2-Wasserstein distance between the generated distribution $\hat{G}_\sharp\xi$ and the data distribution ν; see Appendix F.1. Consequently, if benign overfitting occurs so that the population risk is sufficiently small, it follows that the distance between $\hat{G}_\sharp\xi$ and ν is also small, thereby providing a strong performance guarantee for G^*.

To facilitate a plausible yet theoretical analysis, we extend the random feature eigenlearning framework of Simon et al. [57] to vector-valued random feature regression, with the original framework corresponding to the special case $m = 1$. Under the assumption that $\boldsymbol{g}$ in (11) is coordinatewise square integrable, it admits a singular value decomposition $\boldsymbol{g}(\boldsymbol{w}, \boldsymbol{z}) = \sum_i \sigma_i \zeta_i(\boldsymbol{w})\boldsymbol{\phi}_i(\boldsymbol{z})$, where the singular values satisfy $\sigma_1 \geq \sigma_2 \geq \cdots \geq 0$, and the *eigenfunctions* $\{\zeta_i\}_i \subset L^2(\rho)$ and $\{\boldsymbol{\phi}_i\}_i \subset L^2(\xi;\mathbb{R}^m)$ each form an orthonormal sequence in their respective spaces; see Theorem C.1. We set $\lambda_i = \sigma_i^2$ for further use. With these notations, we adopt the following ansatz.

Assumption 5.1 (Multivariate Gaussian Universality Ansatz)**.** The expected population risk remains unchanged even if we replace $\{\boldsymbol{\phi}_i(\boldsymbol{z})\}$ with i.i.d. samples from $\mathcal{N}(\boldsymbol{0}, \frac{1}{m}\boldsymbol{I})$ when $\boldsymbol{z} \sim \xi$, and $\{\zeta_i\}$ with i.i.d. samples from $\mathcal{N}(0, 1)$ when $\boldsymbol{w} \sim \rho$.

Notice that the covariance matrix associated to $\boldsymbol{\phi}$ is $\frac{1}{m}\boldsymbol{I}$ instead of $\boldsymbol{I}$, as the orthonormality of $\mathbb{R}^m$-valued functions requires $\mathbb{E}\big[\|\boldsymbol{\phi}_i(\boldsymbol{z})\|^2\big] = 1$. As stated in [57], the Gaussian universality ansatz, which is Assumption 5.1 when $m = 1$, is motivated by various universality results in random matrix theory: when computing certain scalar quantities derived from large random matrices, one can often replace the matrix entries with Gaussian random variables whose first two moments match those of the original entries. Assumption 5.1 is simply a translation of it into the multivariate setting.

5.2 Benign overfitting in generative models

In this section, we study when the benign overfitting phenomenon can occur in generative models. Let $\boldsymbol{\Sigma} \coloneqq \mathbb{E}_{\boldsymbol{z}\sim\xi}[\boldsymbol{\psi}(\boldsymbol{z})\boldsymbol{\psi}(\boldsymbol{z})^\top]$ denote the *noncentered covariance matrix*. Then, assuming that the

model is *well-specified* so that there exists $\boldsymbol{\beta}^*$ such that $G^* = \boldsymbol{\psi}(\cdot)^\top \boldsymbol{\beta}^*$, notice that the population risk can be equivalently written as

$$\mathcal{R}(\boldsymbol{\beta}) = \mathbb{E}_{\boldsymbol{z}\sim\xi}[\|\boldsymbol{\psi}(\boldsymbol{z})^\top \boldsymbol{\beta} - \boldsymbol{\psi}(\boldsymbol{z})^\top \boldsymbol{\beta}^*\|^2] = \|\boldsymbol{\beta} - \boldsymbol{\beta}^*\|_{\boldsymbol{\Sigma}}^2.$$

Similar to the standard bias–variance decomposition of the population risk, we can adapt Lemma S.1 in [8] to our setting. Let $\mathfrak{e} \in \mathbb{R}^{mn}$ denote the concatenation of the noises $\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_n$.

Lemma 5.2. *The population risk $\mathcal{R}(\hat{\boldsymbol{\beta}}(\delta))$ satisfies*

$$\mathcal{R}(\hat{\boldsymbol{\beta}}(\delta)) \leq 2\boldsymbol{\beta}^{*\top} \boldsymbol{B}_\delta \boldsymbol{\beta}^* + 2\mathfrak{e}^\top \boldsymbol{C}_\delta \mathfrak{e}$$

where we define $\boldsymbol{B}_\delta \coloneqq \boldsymbol{M}_\delta \boldsymbol{\Sigma} \boldsymbol{M}_\delta$ and $\boldsymbol{C}_\delta \coloneqq (\boldsymbol{\Psi}\boldsymbol{\Psi}^\top + \delta \boldsymbol{I})^{-1} \boldsymbol{\Psi}\boldsymbol{\Sigma}\boldsymbol{\Psi}^\top (\boldsymbol{\Psi}\boldsymbol{\Psi}^\top + \delta \boldsymbol{I})^{-1}$ with $\boldsymbol{M}_\delta \coloneqq \boldsymbol{I} - \boldsymbol{\Psi}^\top (\boldsymbol{\Psi}\boldsymbol{\Psi}^\top + \delta \boldsymbol{I})^{-1} \boldsymbol{\Psi}$.

The bias term $\boldsymbol{\beta}^\top \boldsymbol{B}_\delta \boldsymbol{\beta}$ can be bounded analogously to prior work. The variance term $\mathfrak{e}^\top \boldsymbol{C}_\delta \mathfrak{e}$, while more subtle to the dependency between the noises, can be bounded using the techniques introduced recently by Lee and Lee [38]; see Section E.2 for further details. Then, based on Bartlett et al. [8] and Tsigler and Bartlett [61], when $\delta > 0$, we establish that the expected population risk converges to zero asymptotically. To the best of our knowledge, this constitutes the first observation of benign overfitting in generative models.

Theorem 5.3. *Under Assumptions 4.1 and 5.1, if $\boldsymbol{\Sigma}$ has benign eigenvalues, then for any $\delta > 0$, the expected risk satisfies $\mathbb{E}[\mathcal{R}(\hat{\boldsymbol{\beta}}(\delta))] \to 0$ as $n \to \infty$.*

For instance, if the ith eigenvalue of $\boldsymbol{\Sigma}$ decays as $i^{-1} \ln^{-\alpha}(i+1)$ for some $\alpha > 1$, then the eigenvalues of $\boldsymbol{\Sigma}$ satisfy the benign eigenvalues condition. For further details, see Bartlett et al. [8].

5.3 The "more is better" principle in generative models

We now study how the "more is better" principle appears in generative models, where increasing the model size improves generalization. We particularly focus on the eigenframework analysis by Simon et al. [57], which established the principle in regression tasks. With the connection between regression and generative modeling we established in Section 4, this suggests that a similar phenomenon may hold in generative modeling, despite the common belief that increasing model size without proper regularization is known to impair generalization.

Let $G^*(z) = \sum_i v_i \boldsymbol{\phi}_i(z)$ be the expansion of the target function with respect to $\{\boldsymbol{\phi}_i\}_i$. In Section C, we show that the approximation formula for the expected population risk $\mathbb{E}_{\mathcal{D}}\big[\mathcal{R}(\hat{\boldsymbol{\beta}}(\delta))\big]$ proposed by Simon et al. [57] is largely generalizable, especially to when the target data is vector-valued and the noises $\boldsymbol{\epsilon}_i$ are from the regression model (10). More precisely, we show that the expected population risk $\mathbb{E}_{\mathcal{D}}\big[\mathcal{R}(\hat{\boldsymbol{\beta}})\big]$ of the regressor $\hat{G}$ obtained by performing random feature regression on our regression model (10) can be well approximated by the quantity

$$\mathcal{R}_{\text{te}} \coloneqq -\sigma^2 + \frac{1}{1 - \frac{q(p-2s)+s^2}{nm(p-q)}} \left(\sum_i \left(\frac{\gamma}{\lambda_i + \gamma} - \frac{\kappa \lambda_i}{(\lambda_i + \gamma)^2} \frac{p}{p-q} \right) v_i^2 + \sigma^2 \right) \tag{15}$$

where for $s = \sum_i \frac{\lambda_i}{\lambda_i + \gamma}$ and $q = \sum_i \big(\frac{\lambda_i}{\lambda_i + \gamma}\big)^2$ the scalars κ and γ are the unique nonnegative numbers such that $nm = s + \frac{m\delta}{p\kappa}$ and $p = s + \frac{p\kappa}{\gamma}$, and $\sigma^2 = \mathbb{E}\big[\|\boldsymbol{\epsilon}_1\|^2\big]$.

Once we have such an estimator, as in [57], we can show that increasing the number of features reduces the (approximated) expected population risk, so long as we are free to choose the ridge parameter.

Theorem 5.4. *Let $\mathcal{R}_{\text{te}}(n, p, \delta)$ denote the value of $\mathcal{R}_{\text{te}}$ with dataset size n, number of features p, and ridge parameter δ. Suppose that $p \leq p'$. Then, under Assumptions 4.1 and 5.1, it holds that*

$$\min_\delta \mathcal{R}_{\text{te}}(n, p', \delta) \leq \min_\delta \mathcal{R}_{\text{te}}(n, p, \delta).$$

Moreover, if we further assume that $p > nm$, then denoting $\mathcal{R}_{\text{te},0} = \lim_{\delta \to 0+} \mathcal{R}_{\text{te}}$, it holds that

$$\mathcal{R}_{\text{te},0}(n, p') \leq \mathcal{R}_{\text{te},0}(n, p).$$

6 Conclusion

We established that overparametrized generative models can generalize despite memorizing training data, contrary to the conventional belief that memorization undermines generalization performance. We demonstrated this phenomenon in flow matching with presampled latent variables empirically: trained models interpolate the training data while exhibiting generalization and a low reproduction rate across synthetic data and standard image benchmarks.

On the theoretical side, we demonstrated this through the lens of optimal transport and benign overfitting theories. By reformulating generative modeling into a regression task, quantitative analyses on the generalization performance of generative models became possible, leading to our theoretical demonstrations that benign overfitting can also occur in generative models. Specifically, we showed that increasing the number of features improves performance. Our work hence framed a new approach for studying the interplay between overparametrization and generalization in generative models.

Acknowledgments and Disclosure of Funding

This work was supported in part by the National Research Foundation of Korea (NRF) grants funded by the Korea government (MSIT) (Nos. RS-2019-NR040050 and RS-2025-00515159), and by the KAIST Grand Challenge 30 Project (KC30) funded by MSIT and KAIST (No. N11260046). JC was also supported by the NRF grant funded by the Korea government (MSIT) (No. RS-2024-00408003) and the KAIST Jang Young Sil Fellow Program. KK and DK were also supported by the NRF grant funded by the Korea government (MSIT) (No. RS-2026-25613008).

References

[1] Madhu S. Advani, Andrew M. Saxe, and Haim Sompolinsky. High-dimensional dynamics of generalization error in neural networks. *Neural Networks*, 132:428–446, 2020.

[2] Michael S. Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *International Conference on Learning Representations*, 2023.

[3] Namrata Anand and Possu Huang. Generative modeling for protein structures. *Advances in Neural Information Processing Systems*, 31, 2018.

[4] Greg W. Anderson, Alice Guionnet, and Ofer Zeitouni. *An Introduction to Random Matrices*. Cambridge University Press, 2010.

[5] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[6] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and equilibrium in generative adversarial nets (GANs). In *International conference on machine learning*, pages 224–232. PMLR, 2017.

[7] Ricardo Baptista, Agnimitra Dasgupta, Nikola B. Kovachki, Assad Oberai, and Andrew M. Stuart. Memorization and regularization in generative diffusion models. *arXiv preprint arXiv:2501.15785*, 2025.

[8] Peter L. Bartlett, Philip M. Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020.

[9] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

[10] Joe Benton, George Deligiannidis, and Arnaud Doucet. Error bounds for flow matching methods. *Transactions on Machine Learning Research*, 2024.

[11] Quentin Bertrand, Anne Gagneux, Mathurin Massias, and Rémi Emonet. On the closed-form of flow matching: Generalization does not arise from target stochasticity. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=kVz9uvqUna.

[12] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

[13] Tony Bonnaire, Raphaël Urfin, Giulio Biroli, and Marc Mézard. Why diffusion models don't memorize: The role of implicit dynamical regularization in training. *arXiv preprint arXiv:2505.17638*, 2025.

[14] Léon Bottou, Corinna Cortes, John S. Denker, Harris Drucker, Isabelle Guyon, Larry D. Jackel, Yann LeCun, Urs A. Müller, Edward Säckinger, Patrice Simard, and Vladimir Vapnik. Comparison of classifier methods: a case study in handwritten digit recognition. In *Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. 3-Conference C: Signal Processing (Cat. No. 94CH3440-5)*, volume 2, pages 77–82. IEEE, 1994.

[15] Yann Brenier. Décomposition polaire et réarrangement monotone des champs de vecteurs. *Comptes Rendus de l'Académie des Sciences - Série I - Mathématique*, 305:805–808, 1987.

[16] Claudio Carmeli, Ernesto De Vito, Alessandro Toigo, and Veronica Umanità. Vector valued reproducing kernel Hilbert spaces and universality. *Analysis and Applications*, 8(1):19–61, 2010.

[17] Ting Chen. On the importance of noise scheduling for diffusion models. *arXiv preprint arXiv:2301.10972*, 2023.

[18] Daniel K. Crane. *The singular value expansion for compact and non-compact operators*. PhD thesis, Michigan Technological University, 2020.

[19] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in Neural Information Processing Systems*, 2013.

[20] Aaron Defazio, Xingyu Yang, Harsh Mehta, Konstantin Mishchenko, Ahmed Khaled, and Ashok Cutkosky. The road less scheduled. *Advances in Neural Information Processing Systems*, 37, 2024.

[21] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning*, 2024.

[22] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 2014.

[23] Xiangming Gu, Chao Du, Tianyu Pang, Chongxuan Li, Min Lin, and Ye Wang. On memorization in diffusion models. *Transactions on Machine Learning Research*, 2025.

[24] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[25] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, second edition, 2009.

[26] Reinhard Heckel and Paul Hand. Deep decoder: Concise image representations from untrained non-convolutional networks. In *International Conference on Learning Representations*, 2019.

[27] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[28] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022.

[29] Guillaume Huguet, James Vuckovic, Kilian Fatras, Eric Thibodeau-Laufer, Pablo Lemos, Riashat Islam, Cheng-Hao Liu, Jarrid Rector-Brooks, Tara Akhound-Sadegh, Michael M. Bronstein, Alexander Tong, and Joey Bose. Sequence-augmented SE(3)-flow matching for conditional protein generation. *Advances in Neural Information Processing Systems*, 2024.

[30] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

[31] Vladimir Koltchinskii and Karim Lounici. Concentration inequalities and moment bounds for sample covariance operators. *Bernoulli*, pages 110–133, 2017.

[32] Alexander Korotin, Vage Egiazarian, Arip Asadulaev, Alexander Safin, and Evgeny Burnaev. Wasserstein-2 generative networks. In *International Conference on Learning Representations*, 2021.

[33] Daniel Kressner, Tingting Ni, and André Uschmajew. On the approximation of vector-valued functions by volume sampling. *Journal of Complexity*, 86, 2025.

[34] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 2012.

[35] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.

[36] Dohyun Kwon, Ying Fan, and Kangwook Lee. Score-based generative modeling secretly minimizes the Wasserstein distance. *Advances in Neural Information Processing Systems*, 35: 20205–20217, 2022.

[37] Peter D. Lax. *Functional Analysis*. John Wiley and Sons, Inc., 2002.

[38] Sungyoon Lee and Sokbae Lee. Prediction risk and estimation risk of the ridgeless least squares estimator under general assumptions on regression errors. In *The Thirteenth International Conference on Learning Representations*, 2025.

[39] Yaron Lipman, Ricky T.Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

[40] Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky T.Q. Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024.

[41] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023.

[42] Lorenzo Luzi, Yehuda Dar, and Richard Baraniuk. Double descent and other interpolation phenomena in GANs. *arXiv preprint arXiv:2106.04003*, 2021.

[43] Ashok Makkuva, Amirhossein Taghvaei, Sewoong Oh, and Jason Lee. Optimal transport mapping via input convex neural networks. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[44] Neil Mallinar, James Simon, Amirhesam Abedsoltan, Parthe Pandit, Misha Belkin, and Preetum Nakkiran. Benign, tempered, or catastrophic: A taxonomy of overfitting. *Advances in Neural Information Processing Systems*, 35:1182–1195, 2022.

[45] Charles A. Micchelli and Massimiliano Pontil. On learning vector-valued functions. *Neural Computation*, 17(1):177–204, 2005.

[46] Vaishnavh Nagarajan, Colin Raffel, and Ian J. Goodfellow. Theoretical insights into memorization in GANs. In *Integration of Deep Learning Theories Workshop, NeurIPS*, 2018.

[47] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003, 2021.

[48] Gabriel Peyré and Marco Cuturi. *Computational Optimal Transport*. Now Publishers, 2019.

[49] Jakiw Pidstrigach. Score-based generative models detect manifolds. *Advances in Neural Information Processing Systems*, 35:35852–35865, 2022.

[50] Aldo Pratelli. On the equality between Monge's infimum and Kantorovich's minimum in optimal mass transportation. *Annales de l'Institut Henri Poincare (B) Probability and Statistics*, 43(1):1–13, 2007.

[51] Anthony Ralston. Runge–Kutta methods with minimum error bounds. *Mathematics of Computation*, 16(80):431–437, 1962.

[52] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[53] Litu Rout, Alexander Korotin, and Evgeny Burnaev. Generative modeling with optimal transport maps. In *International Conference on Learning Representations*, 2022.

[54] Filippo Santambrogio. *Optimal transport for applied mathematicians*. Birkhäuser, 2015.

[55] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[56] James B. Simon, Madeline Dickens, Dhruva Karkada, and Michael R. DeWeese. The eigenlearning framework: A conservation law perspective on kernel ridge regression and wide neural networks. *Transactions on Machine Learning Research*, 2023.

[57] James B. Simon, Dhruva Karkada, Nikhil Ghosh, and Mikhail Belkin. More is better in modern machine learning: when infinite overparameterization is optimal and overfitting is obligatory. In *International Conference on Learning Representations*, 2024.

[58] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265, 2015.

[59] Ruoyu Sun, Tiantian Fang, and Alexander Schwing. Towards a better global loss landscape of GANs. *Advances in Neural Information Processing Systems*, 33:10186–10198, 2020.

[60] Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024.

[61] Alexander Tsigler and Peter L. Bartlett. Benign overfitting in ridge regression. *Journal of Machine Learning Research*, 24(123):1–76, 2023.

[62] Vladimir Vapnik. *Statistical learning theory*. John Wiley & Sons, 1998.

[63] Roman Vershynin. *High-Dimensional Probability*. Cambridge University Press, second edition, 2026.

[64] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[65] TaeHo Yoon, Joo Young Choi, Sehyun Kwon, and Ernest K. Ryu. Diffusion probabilistic models generalize when they fail to memorize. In *ICML 2023 workshop on structured probabilistic inference & generative modeling*, 2023.

[66] Kôsaku Yosida. *Functional analysis*. Springer, 6th edition, 1995.

[67] Claudio Zeni, Robert Pinsler, Daniel Zügner, Andrew Fowler, Matthew Horton, Xiang Fu, Zilong Wang, Aliaksandra Shysheya, Jonathan Crabbé, Shoko Ueda, Roberto Sordillo, Lixin Sun, Jake Smith, Bichlien Nguyen, Hannes Schulz, Sarah Lewis, Chin-Wei Huang, Ziheng Lu, Yichi Zhou, Han Yang, Hongxia Hao, Jielan Li, Chunlei Yang, Wenjie Li, Ryota Tomioka, and Tian Xie. A generative model for inorganic materials design. *Nature*, 639:624–632, 2025.

[68] Huijie Zhang, Jinfan Zhou, Yifu Lu, Minzhe Guo, Peng Wang, Liyue Shen, and Qing Qu. The emergence of reproducibility and consistency in diffusion models. In *International Conference on Machine Learning*, 2024.

[69] Pengchuan Zhang, Qiang Liu, Dengyong Zhou, Tao Xu, and Xiaodong He. On the discrimination-generalization tradeoff in GANs. In *International Conference on Learning Representations*, 2018.

Contents

1 Introduction **1**

2 Background **3**

2.1 Regression: Population and empirical risks 3

2.2 Generative modeling: Population and empirical risks 3

3 Empirical Evidence of Benign Overfitting in Flow Matching with Presampled Latent Variables **4**

3.1 Flow Matching with Presampled Latent Variables 4

3.2 Measuring Benign Overfitting: Interpolation Error and Reproduction Rate 4

3.3 Experiments on low-dimensional data 5

3.4 Experiments on image data 5

4 Recasting Generative Modeling with Presampled Latent Variables as Regression via Optimal Transport **6**

4.1 The first step toward bridging generative modeling and regression 6

4.2 Optimal transport map based regression model for generative modeling 7

5 Benign Overfitting Theory in Generative Modeling via Optimal Transport **7**

5.1 Regression with feature maps and the Gaussian universality ansatz 8

5.2 Benign overfitting in generative models 8

5.3 The "more is better" principle in generative models 9

6 Conclusion **10**

A Further Related Work **16**

A.1 From bias-variance trade-off to benign overfitting 16

A.2 Gaussian universality ansatz for scalar-valued regression 16

B Experiments on Flow Matching Models **17**

B.1 Flow matching experimental details 17

B.2 Gaussian data generation tasks with flow matching models 18

B.3 Image generation tasks with flow matching models 19

B.4 More is better principle in flow matching models 21

C Vector-Valued Kernel Regression and the Eigenlearning Frameworks **21**

C.1 Vector-valued linear models 21

C.2 Extending the eigenlearning framework 22

C.3 Singular value decomposition of vector-valued functions 23

C.4 Extending the random feature eigenframework 25

C.5 Neural-network-like random feature models 28

D Missing Details for Section 4 **30**
D.1 Consequences of Assumption 4.1 30
D.2 Annotations to the construction of the regression model (10) 30
D.3 Statistics of the noises in the regression model (10) 31
D.4 Learning optimal transport map from finite training data and presampled latent vectors 33

E Missing details for Section 5.2 **33**
E.1 Proof of Lemma 5.2 33
E.2 Upper bound of the variance term in Lemma 5.2 34
E.3 Proof of Theorem 5.3 35
E.3.1 Upper bound of bias term 35
E.3.2 Upper bound of the variance term 36
E.3.3 Benign condition for ridge regression 39

F Missing details for Section 5.3 **41**
F.1 Population risk and the Wasserstein distance 41
F.2 Incorporating the noise component into the eigenlearning framework 42
F.3 Proof of Theorem 5.4 45
F.4 More features is better also for neural-network-like random feature models 45

G Experimental Results on Random Feature Models **46**
G.1 Random feature models on Gaussian distributions 46
G.2 Neural-network-like model experiments 47

A Further Related Work

A.1 From bias-variance trade-off to benign overfitting

It is well known that the expected population risk of the learned model $\hat{f}$ can be decomposed as

$$\mathbb{E}_{\mathcal{D}}[\mathcal{R}(\hat{f})] = \underbrace{\|\mathbb{E}_{\mathcal{D}}[\hat{f}] - f\|_{L^2}^2}_{\text{(bias)}^2} + \underbrace{\mathbb{E}_{\mathcal{D}}[\|\hat{f} - \mathbb{E}_{\mathcal{D}}[\hat{f}]\|_{L^2}^2]}_{\text{variance}}.$$

The squared *bias* term quantifies the deviation between the expected learned predictor $\mathbb{E}_{\mathcal{D}}[\hat{f}]$ and the true target f. The *variance* term measures the variability of $\hat{f}$ across different training datasets. Classical statistical learning theory suggests a trade-off between bias and variance; see, *e.g.*, [25]. That is, models with high bias tend to underfit the data, failing to capture underlying patterns, whereas models with high variance tend to overfit, undesirably capturing noise in the training data. This trade-off reflects a fundamental tension between memorization and generalization.

However, recent empirical studies have shown that overparametrized models, despite being trained to perfectly memorize the training data, can still exhibit strong generalization [1, 9, 47], suggesting that it is possible to reduce bias and variance simultaneously. This widely observed benign overfitting phenomenon has been further supported by theoretical analyses [8, 57, 61].

A.2 Gaussian universality ansatz for scalar-valued regression

The original random feature eigenlearning framework of Simon et al. [57] considers the regression setting with scalar-valued targets, which corresponds to the case $m = 1$ in our notation. Assume that g is square integrable, so that it admits a singular value decomposition $g(\boldsymbol{w}, z) = \sum_i \sigma_i \zeta_i(\boldsymbol{w})\phi_i(z)$,

where $\{\zeta_i\}_i \subset L^2(\rho)$ and $\{\phi_i\}_i \subset L^2(\xi)$ are each orthonormal sequences in their respective spaces. The Gaussian universality ansatz can be stated as follows.

Assumption A.1 (Simon et al. [57], Gaussian Universality Ansatz)**.** The expected population risk remains unchanged even if we replace $\{\phi_i(z)\}$ with i.i.d. samples from $\mathcal{N}(0,1)$ when $z \sim \xi$, and $\{\zeta_i(\boldsymbol{w})\}$ with i.i.d. samples from $\mathcal{N}(0,1)$ when $\boldsymbol{w} \sim \rho$.

Notice that the above can indeed be recovered by setting $m = 1$ in the multivariate Gaussian universality ansatz (Assumption 5.1).

B Experiments on Flow Matching Models

Numerical computations were performed using the `Python` packages `NumPy` [24] and `SciPy` [64]. For flow matching experiments, we additionally used `PyTorch`. All the experiments were run on a node equipped with three NVIDIA GeForce RTX 4090 GPUs, each with 24GB of memory. The experiments that took the longest were CIFAR-10 training runs, each approximately taking three days.

B.1 Flow matching experimental details

As a proof of concept for our approach, which suggests that memorization can coexist with good generalization, we present experimental results using a "real-world" generative model, *flow matching*, demonstrating that the method remains effective without evident overfitting concerns.

Flow matching [2, 39, 41] is a generative modeling framework where we aim to find a (time-dependent) vector field whose flow transports the latent distribution ξ to the target distribution ν. More precisely, following the common notations within the flow matching framework, let us write $p_0 = \xi$, and assume that $d = m$ so that ξ and ν share the same ambient space. Our goal is to find a vector field $\boldsymbol{u}_t = \boldsymbol{u}(\,\cdot\,, t)$ so that its flow $\boldsymbol{\psi}_t = \boldsymbol{\psi}(\,\cdot\,, t)$ generates a *probability path* $p_t = (\boldsymbol{\psi}_t)_\sharp p_0$, a family of probability distributions indexed by $t \in [0,1]$, such that $p_1 = \nu$. Recall that $\boldsymbol{\psi}_t$ is a flow of the vector field $\boldsymbol{u}_t$ if $\boldsymbol{\psi}_0(\boldsymbol{x}) = \boldsymbol{x}$ and

$$\frac{\partial}{\partial t}\boldsymbol{\psi}_t(\boldsymbol{x}) = \boldsymbol{u}_t(\boldsymbol{\psi}_t(\boldsymbol{x})).$$

There is a plethora of approaches that lead to various constructions of such a vector field; for further details, we refer the interested reader to the survey [40]. Here we focus on a specific instance among those approaches,[2] which exploits the fact that, under any fixed coupling μ with marginals ξ and ν, the vector field $\boldsymbol{u}_t = \boldsymbol{u}(\,\cdot\,, t)$ obtained as

$$\boldsymbol{u}_t = \arg\min_{\boldsymbol{v}} \ \mathbb{E}_{(\boldsymbol{z},\boldsymbol{x})\sim\mu,t}\big[\|\boldsymbol{v}((1-t)\boldsymbol{z} + t\boldsymbol{x}, t) - (\boldsymbol{x} - \boldsymbol{z})\|^2\big], \tag{16}$$

induces a flow, under suitable regularity assumptions, that generates the desired probability path [60]. In many cases, it is assumed that ξ and ν are independent, so that $(\boldsymbol{z}, \boldsymbol{x}) \sim \mu$ reduces to independently sampling $\boldsymbol{z} \sim \xi$ and $\boldsymbol{x} \sim \nu$.

For convenience, let us denote $\boldsymbol{y}_t = (1-t)\boldsymbol{z} + t\boldsymbol{x}$. Notice that the objective of the minimization problem in (16) can be interpreted as the population risk of a regression task for $\boldsymbol{v} : \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^m$, with (possibly noisy) labels $\boldsymbol{x} - \boldsymbol{z} \in \mathbb{R}^m$ given at the point $(\boldsymbol{y}_t, t) \in \mathbb{R}^m \times \mathbb{R}$. Hence, the central idea of our framework carries over to this setting. More specifically, given a dataset $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, we first fix a set of presampled latent vectors $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_l$, and compute a vector field $\boldsymbol{v}$ as if we were given the regression model

$$\boldsymbol{x}_j - \boldsymbol{z}_i = \boldsymbol{v}\big((1-t)\boldsymbol{z}_i + t\boldsymbol{x}_j, t\big)$$

for each $t \in [0,1]$. Even further, one may sample and fix temporal values $t_1, \ldots, t_T$, and perform regression as if provided with the dataset $\big\{\big((1-t_k)\boldsymbol{z}_i + t_k\boldsymbol{x}_j, \boldsymbol{x}_j - \boldsymbol{z}_i\big)\big\}_{i,j,k}$.

Based on our discussions in this work so far, it is natural to expect benign overfitting to arise, allowing the computed regressor $\hat{\boldsymbol{v}}$ to accurately approximate the target vector field $\boldsymbol{u}$, even under

[2] Nonetheless, extending our subsequent discussions to other constructions of the vector field $\boldsymbol{u}_t$ should be straightforward, since typical flow matching objectives have an L^2 loss form similar to (16).

overtraining. In this context, *benign* overfitting should be understood as the regressor $\hat{\boldsymbol{v}}$ satisfying $\mathbb{E}_{\boldsymbol{y},t}\big[\|\hat{\boldsymbol{v}}(\boldsymbol{y},t)-\boldsymbol{u}_t(\boldsymbol{y})\|^2\big] \le \epsilon^2$ for some sufficiently small $\epsilon > 0$. If such a regressor is obtained, denoting by $\hat{p}_1$ the probability distribution generated by the flow of $\hat{\boldsymbol{v}}$ at $t = 1$, then Theorem 1 of Benton et al. [10] tells us that the bound

$$\mathcal{W}_2(\hat{p}_1, p_1) \le \epsilon \exp\left(\int_0^1 L_t \,\mathrm{d}t\right)$$

holds under mild regularity assumptions on $\boldsymbol{u}$ and $\hat{\boldsymbol{v}}$, such as continuous differentiability, where L_t is the Lipschitz constant of $\boldsymbol{y} \mapsto \hat{\boldsymbol{v}}(\boldsymbol{y}, t)$. If, in addition, the Lipschitz constants L_t remain controlled, the bound above shows that benign overfitting in the form of a small ϵ yields a small Wasserstein distance between the target distribution and the generated distribution. Such control may arise implicitly from the training dynamics or from an objective that encourages $\hat{\boldsymbol{v}}$ to remain close to $\boldsymbol{u}$, or it may be enforced by explicitly constraining the Lipschitz norm of $\hat{\boldsymbol{v}}$ with an upper bound that is large enough to still allow accurate approximation of $\boldsymbol{u}$.

Upon these discussions, we trained a flow matching model with fixed presampled latent vectors and temporal values. We train a neural network $\boldsymbol{v}_{\boldsymbol{\theta}}$, parametrized by $\boldsymbol{\theta}$, by minimizing the objective (7), that is,

$$\frac{1}{n\ell T}\sum_{i=1}^{\ell}\sum_{j=1}^{n}\sum_{k=1}^{T}\big\|\big(\boldsymbol{x}_j-\boldsymbol{z}_i\big)-\boldsymbol{v}_{\boldsymbol{\theta}}\big((1-t_k)\boldsymbol{z}_i+t_k\boldsymbol{x}_j,t_k\big)\big\|^2.$$

Notice that we have not specified the distribution of t in (16). It is in fact not necessary to use a uniform distribution for t, and one is free to choose any distribution [40, Section 4.5]. A common alternative [21] is the *logit-normal* distribution, whose density is

$$\omega(t; P_{\text{mean}}, P_{\text{std}}^2) = \frac{1}{t(1-t)\sqrt{2\pi P_{\text{std}}^2}} \exp\left(-\frac{\big(\log(\frac{t}{1-t}) - P_{\text{mean}}\big)^2}{2P_{\text{std}}^2}\right).$$

In our experiments, following this alternative, we fix t_k to be the $\frac{k-1}{T}$-quantile of the logit-normal distribution parameterized by mean P_{mean} and standard deviation P_{std}. One can verify that $0 \le t_k \le 1$ indeed for all $k \in [T]$.

The latent distribution is set to the standard normal distribution in the same dimensional space as the target data, following common practice.

B.2 Gaussian data generation tasks with flow matching models

In this section, we consider an three-dimensional Gaussian data generation task. The target training data consists of $n = 256$ samples drawn i.i.d. from

$$\nu = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \qquad \boldsymbol{\mu} = \left(\frac{1}{3}, \frac{1}{6}, \frac{1}{3}\right),$$

where

$$\boldsymbol{\Sigma} = \boldsymbol{R}\,\mathrm{diag}(1.2, 0.55, 0.18)\boldsymbol{R}^\top,$$

and $\boldsymbol{R}$ is a fixed orthogonal matrix. We also presample and fix $l = 256$ latent variables from the $\xi = \mathcal{N}(0, \boldsymbol{I}_3)$. We train with the objective in (7), using $T = 16$ discretized temporal variables.

The vector field $\boldsymbol{v}_\theta$ is parameterized by an MLP with four hidden layers of width 128 and SiLU activations. We train with AdamW optimizer using learning rate 10^{-3}. We repeat the experiment over five random seeds for each AdamW weight-decay value

$$\lambda \in \{0, 10^{-3}, 10^{-2}, 10^{-1}\}.$$

For evaluation, we generate samples by integrating the learned ODE from newly sampled latent variables $z \sim \xi$. We measure generalization by the squared Wasserstein distance between generated samples $\{\hat{x}_i\}_{i=1}^N$ and fresh target samples $\{x_i^{\text{test}}\}_{i=1}^N$:

$$\widehat{W}_2^2 = \min_{\pi \in S_N} \frac{1}{N}\sum_{i=1}^{N} \|\hat{x}_i - x_{\pi(i)}^{\text{test}}\|_2^2.$$

Table 2: Training configurations of the flow matching models on the MNIST and CIFAR-10 datasets.

Dataset	MNIST	CIFAR-10
OPTIMIZER SPECIFICATIONS		
optimizer	AdamWScheduleFree	AdamWScheduleFree
learning rate	0.001	0.0005
weight decay	0.01	0.01
batch size	256	256
number of epochs	1000	2000
OBJECTIVE FUNCTION SPECIFICATIONS		
dataset size (n)	60000	50000×2
sampled temporal values (T)	100	128
temporal distribution	logit-normal	logit-normal
U-NET ARCHITECTURE		
exponential moving average	disabled	disabled
`model_channels`	32	128
`num_res_blocks`	1	4
`attention_resolutions`	None	(2,)
`dropout`	0.0	0.3
`channel_mult`	(1, 2, 2)	(2, 2, 2)
trainable parameter count	1,020,897	39,012,099
INFERENCE DETAILS		
time steps	uniform, 50	
ODE solver	Ralston's second order	

For computational efficiency, this distance is computed using $N = 512$ samples. Memorization rate is measured using the nearest-neighbor criterion described in the main text.

We also evaluate interpolation error as follows. Because the objective uses fixed, presampled latent variables, it supervises the vector field only along a finite set of interpolation paths connecting these latents to the training data. Thus, interpolation does not force each presampled latent $\boldsymbol{z}_i$ to generate a particular training sample $\boldsymbol{x}_j$. Indeed, near $t = 0$, paths sharing the same $\boldsymbol{z}_i$ can induce an averaging effect over target samples, so the learned trajectory need not exactly follow certain single straight path from $\boldsymbol{z}_i$ to $\boldsymbol{x}_j$. Instead, we test whether a point on each training path can be traced backward to a certain latent variable whose generated sample recovers the corresponding training sample.

For a fixed probe time $t_c \in (0, 1)$ and pairs $(\boldsymbol{z}_i, \boldsymbol{x}_j)$, let $\boldsymbol{y}_{ij} = (1 - t_c)\boldsymbol{z}_i + t_c\boldsymbol{x}_j$. We integrate the learned ODE backward from time t_c to time 0, starting from $\boldsymbol{y}_{ij}$, to obtain $\tilde{\boldsymbol{z}}_{ij}$. We then generate $\hat{\boldsymbol{x}}_{ij}$ by integrating forward from $\tilde{\boldsymbol{z}}_{ij}$ to time 1, and measure the mean squared error of $\|\hat{\boldsymbol{x}}_{ij} - \boldsymbol{x}_j\|_2$. A small interpolation error indicates that the learned flow interpolates each training data point $\boldsymbol{x}_j$ with their associated $\tilde{\boldsymbol{z}}_{ij}$.

The results in Figure 1 show that the fully discrete objective in (7) decreases to a positive plateau, while the interpolation error and squared Wasserstein distance decrease and the memorization rate remains low. These results provide a proof of concept that minimizing the fully discrete flow-matching objective can interpolate the training data without harmful memorization, and can instead be benign in a low-dimensional generative modeling task.

B.3 Image generation tasks with flow matching models

In this section, we describe the image generation experiments used to show benign overfitting in flow matching models. The target datasets we tested on are the MNIST dataset, consisting of $n = 60000$ grayscale images of size 28×28, and the CIFAR-10 dataset, consisting of 50000 color images of size 32×32. For the CIFAR-10 dataset, random horizontal flipping is applied, which effectively doubles the number of data points in the dataset. For each dataset, we presample and fix $\ell \in \{n, 4n\}$ latent vectors from the latent distribution. Thus, $\ell \in \{60{,}000, 240{,}000\}$ for MNIST and $\ell \in \{100{,}000, 400{,}000\}$ for CIFAR-10.

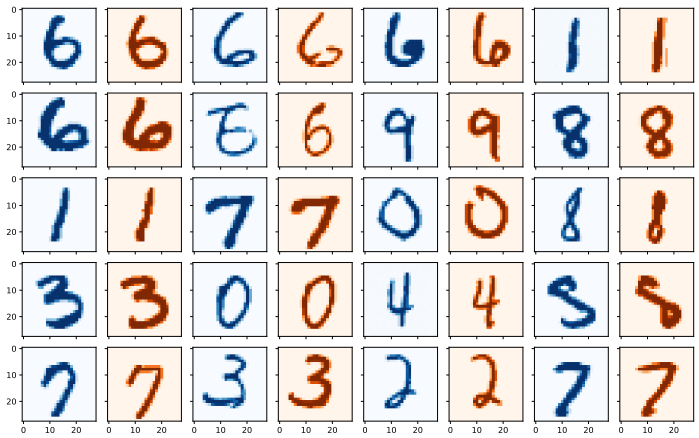


Figure 3: **Uncurated samples generated by a flow-matching model trained on the MNIST dataset with presampled latent vectors.** To illustrate the generalizability of the model, each generated image (shown in blue) is placed in the odd-numbered columns, and the image immediately to its right (shown in orange, in the even-numbered columns) is its nearest neighbor in the training dataset in Euclidean distance.

For the vector field v_θ, we modify the U-Net architecture from the official implementation[3][4] by Lipman et al. [40]. The network is trained using the AdamWScheduleFree optimizer [20]. Owing to the update rule of this optimizer, we do not apply an explicit exponential moving average.

In the inference stage, we solve the ODE using Ralston's second-order method [51] with 50 uniform time steps ($\Delta t = 1/50$). Further details on our experiments, including the training configurations, are provided in Table 2.

On the MNIST dataset, we ran the experiment five times and obtained an FID score of 2.71 ± 0.092. The FID score is computed against the test dataset of 10000 images, which is commonly referred to as FID-10k. An uncurated set of samples generated by our trained model is shown in Figure 3. In this figure, the generated samples appear in the odd-numbered columns (1, 3, 5, and 7). To illustrate the generalizability of the model, the nearest neighbor in the training dataset, with respect to the Euclidean distance, is displayed immediately to the right of each generated sample, in the corresponding even-numbered columns (2, 4, 6, and 8).

Although some generated images appear visually similar to their nearest neighbors in the training dataset, this behavior is somewhat expected on the MNIST dataset, which consists of simple, low-complexity digit symbols. With nearly 6000 examples per digit, a typical image drawn from the same underlying distribution is likely to have a close counterpart in the training set. This issue becomes less pronounced in the following results on the CIFAR-10 dataset, where it becomes more visually apparent that the models are not simply memorizing and regurgitating training samples but are instead genuinely generalizing.

On the CIFAR-10 dataset, we ran the experiment five times and obtained an FID-50k score of 3.45 ± 0.12. While this number might not be the state-of-the-art, we emphasize that our goal is not to optimize FID on this dataset, but rather to provide a proof of concept that our benign overfitting framework extends to realistic flow matching models. To this end, we deliberately use a standard U-Net architecture, without extensive hyperparameter tuning or architectural engineering. In this regime, the obtained FID is competitive with typical flow-based generative models on CIFAR-10 without heavy fine-tuning.

An uncurated set of samples generated by our trained model is shown in Figure 4. We employ the same plotting strategy as in Figure 3. That is, the generated samples appear in the odd-numbered columns, and their nearest neighbors in the training dataset are displayed immediately to the right of each generated sample. As is visually evident, the generated samples are not mere copies of training

[3] `https://github.com/facebookresearch/flow_matching`

[4] We note that our use of the logit-normal temporal distribution is inspired by, and consistent with, this implementation.

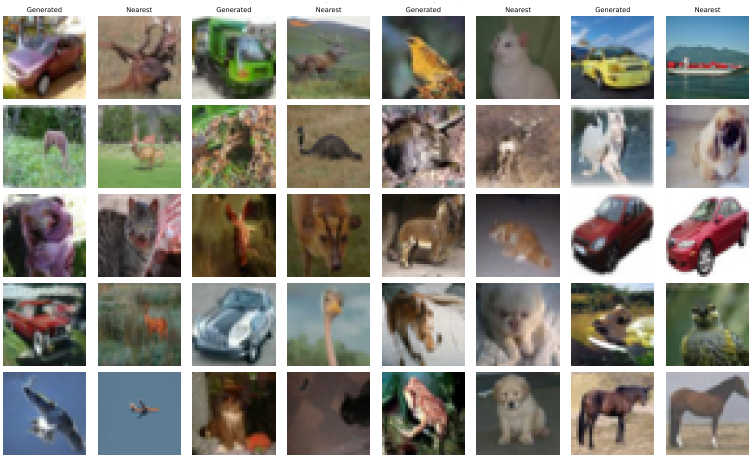


Figure 4: **Uncurated samples generated by a flow matching model trained on CIFAR-10 with presampled latent vectors.** To illustrate the generalizability of the model, each generated image is placed in the odd-numbered columns, and the image shown immediately to its right (in the even-numbered column) is the nearest neighbor in the training dataset in Euclidean distance.

examples. This visualization suggests that the model is not operating in a trivial regurgitation regime and is consistent with our claim that the learned flow, trained according to our framework, could generalize beyond pixel-level memorization in line with the benign overfitting perspective.

B.4 More is better principle in flow matching models

In this section, we present preliminary empirical results suggesting that the "more is better" principle also holds for flow matching models. In the high-dimensional setting $d = 784$, we set ξ to be the standard Gaussian distribution $\mathcal{N}(0, \boldsymbol{I})$ and ν to be an image distribution of the MNIST dataset. Unlike the setting in Section 3.3, the true target distribution is unknown. Therefore, we report the FID as a practical proxy for the population risk as the model width increases.

We train the flow matching models as detailed in Section B.3, with the exception of the `model_channels`, which we vary from 8 to 48. Figure 2 shows the results. We report the mean FID scores over five independent runs, with error bars denoting the standard deviation. As shown, the FID scores improve as the model width increases, which empirically supports the *more is better* principle.

C Vector-Valued Kernel Regression and the Eigenlearning Frameworks

C.1 Vector-valued linear models

As a preliminary step, let us review how the kernel method can be extended to when the targets, or labels, are multidimensional. For further details, see, *e.g.*, [45].

Suppose we are given a dataset $\{(\boldsymbol{z}_i, \boldsymbol{x}_i)\}_{i=1,\dots,n} \subset \mathbb{R}^d \times \mathbb{R}^m$. In finding a regressor, we consider linear models, which are functions of the form $\boldsymbol{f}(\boldsymbol{z}) = \boldsymbol{\psi}(\boldsymbol{z})^\top \boldsymbol{\beta}$ for some *matrix-valued* feature map $\boldsymbol{\psi} : \mathbb{R}^d \to \mathbb{R}^{p \times m}$, so that the model is linear on the p-dimensional parameter $\boldsymbol{\beta} \in \mathbb{R}^p$.

Such a feature map induces a *matrix-valued* kernel $\boldsymbol{k}(\boldsymbol{z}, \boldsymbol{z}') = \boldsymbol{\psi}(\boldsymbol{z})^\top \boldsymbol{\psi}(\boldsymbol{z}') \in \mathbb{R}^{m \times m}$. By mimicking the construction of the reproducing Hilbert kernel space (RKHS) associated with scalar-valued kernels, one can show the existence of the RKHS $\mathcal{H}$ associated with $\boldsymbol{k}$, in the sense that

(i) $\mathcal{H}$ is a Hilbert space consisting of functions $\mathbb{R}^d \to \mathbb{R}^m$,

(ii) there exist functions $\kappa_1, \dots, \kappa_m : \mathbb{R}^d \to \mathcal{H}$ such that the (i, j)-entry of an $m \times m$ matrix $\boldsymbol{k}(\boldsymbol{z}, \boldsymbol{z}')$ is equal to $\langle \kappa_i(\boldsymbol{z}), \kappa_j(\boldsymbol{z}') \rangle_{\mathcal{H}}$ for any $\boldsymbol{z}, \boldsymbol{z}' \in \mathbb{R}^d$, where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ is the inner product defined on $\mathcal{H}$, and

(iii) for any $\boldsymbol{h} \in \mathcal{H}$ and $\boldsymbol{z} \in \mathbb{R}^d$, denoting by h_j the jth component function of $\boldsymbol{h}$ and by $\boldsymbol{e}_j$ the jth standard basis vector of $\mathbb{R}^m$, it holds that

$$\langle \boldsymbol{h}, \boldsymbol{k}(\,\cdot\,, \boldsymbol{z})\boldsymbol{e}_j \rangle_{\mathcal{H}} = h_j(\boldsymbol{z}).$$

Notice that the empirical kernel matrix (or the *Gram matrix*) becomes a block matrix of the form

$$\hat{\boldsymbol{K}} = \begin{bmatrix} \boldsymbol{k}(\boldsymbol{z}_1, \boldsymbol{z}_1) & \cdots & \boldsymbol{k}(\boldsymbol{z}_1, \boldsymbol{z}_n) \\ \vdots & \ddots & \vdots \\ \boldsymbol{k}(\boldsymbol{z}_n, \boldsymbol{z}_1) & \cdots & \boldsymbol{k}(\boldsymbol{z}_n, \boldsymbol{z}_n) \end{bmatrix} \in \mathbb{R}^{nm \times nm}. \tag{17}$$

Now we consider the empirical risk minimization problem, applied to this setting. For a loss function $\mathcal{L} : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$, one can either consider an empirical risk minimization problem of finding an optimal parameter,

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \mathcal{L}(\boldsymbol{x}_i, \boldsymbol{\psi}(\boldsymbol{z}_i)^\top \boldsymbol{\beta}) + \frac{\delta}{2} \|\boldsymbol{\beta}\|^2, \tag{18}$$

or minimizing the empirical risk directly over the RKHS,

$$\min_{\boldsymbol{f} \in \mathcal{H}} \sum_{i=1}^{n} \mathcal{L}(\boldsymbol{x}_i, \boldsymbol{f}(\boldsymbol{z}_i)) + \frac{\delta}{2} \|\boldsymbol{f}\|_{\mathcal{H}}^2. \tag{19}$$

The representer theorem also holds for vector-valued outputs. In other words, there exist vectors $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n \in \mathbb{R}^m$ such that the empirical risk minimization problem (18) has an optimal solution $\hat{\boldsymbol{\beta}}$ which can be written as

$$\hat{\boldsymbol{\beta}} = \sum_{j=1}^{n} \boldsymbol{\psi}(\boldsymbol{z}_j)\boldsymbol{a}_j,$$

and moreover, for the same vectors $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$, the function

$$\hat{\boldsymbol{f}} = \sum_{j=1}^{n} \boldsymbol{k}(\,\cdot\,, \boldsymbol{z}_j)\boldsymbol{a}_j \tag{20}$$

becomes an optimal solution $\hat{\boldsymbol{f}}$ to (19). In particular, for our purpose, we set $\mathcal{L}$ to be the mean squared error $\mathcal{L}(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{x}'\|^2$. Then, (18) becomes an unconstrained convex quadratic programming, for which we readily have a closed-form solution

$$\mathfrak{a} = (\hat{\boldsymbol{K}} + \delta \boldsymbol{I})^{-1} \mathfrak{x}.$$

Here $\mathfrak{a}$ and $\mathfrak{x}$ are concatenations of $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$ and $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, respectively, into vectors in $\mathbb{R}^{nm}$.

C.2 Extending the eigenlearning framework

To simplify our narration, let us temporarily act as if the data points $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$ are sampled without replacement from a discrete set $\mathcal{X} \subset \mathbb{R}^d$ with $|\mathcal{X}| = M$ very large. This is a strategy also taken by Simon et al. [56], and the arguments supporting the validity of this choice therein still apply.

Under very mild assumptions, the kernel $\boldsymbol{k}$ admits a Mercer-type decomposition

$$\boldsymbol{k}(\boldsymbol{z}, \boldsymbol{z}') = \sum_i \lambda_i \boldsymbol{\phi}_i(\boldsymbol{z})\boldsymbol{\phi}_i(\boldsymbol{z}')^\top. \tag{21}$$

For example, only by assuming that $\boldsymbol{\psi}$ is continuous and $\boldsymbol{z} \mapsto \|\boldsymbol{k}(\boldsymbol{z}, \boldsymbol{z})\|$ is locally bounded, one can get a decomposition with each eigenvector $\boldsymbol{\phi}_i$ being continuous [16].

The decomposition (21) of the kernel $\boldsymbol{k}$ leads to a decomposition $\boldsymbol{K} = \boldsymbol{\Phi}^\top \boldsymbol{\Lambda} \boldsymbol{\Phi}$ of the empirical kernel matrix, where $\boldsymbol{\Phi}$ is the design matrix having a block matrix form

$$\boldsymbol{\Phi} = \begin{bmatrix} — \boldsymbol{\phi}_1(\boldsymbol{z}_1)^\top — & \cdots & — \boldsymbol{\phi}_1(\boldsymbol{z}_n)^\top — \\ \vdots & \ddots & \vdots \\ — \boldsymbol{\phi}_M(\boldsymbol{z}_1)^\top — & \cdots & — \boldsymbol{\phi}_M(\boldsymbol{z}_n)^\top — \end{bmatrix}$$

and $\boldsymbol{\Lambda} = \operatorname{diag}(\lambda_1, \ldots, \lambda_M)$.

Let $\boldsymbol{f} = \sum_i v_i \boldsymbol{\phi}_i$ and $\hat{\boldsymbol{f}} = \sum_i \hat{v}_i \boldsymbol{\phi}_i$ be the expansions of the target function $\boldsymbol{f}$ and the estimator $\hat{\boldsymbol{f}}$, respectively, with respect to the eigenbasis. With some algebra (which we omit here, as the derivations are mostly identical to what we detail in Section C.4), in the noiseless setting, one can verify that the identity

$$\hat{\boldsymbol{v}} = \underbrace{\boldsymbol{\Lambda}\boldsymbol{\Phi}(\boldsymbol{\Phi}^\top \boldsymbol{\Lambda}\boldsymbol{\Phi} + \delta \boldsymbol{I})^{-1}\boldsymbol{\Phi}^\top}_{=:\boldsymbol{T}} \boldsymbol{v} \tag{22}$$

still holds *mutatis mutandis*, with the *learning transfer matrix* $\boldsymbol{T}$.

C.3 Singular value decomposition of vector-valued functions

Now we wish to move on to random feature models, where the feature map $\boldsymbol{\psi}$ is possibly nondeterministic. However, to do so, there is one key result that has to be established beforehand. In the random feature eigenlearning framework [57] which considers the case where $m = 1$, the singular value decomposition (SVD) of $\boldsymbol{g}(\boldsymbol{w}, \boldsymbol{z})$, a scalar-valued function in that case, plays an important role. Thus, a key step in extending this framework is to show that SVD is also possible for vector-valued functions.

Theorem C.1 (Singular Value Decomposition of Vector-Valued Functions)**.** *Assume that $L^2(\rho)$ and $L^2(\xi; \mathbb{R}^m)$ are separable, and $\boldsymbol{g} \in L^2(\rho \times \xi; \mathbb{R}^m)$. Then, there exist positive numbers $\sigma_1 \geq \sigma_2 \geq \cdots$ and orthonormal sequences $\{\zeta_i\}_i \subset L^2(\rho)$ and $\{\boldsymbol{\phi}_i\}_i \subset L^2(\xi; \mathbb{R}^m)$ such that*

$$\boldsymbol{g}(\boldsymbol{w}, \boldsymbol{z}) = \sum_i \sigma_i \zeta_i(\boldsymbol{w}) \boldsymbol{\phi}_i(\boldsymbol{z}) \tag{23}$$

where the sum is either finite or converges in the L^2 norm.

Proof. Consider an integral operator $\mathcal{I} : L^2(\rho) \to L^2(\xi\,; \mathbb{R}^m)$ defined as

$$\mathcal{I}(f) = \int \boldsymbol{g}(\boldsymbol{w}, \cdot) f(\boldsymbol{w}) \, \mathrm{d}\rho(\boldsymbol{w}),$$

and denote the component operators of $\mathcal{I}$ by $\mathcal{I}(f) = (\mathcal{I}_1(f), \ldots, \mathcal{I}_m(f))$.

We first claim that $\mathcal{I}$ is a compact operator. As $\boldsymbol{g}$ is coordinatewise square-integrable, each of $\mathcal{I}_1, \ldots, \mathcal{I}_m$ is compact [37, Theorem 22.4]. Now suppose that $f_1, f_2, \ldots$ is a bounded sequence in $L^2(\rho)$. By passing into a subsequence m times, with at the jth pass ensuring that the image of the obtained subsequence under $\mathcal{I}_j$ converges, we can find a subsequence, say $\{f_{k_j}\}_j$, of $\{f_k\}_k$ such that all of $\{\mathcal{I}_1(f_{k_j})\}_j, \ldots, \{\mathcal{I}_m(f_{k_j})\}_j$ converge. This is equivalent to the convergence of $\{\mathcal{I}(f_{k_j})\}_j$, from which we can conclude that $\mathcal{I}$ itself is also indeed a compact operator.

The compactness of $\mathcal{I}$ implies that it admits a singular value decomposition [18, Theorem 1.6]; there exist positive numbers $\sigma_1 \geq \sigma_2 \geq \ldots$ and orthonormal sequences $\{\zeta_i\}_i \subset L^2(\rho)$ and $\{\boldsymbol{\phi}_i\}_i \subset L^2(\xi; \mathbb{R}^m)$ such that

$$\mathcal{I}(\,\cdot\,) = \sum_i \sigma_i \left\langle \zeta_i, \cdot \right\rangle_{L^2(\rho)} \boldsymbol{\phi}_i \tag{24}$$

where the sum converges in the operator norm.

Let $\{\eta_i\}_i$ an orthonormal sequence in $L^2(\rho)$ so that $\{\zeta_i\} \cup \{\eta_i\}$ is an orthonormal basis of $L^2(\rho)$. Similarly, let $\{\boldsymbol{\chi}_i\}_i \subset L^2(\xi; \mathbb{R}^m)$ be so that $\{\boldsymbol{\phi}_i\}_i \cup \{\boldsymbol{\chi}_i\}_i$ is an orthonormal basis of $L^2(\xi; \mathbb{R}^m)$. For the moment, let us assume that the right hand side of (24) is an infinite sum. For each $k = 1, 2, \ldots$, let $\boldsymbol{g}_k(\boldsymbol{w}, \boldsymbol{z}) = \sum_{i=1}^k \sigma_i \zeta_i(\boldsymbol{w}) \boldsymbol{\phi}_i(\boldsymbol{z})$, $\boldsymbol{r}_{k+1} = \boldsymbol{g} - \boldsymbol{g}_k$, and

$$\mathcal{J}_k(f) = \int \boldsymbol{r}_{k+1}(\boldsymbol{w}, \cdot) f(\boldsymbol{w}) \, \mathrm{d}\rho(\boldsymbol{w}).$$

Then by (24), we have

$$\begin{aligned}\mathcal{J}_k(\zeta_i) &= \int \boldsymbol{g}(\boldsymbol{w},\,\cdot\,)\zeta_i(\boldsymbol{w})\,\mathrm{d}\rho(\boldsymbol{w}) - \int \boldsymbol{g}_k(\boldsymbol{w},\,\cdot\,)\zeta_i(\boldsymbol{w})\,\mathrm{d}\rho(\boldsymbol{w}) \\ &= \mathcal{I}(\zeta_i) - \sum_{j=1}^{k} \int \sigma_j \zeta_i(\boldsymbol{w})\zeta_j(\boldsymbol{w})\boldsymbol{\phi}_j\,\mathrm{d}\rho(\boldsymbol{w}) \\ &= \sigma_i\boldsymbol{\phi}_i - \sigma_i\boldsymbol{\phi}_i\mathbb{1}_{\{i\le k\}} \\ &= \sigma_i\boldsymbol{\phi}_i\mathbb{1}_{\{i>k\}}.\end{aligned} \tag{25}$$

For any $i, j = 1, 2, \ldots$, using $\zeta_i\boldsymbol{\phi}_j$ to denote $(\boldsymbol{w}, \boldsymbol{z}) \mapsto \zeta_i(\boldsymbol{w})\boldsymbol{\phi}_j(\boldsymbol{z})$, it follows that

$$\begin{aligned}\langle \boldsymbol{r}_{k+1}, \zeta_i\boldsymbol{\phi}_j\rangle_{L^2(\rho\times\xi;\mathbb{R}^m)} &= \iint \boldsymbol{r}_{k+1}(\boldsymbol{w},\boldsymbol{z})^\top \left(\zeta_i(\boldsymbol{w})\boldsymbol{\phi}_j(\boldsymbol{z})\right)\,\mathrm{d}(\rho\times\xi)(\boldsymbol{w},\boldsymbol{z}) \\ &= \int \left(\int \boldsymbol{r}_{k+1}(\boldsymbol{w},\boldsymbol{z})\zeta_i(\boldsymbol{w})\,d\rho(\boldsymbol{w})\right)^\top \boldsymbol{\phi}_j(\boldsymbol{z})\,\mathrm{d}\xi(\boldsymbol{z}) \\ &= \int \mathcal{J}_k(\zeta_i)^\top\boldsymbol{\phi}_j\,\mathrm{d}\xi \\ &= \langle \mathcal{J}_k(\zeta_i), \boldsymbol{\phi}_j\rangle_{L^2(\xi;\mathbb{R}^m)} \\ &= \sigma_i\delta_{ij}\mathbb{1}_{\{i>k\}}.\end{aligned} \tag{26}$$

Let us use analogous notations for $\eta_i\boldsymbol{\phi}_j$, $\zeta_i\boldsymbol{\chi}_j$, and $\eta_i\boldsymbol{\chi}_j$. As $\langle\zeta_I, \eta_i\rangle_{L^2(\rho)} = 0$ for any pair (I, i), it is immediate from (24) that $\mathcal{I}(\eta_i) = \mathbf{0}$. Following the same steps as in (25) and (26), we also get $\mathcal{J}_k(\eta_i) = \mathbf{0}$ and hence

$$\langle \boldsymbol{r}_{k+1}, \eta_i\boldsymbol{\phi}_j\rangle_{L^2(\rho\times\xi;\mathbb{R}^m)} = 0. \tag{27}$$

Moreover, as $\langle\boldsymbol{\phi}_I, \boldsymbol{\chi}_i\rangle_{L^2(\xi;\mathbb{R}^m)} = 0$ for any pair (I, i), for any $\theta \in \{\zeta_i\}\cup\{\eta_i\}$, from (24) we get $\langle\mathcal{I}(\theta), \boldsymbol{\chi}_j\rangle_{L^2(\xi;\mathbb{R}^m)} = 0$, and by the construction of $\boldsymbol{g}_k$ we further have $\langle\mathcal{J}_k(\theta), \boldsymbol{\chi}_j\rangle_{L^2(\xi;\mathbb{R}^m)} = 0$. So it follows that

$$\langle \boldsymbol{r}_{k+1}, \theta\boldsymbol{\chi}_j\rangle_{L^2(\rho\times\xi;\mathbb{R}^m)} = \langle\mathcal{J}_k(\theta), \boldsymbol{\chi}_j\rangle_{L^2(\xi;\mathbb{R}^m)} = 0. \tag{28}$$

As $\{\zeta_i\boldsymbol{\phi}_j\}_{i,j}\cup\{\eta_i\boldsymbol{\phi}_j\}_{i,j}\cup\{\zeta_i\boldsymbol{\chi}_j\}_{i,j}\cup\{\eta_i\boldsymbol{\chi}_j\}_{i,j}$ is an orthonormal basis of $L^2(\rho\times\xi;\mathbb{R}^m)$ with (27) and (28), we conclude that

$$\begin{aligned}\|\boldsymbol{r}_{k+1}\|^2_{L^2(\rho\times\xi;\mathbb{R}^m)} &= \sum_{i,j}\left|\langle \boldsymbol{r}_{k+1}, \zeta_i\boldsymbol{\phi}_j\rangle_{L^2(\rho\times\xi;\mathbb{R}^m)}\right|^2 \\ &= \sum_{i,j}\sigma_i^2\delta_{ij}\mathbb{1}_{\{i>k\}} \\ &= \sum_{i>k}\sigma_i^2.\end{aligned}$$

In particular, by considering when $k = 0$, because $\boldsymbol{r}_1 = \boldsymbol{g}$ is square-integrable we get $\sum_i\sigma_i^2 < \infty$, and consequently, $\|\boldsymbol{g} - \boldsymbol{g}_k\|^2_{L^2(\rho\times\xi;\mathbb{R}^m)} = \|\boldsymbol{r}_{k+1}\|^2_{L^2(\rho\times\xi;\mathbb{R}^m)} \to 0$ as $k\to\infty$. That is,

$$\boldsymbol{g}(\boldsymbol{w},\boldsymbol{z}) = \sum_i \sigma_i\zeta_i(\boldsymbol{w})\boldsymbol{\phi}_i(\boldsymbol{z}), \tag{29}$$

where the sum converges in the L^2 norm. Now, if the right hand side of (24) were a finite sum, say i ranging from 1 to I, the same logic applies, but it suffices to consider k ranging only up to I, and we would have $\boldsymbol{g} - \boldsymbol{g}_I = \mathbf{0}$. Still, we do have (29) in this case also, hence the proof is complete. □

We remark that a similar type of function decomposition was recently considered by Kressner et al. [33, Theorem 2].

To avoid unnecessary complexities, it is desirable that $\{\zeta_i\}_i$ and $\{\boldsymbol{\phi}_i\}$ are orthonormal bases of their respective ambient spaces. To this end, from now on let us assume that the dimensions of the

orthogonal complements of the subspaces spanned by singular vectors corresponding to nonzero singular values, in each $L^2(\rho)$ and $L^2(\xi;\mathbb{R}^m)$, are equal. That is, we assume that

$$\dim\big(\operatorname{span}\{\zeta_i\}^\perp\big) = \dim\big(\operatorname{span}\{\boldsymbol{\phi}_i\}^\perp\big),$$

where we consider the equality as valid also when both dimensionalities are countably infinite. By doing so, even if we extend $\{\zeta_i\}_i$ and $\{\boldsymbol{\phi}_i\}_i$ into orthonormal bases, we still have the expansion of $\boldsymbol{g}$ as in (23) by allowing $\sigma_i = 0$ if necessary.

C.4 Extending the random feature eigenframework

Under the random feature eigenframework and the Gaussian universality ansatz (Assumption A.1), for the case when $m = 1$ and the noises $\epsilon_1, \dots, \epsilon_n$ are i.i.d. Gaussians with variance $\check{\sigma}^2$, a formula for approximating the expected population risk $\mathbb{E}_{\mathcal{D}}\big[\mathcal{R}(\hat{\boldsymbol{\beta}}(\delta))\big]$ has been established by Simon et al. [57]. We show that their result can be largely generalized, as mentioned in Section 5.3.

The SVD result now enables us to analyze random feature models with vector-valued outputs, which we introduced in Section 5. More precisely, we now consider when, for some function $\boldsymbol{g} : \mathbb{R}^h \times \mathbb{R}^d \to \mathbb{R}^m$ and a probability distribution ρ on $\mathbb{R}^h$, the feature map $\boldsymbol{\psi}$ is given by

$$\boldsymbol{\psi}(\boldsymbol{z}) = \begin{bmatrix} — \, \boldsymbol{g}(\boldsymbol{w}_1, \boldsymbol{z})^\top \, — \\ \vdots \\ — \, \boldsymbol{g}(\boldsymbol{w}_p, \boldsymbol{z})^\top \, — \end{bmatrix}$$

where $\boldsymbol{w}_1, \dots, \boldsymbol{w}_p$ are i.i.d. samples from ρ.

Due to the targets being multidimensional, we have to deal with vector spaces of $\mathbb{R}^m$-valued functions. In particular, we have to work with $L^2(\xi;\mathbb{R}^m)$, the vector space of $\mathbb{R}^m$-valued functions that are "square-integrable with respect to ξ". Strictly speaking, to define measurability and integrability of Banach-space-valued functions in general, one has to introduce the concept of *Bochner integrals*; see, *e.g.*, Sections V.4 and V.5 in [66]. Fortunately, when the codomain is a Euclidean space $\mathbb{R}^m$, the measurability (resp., integrability) of a function reduces to the measurability (resp., integrability) of each of the component functions. Under this notion of integrability, $L^2(\xi;\mathbb{R}^m)$ becomes a Hilbert space once we give the inner product as

$$\langle \boldsymbol{f}, \boldsymbol{h} \rangle_{L^2(\xi;\mathbb{R}^m)} \coloneqq \int \boldsymbol{f}(\boldsymbol{z})^\top \boldsymbol{h}(\boldsymbol{z}) \, \mathrm{d}\xi(\boldsymbol{z}).$$

To match the notations with Simon et al. [57], for random feature regression we consider a rescaled version of the empirical kernel. More precisely, equipped with the SVD of vector-valued functions, we consider

$$\begin{aligned} \hat{\boldsymbol{k}}(\boldsymbol{z}, \boldsymbol{z}') = \frac{1}{p} \boldsymbol{\psi}(\boldsymbol{z})^\top \boldsymbol{\psi}(\boldsymbol{z}') &= \frac{1}{p} \sum_{i=1}^{p} \boldsymbol{g}(\boldsymbol{w}_i, \boldsymbol{z}) \boldsymbol{g}(\boldsymbol{w}_i, \boldsymbol{z}')^\top \\ &= \frac{1}{p} \sum_{j,j'} \sum_{i=1}^{p} \sigma_j \sigma_{j'} \zeta_j(\boldsymbol{w}_i) \zeta_{j'}(\boldsymbol{w}_i) \boldsymbol{\phi}_j(\boldsymbol{z}) \boldsymbol{\phi}_{j'}(\boldsymbol{z}')^\top \end{aligned} \tag{30}$$

as the empirical kernel, and the corresponding deterministic kernel

$$\boldsymbol{k}(\boldsymbol{z}, \boldsymbol{z}') = \mathbb{E}_{\boldsymbol{w}}[\boldsymbol{g}(\boldsymbol{w}, \boldsymbol{z}) \boldsymbol{g}(\boldsymbol{w}, \boldsymbol{z}')^\top] = \sum_j \lambda_j \boldsymbol{\phi}_j(\boldsymbol{z}) \boldsymbol{\phi}_j(\boldsymbol{z}')^\top \tag{31}$$

where we set $\lambda_j = \sigma_j^2$. Then, thanks to the similarity of the equations (30) and (31) to their counterparts in the scalar-valued random feature eigenframework [57], the remaining steps are mostly straightforward, resembling what we did in Section C.2 to extend the scalar-valued framework into the vector-valued setting. Define $\boldsymbol{\Lambda} \coloneqq \operatorname{diag}(\lambda_1, \lambda_2, \dots)$ and

$$\boldsymbol{Z} \coloneqq \begin{bmatrix} \zeta_1(\boldsymbol{w}_1) & \cdots & \zeta_1(\boldsymbol{w}_p) \\ \zeta_2(\boldsymbol{w}_1) & \cdots & \zeta_2(\boldsymbol{w}_p) \\ \vdots & \vdots & \vdots \end{bmatrix}$$

so that

$$\tilde{\boldsymbol{\Lambda}} \coloneqq \frac{1}{p}\boldsymbol{\Lambda}^{1/2}\boldsymbol{Z}\boldsymbol{Z}^\top\boldsymbol{\Lambda}^{1/2} \tag{32}$$

is a matrix whose entries are $\tilde{\Lambda}_{jj'} = \frac{1}{p}\sum_{i=1}^{p}\sigma_j\sigma_{j'}\zeta_j(\boldsymbol{w}_i)\zeta_{j'}(\boldsymbol{w}_i)$. Then for a design matrix

$$\boldsymbol{\Phi} = \begin{bmatrix} — \boldsymbol{\phi}_1(\boldsymbol{z}_1)^\top — & \cdots & — \boldsymbol{\phi}_1(\boldsymbol{z}_n)^\top — \\ — \boldsymbol{\phi}_2(\boldsymbol{z}_1)^\top — & \cdots & — \boldsymbol{\phi}_2(\boldsymbol{z}_n)^\top — \\ \vdots & \vdots & \vdots \end{bmatrix},$$

it is clear from (30) that the empirical kernel matrix $\hat{\boldsymbol{K}} = \left[\hat{\boldsymbol{k}}(\boldsymbol{z}_i, \boldsymbol{z}_j)\right]_{i,j}$ can be written as

$$\hat{\boldsymbol{K}} = \boldsymbol{\Phi}^\top\tilde{\boldsymbol{\Lambda}}\boldsymbol{\Phi}. \tag{33}$$

We remark that, as we are using the rescaled empirical kernel, the Gram matrix should also be understood as being rescaled; *i.e.*, the Gram matrix in (17) actually corresponds to $p\hat{\boldsymbol{K}}$ where $\hat{\boldsymbol{K}}$ here is as in (33). In particular, (20) now becomes

$$\hat{\boldsymbol{f}} = p\sum_{j=1}^{n}\hat{\boldsymbol{k}}(\,\cdot\,, \boldsymbol{z}_j)\boldsymbol{a}_j,$$

where $\mathfrak{a} = (p\hat{\boldsymbol{K}} + \delta\boldsymbol{I})^{-1}\mathfrak{x}$.

So, let $\boldsymbol{f} = \sum_i v_i\boldsymbol{\phi}_i$ and $\hat{\boldsymbol{f}} = \sum_i \hat{v}_i\boldsymbol{\phi}_i$ be the expansions with respect to the eigenbasis of the target function $\boldsymbol{f}$ and the estimator $\hat{\boldsymbol{f}}$, respectively. Then for each $J = 1, 2, \ldots,$ noting that the kernel we actually use in the regression is the empirical kernel $\hat{\boldsymbol{k}}$, one can observe that

$$\begin{aligned}
\hat{v}_J = \left\langle \boldsymbol{\phi}_J, \hat{\boldsymbol{f}} \right\rangle_{L^2(\xi;\mathbb{R}^m)} &= \left\langle \boldsymbol{\phi}_J, p\sum_{s=1}^{n}\hat{\boldsymbol{k}}(\,\cdot\,, \boldsymbol{z}_s)\boldsymbol{a}_s \right\rangle_{L^2(\xi;\mathbb{R}^m)} \\
&= \sum_{s=1}^{n}\sum_{j'}\sum_{i=1}^{p}\sigma_J\sigma_{j'}\zeta_J(\boldsymbol{w}_i)\zeta_{j'}(\boldsymbol{w}_i)\boldsymbol{\phi}_{j'}(\boldsymbol{z}_s)^\top\boldsymbol{a}_s \\
&= p\sum_{s=1}^{n}\sum_{j'}\tilde{\boldsymbol{\Lambda}}_{Jj'}\boldsymbol{\phi}_{j'}(\boldsymbol{z}_s)^\top\boldsymbol{a}_s \\
&= p\left[\tilde{\boldsymbol{\Lambda}}\boldsymbol{\Phi}\mathfrak{a}\right]_J
\end{aligned}$$

where $[\,\cdot\,]_J$ denotes the Jth component of a vector. Therefore, when there is no noise in the targets (*i.e.,* labels), the identity

$$\begin{aligned}
\hat{\boldsymbol{v}} &= p\tilde{\boldsymbol{\Lambda}}\boldsymbol{\Phi}\mathfrak{a} \\
&= p\tilde{\boldsymbol{\Lambda}}\boldsymbol{\Phi}(p\hat{\boldsymbol{K}} + \delta\boldsymbol{I})^{-1}\mathfrak{x} \\
&= \underbrace{\tilde{\boldsymbol{\Lambda}}\boldsymbol{\Phi}\left(\boldsymbol{\Phi}^\top\tilde{\boldsymbol{\Lambda}}\boldsymbol{\Phi} + \frac{\delta}{p}\boldsymbol{I}\right)^{-1}\boldsymbol{\Phi}^\top}_{=:\tilde{\boldsymbol{T}}}\boldsymbol{v}
\end{aligned}$$

holds, with the *learning transfer matrix* $\tilde{\boldsymbol{T}}$.

For further analyses, it would be desirable to have an assumption similar to the Gaussian universality ansatz (Assumption A.1). The orthonormality equations

$$\mathbb{E}_{\boldsymbol{w}\sim\rho}[\zeta_i(\boldsymbol{w})\zeta_j(\boldsymbol{w})] = \delta_{ij} \tag{34a}$$

$$\mathbb{E}_{\boldsymbol{z}\sim\xi}[\boldsymbol{\phi}_i(\boldsymbol{z})^\top\boldsymbol{\phi}_j(\boldsymbol{z})] = \delta_{ij} \tag{34b}$$

suggest a natural multivariate rendition of the ansatz, as we introduced in Section 5.

Assumption 5.1 (*Multivariate Gaussian Universality Ansatz*)**.** The expected population risk remains unchanged even if we replace $\{\boldsymbol{\phi}_i(\boldsymbol{z})\}$ with i.i.d. samples from $\mathcal{N}(\boldsymbol{0}, \frac{1}{m}\boldsymbol{I})$ when $\boldsymbol{z} \sim \xi$, and $\{\zeta_i\}$ with i.i.d. samples from $\mathcal{N}(0, 1)$ when $\boldsymbol{w} \sim \rho$.

Under the multivariate Gaussian universality ansatz, $\boldsymbol{\Phi}$ is a matrix whose entries are i.i.d. samples from $\mathcal{N}(0, \frac{1}{m})$, or equivalently, $\frac{1}{\sqrt{m}}\mathcal{N}(0,1)$. Hence, rewriting the learning transfer matrix in terms of $\boldsymbol{\Omega} = \sqrt{m}\boldsymbol{\Phi}$ as

$$\tilde{\boldsymbol{T}} = \tilde{\boldsymbol{\Lambda}}\boldsymbol{\Omega}\left(\boldsymbol{\Omega}^\top\tilde{\boldsymbol{\Lambda}}\boldsymbol{\Omega} + \frac{m\delta}{p}\boldsymbol{I}\right)^{-1}\boldsymbol{\Omega}^\top \tag{35}$$

shows that $\tilde{\boldsymbol{T}}$ is statistically equivalent to the learning transfer matrix appearing in the scalar-valued random feature eigenframework, up to the number of samples n and the ridge parameter δ being replaced by nm and $\frac{m\delta}{p}$, respectively. That is, we can apply the results from the random feature eigenframework [57], with necessary modifications, as follows.

Observation C.2. *Assume that the multivariate Gaussian universality ansatz (Assumption 5.1) holds. For the given dataset $\{(\boldsymbol{z}_i, \boldsymbol{x}_i)\}_{i=1,\dots,n} \subset \mathbb{R}^d \times \mathbb{R}^m$, suppose that $\boldsymbol{x}_i = \boldsymbol{f}(\boldsymbol{z}_i) + \boldsymbol{\omega}_i$, where $\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_n$ are i.i.d. Gaussian noises from $\mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I})$, and we do a vector-valued random feature regression on this dataset. Let $s \coloneqq \sum_i \frac{\lambda_i}{\lambda_i+\gamma}$ and $q \coloneqq \sum_i \frac{\lambda_i^2}{(\lambda_i+\gamma)^2}$, where $\kappa, \gamma \geq 0$ are the unique nonnegative scalars such that*

$$nm = s + \frac{m\delta}{p\kappa} \qquad \text{and} \qquad p = s + \frac{p\kappa}{\gamma}.$$

Then, the test mean squared error from the vector-valued random feature regression can be approximated by as

$$\mathcal{E}_{\text{te}} \coloneqq \frac{1}{1 - \frac{q(p-2s)+s^2}{nm(p-q)}}\left(\sum_i \left(\frac{\gamma}{\lambda_i+\gamma} - \frac{\kappa\lambda_i}{(\lambda_i+\gamma)^2}\frac{p}{p-q}\right)v_i^2 + m\sigma^2\right). \tag{36}$$

Proof. One can derive (36) by plugging in the appropriate parameters, as discussed above, into the risk estimate computed by Simon et al. [57] using the one-dimensional eigenlearning framework. The only term that we have not yet discussed is the one corresponding to the label noise. Since each label noise $\boldsymbol{\omega}_i$ has coordinatewise variance σ^2 and each label is m-dimensional, it follows immediately that the total noise level is $m\sigma^2$. This explains the $m\sigma^2$ term in (36), and we are done. □

The renormalizing trick we used in (35) also applies to the learning transfer equation (22) in the eigenlearning framework for deterministic feature maps. Thus, applying the results from Simon et al. [56], a risk estimate analogous to (36) can be obtained, which we state here for future reference.

Observation C.3. *Assume that the multivariate Gaussian universality ansatz (Assumption 5.1) holds. For the given dataset $\{(\boldsymbol{z}_i, \boldsymbol{x}_i)\}_{i=1,\dots,n} \subset \mathbb{R}^d \times \mathbb{R}^m$, suppose that $\boldsymbol{x}_i = \boldsymbol{f}(\boldsymbol{z}_i) + \boldsymbol{\omega}_i$, where $\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_n$ are i.i.d. Gaussian noises from $\mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I})$, and we do a vector-valued kernel ridge regression of solving* (19) *on this dataset. Let $s \coloneqq \sum_i \frac{\lambda_i}{\lambda_i+\kappa}$ and $q \coloneqq \sum_i \frac{\lambda_i^2}{(\lambda_i+\kappa)^2}$, where $\kappa \geq 0$ is the unique nonnegative scalar such that*

$$nm = s + \frac{m\delta}{\kappa}.$$

Then, we can approximate the test error as

$$\mathcal{E}_{\text{te}} \coloneqq \frac{1}{1 - \frac{q}{nm}}\left(\sum_i \frac{\kappa^2 v_i^2}{(\lambda_i+\kappa)^2} + m\sigma^2\right). \tag{37}$$

Remark C.4. Notice that in Observations C.2 and C.3 we stated that $\mathcal{E}_{\text{te}}$ approximates the *test mean squared error*. Indeed, the quantity $\mathcal{E}_{\text{te}}$ estimates is $\|\hat{\boldsymbol{f}} - \boldsymbol{f}\|^2 + m\sigma^2$, or equivalently $\mathbb{E}\big[\|\boldsymbol{x} - \hat{\boldsymbol{f}}(\boldsymbol{z})\|^2\big]$, where $\boldsymbol{z} \sim \xi$ and $\boldsymbol{x} = f(\boldsymbol{z}) + \boldsymbol{\omega}$ with $\boldsymbol{\omega} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I})$. Hence, to estimate the *population risk*, defined in (2) as the mean squared error of the regressor against a clean label without noise, one should instead use $\mathcal{E}_{\text{te}} - m\sigma^2$ instead.

C.5 Neural-network-like random feature models

When the labels are scalar-valued, a noteworthy interpretation of linear models is as two-layer fully connected feed-forward neural networks with a fixed first layer and a trainable second layer, or more generally as neural networks with a fully connected linear last layer where only the final layer is trained. Thus, it is natural to ask whether this interpretation also extends to the vector-valued case. The answer is yes, but with a slight caveat, which we now detail.

Let us focus on 2-layer networks, as generalizing to more general neural networks of the form described above is mostly straightforward; see Remark C.5 below. A 2-layer network with p hidden neurons, taking d-dimensional inputs and producing m-dimensional outputs, can be modeled as $\boldsymbol{f}(\boldsymbol{z}) = \boldsymbol{B}\sigma(\boldsymbol{W}\boldsymbol{z} + \boldsymbol{c})$, for weights $\boldsymbol{W} \in \mathbb{R}^{p\times d}$, $\boldsymbol{c} \in \mathbb{R}^p$, $\boldsymbol{B} \in \mathbb{R}^{m\times p}$, and an activation function σ applied elementwise. As $\boldsymbol{f}$ is linear on $\boldsymbol{B}$, it indeed is a linear model, and this becomes more apparent from the reformulation

$$\boldsymbol{f}(\boldsymbol{z}) = \begin{bmatrix} \sigma(\boldsymbol{W}\boldsymbol{z}+\boldsymbol{c})^\top & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{0} & \sigma(\boldsymbol{W}\boldsymbol{z}+\boldsymbol{c})^\top & \dots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \dots & \sigma(\boldsymbol{W}\boldsymbol{z}+\boldsymbol{c})^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta}_1 \\ \boldsymbol{\beta}_2 \\ \vdots \\ \boldsymbol{\beta}_m \end{bmatrix}$$

where $\boldsymbol{\beta}_i$ is the ith row vector of $\boldsymbol{B}$ in the column vector form. In other words, $\boldsymbol{f}$ is a linear model with a block diagonal feature map.

That said, this is exactly where the caveat arises. The issue is that the feature map is not of the form of (11), as that formulation cannot produce the precise pattern of zeros we now require. This shows that a separate analysis from Section C.4 is required for this case.

To this end, let us consider a setting where the feature map now is a block diagonal matrix

$$\boldsymbol{\psi}(\boldsymbol{z}) = \begin{bmatrix} \boldsymbol{\psi}_1(\boldsymbol{z}) & & \\ & \ddots & \\ & & \boldsymbol{\psi}_1(\boldsymbol{z}) \end{bmatrix} \in \mathbb{R}^{mp\times m}. \tag{38}$$

The notation $\boldsymbol{\psi}_1$ is used to emphasize that $\boldsymbol{\psi}_1(\boldsymbol{z})$ takes the same form as feature maps (11) for when $m = 1$. In other words, as in the scalar-valued labels setting, for some scalar valued map $g : \mathbb{R}^h \times \mathbb{R}^d \to \mathbb{R}$ we consider when $\boldsymbol{\psi}_1(\boldsymbol{z}) = \big(g(\boldsymbol{w}_1, \boldsymbol{z}), \dots, g(\boldsymbol{w}_p, \boldsymbol{z})\big)$ with $\boldsymbol{w}_1, \dots, \boldsymbol{w}_p \overset{\text{i.i.d.}}{\sim} \rho$. Let us call the linear models that arise from (random) feature maps having such block diagonal form the *neural-network-like linear models*. For convenience, we also let $\boldsymbol{\Psi}_1 \coloneqq \big[\boldsymbol{\psi}_1(\boldsymbol{z}_1) \ \cdots \ \boldsymbol{\psi}_1(\boldsymbol{z}_n)\big]^\top$.

Similar to before, we define the empirical kernel as $\hat{\boldsymbol{k}}(\boldsymbol{z}, \boldsymbol{z}') = \frac{1}{p}\boldsymbol{\psi}(\boldsymbol{z})^\top\boldsymbol{\psi}(\boldsymbol{z}')$,[5] then we get

$$\begin{aligned} \hat{\boldsymbol{k}}(\boldsymbol{z}, \boldsymbol{z}') &= \frac{1}{p}\boldsymbol{\psi}_1(\boldsymbol{z})^\top\boldsymbol{\psi}_1(\boldsymbol{z}')\boldsymbol{I} \\ &= \frac{1}{p}\sum_{i=1}^{p} g(\boldsymbol{w}_i, \boldsymbol{z})g(\boldsymbol{w}_i, \boldsymbol{z}')\boldsymbol{I}. \end{aligned}$$

Observe that $\frac{1}{p}\sum_{i=1}^{p} g(\boldsymbol{w}_i, \boldsymbol{z})g(\boldsymbol{w}_i, \boldsymbol{z}')$ is in fact what we would get as the empirical kernel when $m = 1$ so that $\boldsymbol{\psi}_1$ is used as the feature map. Thus, for $\hat{\boldsymbol{K}}_1$ denoting the matrix whose entries are $\hat{K}_{j,j'} = \frac{1}{p}\sum_{i=1}^{p} g(\boldsymbol{w}_i, \boldsymbol{z}_j)g(\boldsymbol{w}_i, \boldsymbol{z}_{j'})$, the empirical kernel matrix $\hat{\boldsymbol{K}} = \Big[\hat{\boldsymbol{k}}(\boldsymbol{z}_i, \boldsymbol{z}_j)\Big]_{i,j}$ is a block matrix whose block entries are all constant multiples of the identity matrix. More precisely, we have $\hat{\boldsymbol{K}} = \hat{\boldsymbol{K}}_1 \otimes \boldsymbol{I}$, where $\otimes$ denotes the Kronecker product of two matrices.

Recall that $\boldsymbol{x} = (x_{11}, \dots, x_{1m}, \dots, x_{n1}, \dots, x_{nm})$. Define a permutation matrix $\boldsymbol{P} \in \mathbb{R}^{nm\times nm}$ such that $\boldsymbol{P}\boldsymbol{x} = (x_{11}, \dots, x_{n1}, \dots, x_{1m}, \dots, x_{nm})$. For $j = 1, \dots, m$, denote $\boldsymbol{y}_j = (x_{1j}, \dots, x_{nj})$.

[5] Because the number of rows in $\boldsymbol{\psi}$ is now mp, one might consider choosing the factor multiplied to the sum to be $\frac{1}{mp}$ instead of $\frac{1}{p}$ to maintain consistency with Section C.4. However, as the main functionality of that factor is to convert the sum into a mean, and as each column of $\boldsymbol{\psi}$ now contains only p nonzero elements, we prefer $\frac{1}{p}$ over $\frac{1}{mp}$. As we soon observe, it turns out that such a choice also allows us to directly apply one-dimensional results to this setting.

Then as permutation matrices satisfy $\boldsymbol{P}^\top = \boldsymbol{P}^{-1}$, it holds that

$$\begin{aligned}
\left(\hat{\boldsymbol{K}} + \frac{\delta}{p}\boldsymbol{I}\right)^{-1} \text{𝔁} &= \boldsymbol{P}^\top \boldsymbol{P} \left(\hat{\boldsymbol{K}} + \frac{\delta}{p}\boldsymbol{I}\right)^{-1} \boldsymbol{P}^\top \boldsymbol{P} \text{𝔁} \\
&= \boldsymbol{P}^\top \left(\boldsymbol{P}\hat{\boldsymbol{K}}\boldsymbol{P}^\top + \frac{\delta}{p}\boldsymbol{I}\right)^{-1} \boldsymbol{P} \text{𝔁} \\
&= \boldsymbol{P}^\top \begin{bmatrix} \hat{\boldsymbol{K}}_1 + (\delta/p)\boldsymbol{I} & \cdots & \boldsymbol{0} \\ \vdots & \ddots & \vdots \\ \boldsymbol{0} & \cdots & \hat{\boldsymbol{K}}_1 + (\delta/p)\boldsymbol{I} \end{bmatrix}^{-1} \boldsymbol{P} \text{𝔁} \\
&= \boldsymbol{P}^\top \begin{bmatrix} \left(\hat{\boldsymbol{K}}_1 + (\delta/p)\boldsymbol{I}\right)^{-1} & \cdots & \boldsymbol{0} \\ \vdots & \ddots & \vdots \\ \boldsymbol{0} & \cdots & \left(\hat{\boldsymbol{K}}_1 + (\delta/p)\boldsymbol{I}\right)^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_m \end{bmatrix},
\end{aligned}$$

and thus denoting by $\hat{\text{𝔅}}$ the concatenation of $\hat{\boldsymbol{\beta}}_1, \ldots, \hat{\boldsymbol{\beta}}_m$ into a vector in $\mathbb{R}^{mp}$, we obtain

$$\begin{aligned}
\hat{\text{𝔅}} &= \frac{1}{p} \begin{bmatrix} \boldsymbol{\psi}(\boldsymbol{z}_1) & \cdots & \boldsymbol{\psi}(\boldsymbol{z}_n) \end{bmatrix} \left(\hat{\boldsymbol{K}} + \frac{\delta}{p}\boldsymbol{I}\right)^{-1} \text{𝔁} \\
&= \frac{1}{p} \begin{bmatrix} \boldsymbol{\psi}(\boldsymbol{z}_1) & \cdots & \boldsymbol{\psi}(\boldsymbol{z}_n) \end{bmatrix} \boldsymbol{P}^\top \begin{bmatrix} \left(\hat{\boldsymbol{K}}_1 + (\delta/p)\boldsymbol{I}\right)^{-1} \boldsymbol{y}_1 \\ \vdots \\ \left(\hat{\boldsymbol{K}}_1 + (\delta/p)\boldsymbol{I}\right)^{-1} \boldsymbol{y}_m \end{bmatrix} \\
&= \frac{1}{p} \begin{bmatrix} \boldsymbol{\Psi}_1^\top & \cdots & \boldsymbol{0} \\ \vdots & \ddots & \vdots \\ \boldsymbol{0} & \cdots & \boldsymbol{\Psi}_1^\top \end{bmatrix} \begin{bmatrix} \left(\hat{\boldsymbol{K}}_1 + (\delta/p)\boldsymbol{I}\right)^{-1} \boldsymbol{y}_1 \\ \vdots \\ \left(\hat{\boldsymbol{K}}_1 + (\delta/p)\boldsymbol{I}\right)^{-1} \boldsymbol{y}_m \end{bmatrix} \\
&= \frac{1}{p} \begin{bmatrix} \boldsymbol{\Psi}_1^\top \left(\hat{\boldsymbol{K}}_1 + (\delta/p)\boldsymbol{I}\right)^{-1} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{\Psi}_1^\top \left(\hat{\boldsymbol{K}}_1 + (\delta/p)\boldsymbol{I}\right)^{-1} \boldsymbol{y}_m \end{bmatrix}.
\end{aligned}$$

That is, for each $i = 1, \ldots, m$, each $\hat{\boldsymbol{\beta}}_i$ only depends on $\boldsymbol{y}_i$, the collection of the ith coordinates from $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$. In other words, each $\hat{\boldsymbol{\beta}}_i$ is computed independently from others, as if it is obtained from a kernel regression in a scalar-valued labels setting by only looking at the ith coordinates in the data. On top of this, we also have

$$\begin{aligned}
\boldsymbol{\Sigma} &= \mathbb{E}_{\boldsymbol{z}\sim\xi}[\boldsymbol{\psi}(\boldsymbol{z})\boldsymbol{\psi}(\boldsymbol{z})^\top] \\
&= \begin{bmatrix} \mathbb{E}_{\boldsymbol{z}\sim\xi}[\boldsymbol{\psi}_1(\boldsymbol{z})\boldsymbol{\psi}_1(\boldsymbol{z})^\top] & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \mathbb{E}_{\boldsymbol{z}\sim\xi}[\boldsymbol{\psi}_1(\boldsymbol{z})\boldsymbol{\psi}_1(\boldsymbol{z})^\top] & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \mathbb{E}_{\boldsymbol{z}\sim\xi}[\boldsymbol{\psi}_1(\boldsymbol{z})\boldsymbol{\psi}_1(\boldsymbol{z})^\top] \end{bmatrix},
\end{aligned}$$

allowing us to conclude that the population risk is the sum of the coordinatewise population risk.

Meanwhile, note that the mean squared error of the noise equals the sum of the mean squared errors of its coordinatewise components, irrespective of the specific noise model. Consequently, the population risk can be estimated by computing the sum of the coordinatewise estimates given by $\mathcal{R}_{\text{te}}$ from (15) evaluated in the case $m = 1$. It follows that the proper assumption to make when studying neural-network-like linear models is the singular value decomposition of the scalar-valued function $g(\boldsymbol{w}, \boldsymbol{z}) = \sum_i \sigma_i \zeta_i(\boldsymbol{w})\phi_i(\boldsymbol{z})$ and the Gaussian universality ansatz (Assumption A.1), not their multivariate counterparts.

With these established, let us now consider the expansions $\boldsymbol{f} = \left(\sum_j v_{1j}\phi_j, \ldots, \sum_j v_{mj}\phi_j\right)$ and $\hat{\boldsymbol{f}} = \left(\sum_j \hat{v}_{1j}\phi_j, \ldots, \sum_j \hat{v}_{mj}\phi_j\right)$. Denote by $\sigma_{(i)}^2$ the noise level in the ith coordinate. Then,

approximating the population risk coordinatewise using the formula (60) with $m = 1$ and then summing them up, we get

$$\mathbb{E}_{\mathcal{D}}[\mathcal{R}(\hat{\beta})] \approx \sum_{i=1}^{m} \left(\frac{1}{1 - \frac{q(p-2s)+s^2}{n(p-q)}} \left(\sum_j \left(\frac{\gamma}{\lambda_j + \gamma} - \frac{\kappa \lambda_j}{(\lambda_j + \gamma)^2} \frac{p}{p-q} \right) v_{ij}^2 + \sigma_{(i)}^2 \right) - \sigma_{(i)}^2 \right). \tag{39}$$

Recall that $s = \sum_j \frac{\lambda_j}{\lambda_j + \gamma}$ and $q = \sum_j \left(\frac{\lambda_j}{\lambda_j + \gamma} \right)^2$ for some scalars κ and γ, so they do not depend on the summation index i in (39). Therefore, denoting by $\varsigma^2 = \sum_{i=1}^m \sigma_{(i)}^2$ the overall mean squared noise, we conclude that

$$\mathbb{E}_{\mathcal{D}}[\mathcal{R}(\hat{\beta})] \approx \frac{1}{1 - \frac{q(p-2s)+s^2}{n(p-q)}} \left(\sum_{i=1}^{m} \sum_j \left(\frac{\gamma}{\lambda_j + \gamma} - \frac{\kappa \lambda_j}{(\lambda_j + \gamma)^2} \frac{p}{p-q} \right) v_{ij}^2 + \varsigma^2 \right) - \varsigma^2. \tag{40}$$

Remark C.5. Consider a neural network with a fully connected linear last layer, where only the final layer is trained. Notice that such a network can be modeled as $\boldsymbol{f}(\boldsymbol{z}) = \boldsymbol{B}\boldsymbol{g}(\boldsymbol{z}; \boldsymbol{W})$, where $\boldsymbol{g}(\boldsymbol{z}; \boldsymbol{W})$ denotes the representation obtained when the input $\boldsymbol{z}$ has passed through all layers of the neural network except for the last layer and $\boldsymbol{W}$ denotes the (frozen) parameters. Hence, we may simply replace every instance of $\sigma(\boldsymbol{W}\boldsymbol{z} + \boldsymbol{c})$ above with $\boldsymbol{g}(\boldsymbol{z}; \boldsymbol{W})$ to observe that the results obtained still apply to neural networks beyond two-layer networks.

D Missing Details for Section 4

For convenience, for a positive integer n, let us denote the set $\{1, \ldots, n\} \subset \mathbb{Z}$ by $[n]$.

D.1 Consequences of Assumption 4.1

Proposition D.1. *Assumption 4.1 implies that* $\mathbb{E}_{\boldsymbol{x} \sim \nu}\left[\|\boldsymbol{x}\|^2\right] = \operatorname{tr}(\boldsymbol{N}) < \infty$.

Proof. By the linearity of expectations and the invariance of the trace under cyclic permutations, we have

$$\operatorname{tr}\left(\mathbb{E}_{\boldsymbol{x} \sim \nu}[\boldsymbol{x}\boldsymbol{x}^\top]\right) = \mathbb{E}_{\boldsymbol{x} \sim \nu}\left[\operatorname{tr}\left(\boldsymbol{x}\boldsymbol{x}^\top\right)\right] = \mathbb{E}_{\boldsymbol{x} \sim \nu}\left[\boldsymbol{x}^\top \boldsymbol{x}\right] = \mathbb{E}_{\boldsymbol{x} \sim \nu}\left[\|\boldsymbol{x}\|^2\right] \tag{41}$$

and the conclusion follows. □

Proposition D.2. *Assumption 4.1 implies that ν has a finite first moment.*

Proof. For each $i \in [m]$, let $x_{(i)}$ denote the ith component of $\boldsymbol{x}$. Then, Jensen's inequality gives us

$$\mathbb{E}_{\boldsymbol{x} \sim \nu}\left[|x_{(i)}|\right] \leq \sqrt{\mathbb{E}_{\boldsymbol{x} \sim \nu}\left[x_{(i)}^2\right]} \leq \sqrt{\mathbb{E}_{\boldsymbol{x} \sim \nu}\left[\|\boldsymbol{x}\|^2\right]}.$$

It is then immediate that ν has a finite first moment. □

Similar arguments can be applied to ξ to obtain analogous conclusions.

D.2 Annotations to the construction of the regression model (10)

First, let us consider when both ν and ξ are distributions on $\mathbb{R}$. Let $x_{i:n}$ denote the ith order statistic of $\{x_1, \ldots, x_n\}$, *i.e.*, the ith smallest element of this set. The order statistics $z_{i:n}$ are defined analogously. When the empirical distributions $\hat{\xi}$ and $\hat{\nu}$ are defined on $\mathbb{R}$, it is well known that the OT map pairs $z_{i:n}$ with $x_{i:n}$ for each $i \in [n]$; see, *e.g.*, [48, Remark 2.28]. Hence in this case, the paired dataset we obtain is $\{(z_{i:n}, x_{i:n})\}_{i=1,\ldots,n}$, from which we can construct the regression model as

$$x_{i:n} = G^*(z_{i:n}) + \check{\epsilon}_i. \tag{42}$$

Here, we note that the source of stochasticity in the noise term $\check{\epsilon}_i = x_{i:n} - G^*(z_{i:n})$ arises from the sampling process used to obtain $\{z_1, \ldots, z_n\}$ and $\{x_1, \ldots, x_n\}$, rather than from additive label noise on $G^*(z_{i:n})$, as is commonly assumed in regression tasks.

Converting (42) back to a regression model in terms of $\{z_1, \ldots, z_n\}$ and $\{x_1, \ldots, x_n\}$, we obtain

$$x_{\tilde{\pi}(i)} = G^*(z_i) + \epsilon_i.$$

Here, since $\{z_1, \ldots, z_n\}$ are i.i.d. samples from ξ, the joint distribution of $\{z_1, \ldots, z_n\}$ is invariant under any permutation of the indices. Consequently, each order statistic $z_{i:n}$ has probability $\frac{1}{n}$ of coinciding with any particular sample z_j, for $j \in [n]$. Then, as we may regard the triplets $\{(z_i, x_{\tilde{\pi}(i)}, \epsilon_i)\}_{i=1,\ldots,n}$ as a random permutation of the triplets $\{(z_{i:n}, x_{i:n}, \check{\epsilon}_i)\}_{i=1,\ldots,n}$, each noise ϵ_i also has probability $\frac{1}{n}$ of coinciding with any particular $\check{\epsilon}_j$, for $j \in [n]$. It follows that the (marginal) distribution of ϵ_i is a uniform mixture of the distributions of $\check{\epsilon}_j$, $j \in [n]$.

Now, let us consider the problem in general dimensions. Once we move on to higher dimensions, we unfortunately no longer have an explicit characterization of $\tilde{\pi}$, as we did in the one-dimensional setting via order statistics. Nonetheless, the core idea remains the same.

Given the samples $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$, it is still possible to impose an order on them, for example, by lexicographical ordering, as mentioned in Section 4.2. Let $\boldsymbol{z}_{1:n}, \ldots, \boldsymbol{z}_{n:n}$ denote such ordered samples. We then can relabel $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ as $\{\boldsymbol{x}_{1:n}, \ldots, \boldsymbol{x}_{n:n}\}$ so that $\{(\boldsymbol{z}_{i:n}, \boldsymbol{x}_{i:n})\}_{i=1,\ldots,n}$ forms the pairing induced by the discrete OT coupling. This leads to a regression model analogous to (42), namely

$$\boldsymbol{x}_{i:n} = G^*(\boldsymbol{z}_{i:n}) + \check{\boldsymbol{\epsilon}}_i.$$

As in the one-dimensional setting, we can convert this back to a regression model expressed in terms of $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$ and $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, to get

$$\boldsymbol{x}_{\tilde{\pi}(i)} = G^*(\boldsymbol{z}_i) + \boldsymbol{\epsilon}_i.$$

Again, the (marginal) distribution of each $\boldsymbol{\epsilon}_i$ is a uniform mixture of the distributions of $\check{\boldsymbol{\epsilon}}_j$, $j \in [n]$.

D.3 Statistics of the noises in the regression model (10)

Let us recall the OT based regression formulation (10) of generative modeling, namely

$$\boldsymbol{x}_{\tilde{\pi}(i)} = G^*(\boldsymbol{z}_i) + \boldsymbol{\epsilon}_i.$$

Let $p_{\boldsymbol{\epsilon}_i}$ denote the distribution of $\boldsymbol{\epsilon}_i$. Recall the definition (9) of $\check{\boldsymbol{\epsilon}}_i$, and let $p_{\check{\boldsymbol{\epsilon}}_i}$ denote the distribution of $\check{\boldsymbol{\epsilon}}_i$. Then, we have already shown that

$$p_{\boldsymbol{\epsilon}_i} = \frac{1}{n} \sum_{j=1}^{n} p_{\check{\boldsymbol{\epsilon}}_j}$$

for any $i \in [n]$. Hence, $\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_n$ are identically distributed; however, we should note that they are not independent. Nonetheless, the invariance under random permutation produces a symmetrization effect, which in turn yields a mixture distribution with zero-mean noise, as shown below.

Proposition D.3. *For any $i \in [n]$, it holds that $\mathbb{E}[\boldsymbol{\epsilon}_i] = \mathbf{0}$.*

Proof. By the linearity of expectations, we have

$$\begin{aligned}
\mathbb{E}[\boldsymbol{\epsilon}_i] &= \frac{1}{n} \sum_{j=1}^{n} \mathbb{E}[\check{\boldsymbol{\epsilon}}_j] \\
&= \frac{1}{n} \sum_{j=1}^{n} \mathbb{E}[\boldsymbol{x}_{j:n} - G^*(\boldsymbol{z}_{j:n})] \\
&= \frac{1}{n} \mathbb{E}\left[\sum_{j=1}^{n} \boldsymbol{x}_{j:n}\right] - \frac{1}{n} \mathbb{E}\left[\sum_{j=1}^{n} G^*(\boldsymbol{z}_{j:n})\right] \\
&= \frac{1}{n} \mathbb{E}\left[\sum_{j=1}^{n} \boldsymbol{x}_j\right] - \frac{1}{n} \mathbb{E}\left[\sum_{j=1}^{n} G^*(\boldsymbol{z}_j)\right].
\end{aligned}$$

Here, as G^* is a transport map from ξ to ν, each $G^*(z_j)$ follows the distribution ν. Hence, from $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$ and $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ each being i.i.d. samples, we have

$$\frac{1}{n}\,\mathbb{E}\left[\sum_{j=1}^{n} \boldsymbol{x}_j\right] - \frac{1}{n}\,\mathbb{E}\left[\sum_{j=1}^{n} G^*(\boldsymbol{z}_j)\right] = \mathbb{E}_{\boldsymbol{x}\sim\nu}[\boldsymbol{x}] - \mathbb{E}_{G(\boldsymbol{z})\sim\nu}[G(\boldsymbol{z})] = \mathbf{0}.$$

It follows that $\mathbb{E}[\boldsymbol{\epsilon}_i] = \mathbf{0}$. □

We can also show that the second moments of the noises are bounded. Notice that this also implies that $\mathbb{E}[\mathrm{tr}(\boldsymbol{\epsilon}_i\boldsymbol{\epsilon}_i^\top)] = \mathbb{E}[\|\boldsymbol{\epsilon}_i\|^2]$ is bounded, as we show below.

Proposition D.4. *Under Assumption 4.1, let $\sigma^2 \coloneqq \mathrm{tr}(\boldsymbol{N}) + \mathrm{tr}(\boldsymbol{\Xi})$. Then for any $i \in [n]$, it holds that*

$$\mathbb{E}[\|\boldsymbol{\epsilon}_i\|^2] = \frac{1}{n}\sum_{j=1}^{n} \mathbb{E}[\|\check{\boldsymbol{\epsilon}}_j\|^2] \le 8\sigma^2. \tag{43}$$

Proof. For the equality part of (43), recall that $\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_n$ is a permutation of $\check{\boldsymbol{\epsilon}}_1, \ldots, \check{\boldsymbol{\epsilon}}_n$ to get

$$\sum_{i=1}^{n} \mathbb{E}[\|\boldsymbol{\epsilon}_i\|^2] = \sum_{j=1}^{n} \mathbb{E}[\|\check{\boldsymbol{\epsilon}}_j\|^2].$$

Since the marginal distributions of $\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_n$ are identical, we conclude that

$$\mathbb{E}[\|\boldsymbol{\epsilon}_i\|^2] = \frac{1}{n}\sum_{j=1}^{n} \mathbb{E}[\|\check{\boldsymbol{\epsilon}}_j\|^2]$$

for any $i \in [n]$.

For the inequality part of (43), observe that Young's inequality yields

$$\begin{aligned}
\frac{1}{n}\sum_{j=1}^{n} \mathbb{E}[\|\check{\boldsymbol{\epsilon}}_j\|^2] &= \frac{1}{n}\,\mathbb{E}\left[\sum_{j=1}^{n} \|\boldsymbol{x}_{j:n} - G^*(\boldsymbol{z}_{j:n})\|^2\right] \\
&\le \frac{2}{n}\,\mathbb{E}\left[\sum_{j=1}^{n} \|\boldsymbol{x}_{j:n} - \boldsymbol{z}_{j:n}\|^2\right] + \frac{4}{n}\,\mathbb{E}\left[\sum_{j=1}^{n} \|\boldsymbol{z}_{j:n}\|^2\right] + \frac{4}{n}\,\mathbb{E}\left[\sum_{j=1}^{n} \|G^*(\boldsymbol{z}_{j:n})\|^2\right] \\
&= \frac{2}{n}\,\mathbb{E}\left[\sum_{j=1}^{n} \|\boldsymbol{x}_{\tilde{\pi}(j)} - \boldsymbol{z}_j\|^2\right] + \frac{4}{n}\,\mathbb{E}\left[\sum_{j=1}^{n} \|\boldsymbol{z}_j\|^2\right] + \frac{4}{n}\,\mathbb{E}\left[\sum_{j=1}^{n} \|G^*(\boldsymbol{z}_j)\|^2\right] \qquad (44)
\end{aligned}$$

where the last line follows from the fact that $(\boldsymbol{z}_{1:n}, \boldsymbol{x}_{1:n}) \ldots, (\boldsymbol{z}_{n:n}, \boldsymbol{x}_{n:n})$ is a permutation of the pairs $(\boldsymbol{z}_1, \boldsymbol{x}_{\tilde{\pi}(1)}), \ldots, (\boldsymbol{z}_n, \boldsymbol{x}_{\tilde{\pi}(n)})$. Now recall that $\tilde{\pi} = \arg\min_{\pi\in S_n} \frac{1}{n}\sum_{i=1}^{n} \|\boldsymbol{x}_{\pi(i)} - \boldsymbol{z}_i\|^2$. Hence, with explicitly writing out the source of randomness over which the expectation is taken, it holds that

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{z}_1,\ldots,\boldsymbol{z}_n \overset{\text{i.i.d.}}{\sim} \xi,\ \boldsymbol{x}_1,\ldots,\boldsymbol{x}_n \overset{\text{i.i.d.}}{\sim} \nu}\left[\sum_{j=1}^{n} \|\boldsymbol{x}_{\tilde{\pi}(j)} - \boldsymbol{z}_j\|^2\right] \\
&\quad\le \mathbb{E}_{\boldsymbol{z}_1,\ldots,\boldsymbol{z}_n \overset{\text{i.i.d.}}{\sim} \xi,\ \boldsymbol{x}_1,\ldots,\boldsymbol{x}_n \overset{\text{i.i.d.}}{\sim} \nu,\ \pi\sim\mathrm{Uniform}(S_n)}\left[\sum_{j=1}^{n} \|\boldsymbol{x}_{\pi(j)} - \boldsymbol{z}_j\|^2\right] \\
&\quad= \mathbb{E}_{\boldsymbol{z}_1,\ldots,\boldsymbol{z}_n \overset{\text{i.i.d.}}{\sim} \xi}\left[\mathbb{E}_{\pi\sim\mathrm{Uniform}(S_n)}\left[\mathbb{E}_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n \overset{\text{i.i.d.}}{\sim} \nu}\left[\sum_{j=1}^{n} \|\boldsymbol{x}_{\pi(j)} - \boldsymbol{z}_j\|^2 \,\middle|\, \pi, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\right]\right]\right]
\end{aligned}$$

where the last line follows from the law of total expectation. Here, because $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ are i.i.d. samples, notice that their joint distribution is invariant under a permutation. Hence, continuing from

the last line of the above, we obtain

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{z}_1,\ldots,\boldsymbol{z}_n \overset{\text{i.i.d.}}{\sim} \xi}\left[\mathbb{E}_{\pi\sim\mathrm{Uniform}(S_n)}\left[\mathbb{E}_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n \overset{\text{i.i.d.}}{\sim} \nu}\left[\sum_{j=1}^n \|\boldsymbol{x}_{\pi(j)} - \boldsymbol{z}_j\|^2 \middle| \pi, \boldsymbol{z}_1,\ldots,\boldsymbol{z}_n\right]\right]\right] \\
&= \mathbb{E}_{\boldsymbol{z}_1,\ldots,\boldsymbol{z}_n \overset{\text{i.i.d.}}{\sim} \xi}\left[\mathbb{E}_{\pi\sim\mathrm{Uniform}(S_n)}\left[\mathbb{E}_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n \overset{\text{i.i.d.}}{\sim} \nu}\left[\sum_{j=1}^n \|\boldsymbol{x}_{j} - \boldsymbol{z}_j\|^2 \middle| \pi, \boldsymbol{z}_1,\ldots,\boldsymbol{z}_n\right]\right]\right] \\
&= \mathbb{E}_{\boldsymbol{z}_1,\ldots,\boldsymbol{z}_n \overset{\text{i.i.d.}}{\sim} \xi,\, \boldsymbol{x}_1,\ldots,\boldsymbol{x}_n \overset{\text{i.i.d.}}{\sim} \nu}\left[\sum_{j=1}^n \|\boldsymbol{x}_{j} - \boldsymbol{z}_j\|^2\right] \\
&= n\,\mathbb{E}_{\boldsymbol{z}\sim\xi,\,\boldsymbol{x}\sim\nu}\left[\|\boldsymbol{x}-\boldsymbol{z}\|^2\right] \\
&\le 2n\,\mathbb{E}_{\boldsymbol{x}\sim\nu}\left[\|\boldsymbol{x}\|^2\right] + 2n\,\mathbb{E}_{\boldsymbol{z}\sim\xi}\left[\|\boldsymbol{z}\|^2\right]
\end{aligned}$$

where the inequality step is once again by Young's inequality. Meanwhile, for the third term in (44), as G^* is a transport map from ξ to ν, it holds that

$$\mathbb{E}_{\boldsymbol{z}_1,\ldots,\boldsymbol{z}_n \overset{\text{i.i.d.}}{\sim} \xi}\left[\sum_{j=1}^n \|G^*(\boldsymbol{z}_j)\|^2\right] = n\,\mathbb{E}_{\boldsymbol{z}\sim\xi}\left[\|G^*(\boldsymbol{z})\|^2\right] = n\,\mathbb{E}_{\boldsymbol{x}\sim\nu}\left[\|\boldsymbol{x}\|^2\right].$$

Collecting the intermediate results, from (44) we obtain that

$$\frac{1}{n}\sum_{j=1}^n \mathbb{E}[\|\boldsymbol{\epsilon}_j\|^2] \le 8\,\mathbb{E}_{\boldsymbol{x}\sim\nu}\left[\|\boldsymbol{x}\|^2\right] + 8\,\mathbb{E}_{\boldsymbol{z}\sim\xi}\left[\|\boldsymbol{z}\|^2\right].$$

The conclusion is now immediate from the identity shown in (41). □

D.4 Learning optimal transport map from finite training data and presampled latent vectors

A straightforward way to approximate G^* is to minimize the ℓ_2-regularized least squares objective as

$$\min_G \sum_{i=1}^n \frac{1}{2}\|G(\boldsymbol{z}_i) - \boldsymbol{x}_{\tilde{\pi}(i)}\|^2 + \frac{\delta}{2}\|\boldsymbol{\beta}\|^2 \tag{45}$$

for a *ridge parameter* δ, which upper bounds $\hat{\mathcal{R}}_{\mathrm{fully}}(G)$ in (4) when D is the squared 2-Wasserstein distance. This problem is possibly nonconvex, yet empirical evidence suggests that it can be effectively solved. We thus use the optimal solution of (45) for the benign overfitting analysis in Section 5, as in regression [8, 57, 61]. The computation of $\tilde{\pi}$ remains expensive [19], and we leave developing more efficient methods as future work.

E Missing details for Section 5.2

E.1 Proof of Lemma 5.2

Lemma E.1 (Lemma 5.2)**.** *The population risk $\mathcal{R}(\hat{\boldsymbol{\beta}}(\delta))$ satisfies*

$$\mathcal{R}(\hat{\boldsymbol{\beta}}(\delta)) \le 2\boldsymbol{\beta}^{*\top}\boldsymbol{B}_\delta\boldsymbol{\beta}^* + 2\boldsymbol{\mathfrak{e}}^\top\boldsymbol{C}_\delta\boldsymbol{\mathfrak{e}}$$

where we define $\boldsymbol{B}_\delta \coloneqq \boldsymbol{M}_\delta\boldsymbol{\Sigma}\boldsymbol{M}_\delta$ and $\boldsymbol{C}_\delta \coloneqq (\boldsymbol{\Psi}\boldsymbol{\Psi}^\top + \delta\boldsymbol{I})^{-1}\boldsymbol{\Psi}\boldsymbol{\Sigma}\boldsymbol{\Psi}^\top(\boldsymbol{\Psi}\boldsymbol{\Psi}^\top + \delta\boldsymbol{I})^{-1}$ with $\boldsymbol{M}_\delta \coloneqq \boldsymbol{I} - \boldsymbol{\Psi}^\top(\boldsymbol{\Psi}\boldsymbol{\Psi}^\top + \delta\boldsymbol{I})^{-1}\boldsymbol{\Psi}$.

Proof. For any $\boldsymbol{\beta}$, the population risk can be expressed as

$$\begin{aligned}
\mathcal{R}(\boldsymbol{\beta}) &= \mathbb{E}_{\boldsymbol{z}_0}\left[\left\|\boldsymbol{x}_0 - \boldsymbol{\psi}(\boldsymbol{z}_0)^\top\boldsymbol{\beta}\right\|^2\right] = \mathbb{E}_{\boldsymbol{z}_0}\left[\left\|\boldsymbol{\psi}(\boldsymbol{z}_0)^\top\boldsymbol{\beta}^* - \boldsymbol{\psi}(\boldsymbol{z}_0)^\top\boldsymbol{\beta}\right\|^2\right] \\
&= \mathbb{E}_{\boldsymbol{z}_0}[(\boldsymbol{\beta}^* - \boldsymbol{\beta})^\top\boldsymbol{\psi}(\boldsymbol{z}_0)\boldsymbol{\psi}(\boldsymbol{z}_0)^\top(\boldsymbol{\beta}^* - \boldsymbol{\beta})] \\
&= (\boldsymbol{\beta}^* - \boldsymbol{\beta})^\top\,\mathbb{E}_{\boldsymbol{z}_0}[\boldsymbol{\psi}(\boldsymbol{z}_0)\boldsymbol{\psi}(\boldsymbol{z}_0)^\top](\boldsymbol{\beta}^* - \boldsymbol{\beta}) \\
&= (\boldsymbol{\beta}^* - \boldsymbol{\beta})^\top\boldsymbol{\Sigma}(\boldsymbol{\beta}^* - \boldsymbol{\beta}) \\
&= \|\boldsymbol{\beta} - \boldsymbol{\beta}^*\|_{\boldsymbol{\Sigma}}^2.
\end{aligned}$$

As $\hat{\boldsymbol{\beta}}(\delta) = \boldsymbol{\Psi}^\top(\boldsymbol{\Psi}\boldsymbol{\Psi}^\top + \delta \boldsymbol{I})^{-1}\text{ɤ}$ and $\text{ɤ} = \boldsymbol{\Psi}\boldsymbol{\beta}^* + \mathfrak{e}$, the population risk of the minimum-norm solution is

$$
\begin{aligned}
\mathcal{R}(\hat{\boldsymbol{\beta}}(\delta)) &= \mathbb{E}_{\boldsymbol{z}_0}\left[\left\|\boldsymbol{\psi}(\boldsymbol{z}_0)^\top\boldsymbol{\beta}^* - \boldsymbol{\psi}(\boldsymbol{z}_0)^\top\hat{\boldsymbol{\beta}}(\delta)\right\|^2\right] \\
&= \mathbb{E}_{\boldsymbol{z}_0}\left[\left\|\boldsymbol{\psi}(\boldsymbol{z}_0)^\top\boldsymbol{\beta}^* - \boldsymbol{\psi}(\boldsymbol{z}_0)^\top\boldsymbol{\Psi}^\top(\boldsymbol{\Psi}\boldsymbol{\Psi}^\top+\delta\boldsymbol{I})^{-1}\text{ɤ}\right\|^2\right] \\
&= \mathbb{E}_{\boldsymbol{z}_0}\left[\left\|\boldsymbol{\psi}(\boldsymbol{z}_0)^\top\boldsymbol{\beta}^* - \boldsymbol{\psi}(\boldsymbol{z}_0)^\top\boldsymbol{\Psi}^\top(\boldsymbol{\Psi}\boldsymbol{\Psi}^\top+\delta\boldsymbol{I})^{-1}\boldsymbol{\Psi}\boldsymbol{\beta}^* - \boldsymbol{\psi}(\boldsymbol{z}_0)^\top\boldsymbol{\Psi}^\top(\boldsymbol{\Psi}\boldsymbol{\Psi}^\top+\delta\boldsymbol{I})^{-1}\mathfrak{e}\right\|^2\right] \\
&= \mathbb{E}_{\boldsymbol{z}_0}\left[\left\|\boldsymbol{\psi}(\boldsymbol{z}_0)^\top[I - \boldsymbol{\Psi}^\top(\boldsymbol{\Psi}\boldsymbol{\Psi}^\top+\delta\boldsymbol{I})^{-1}\boldsymbol{\Psi}]\boldsymbol{\beta}^* - \boldsymbol{\psi}(\boldsymbol{z}_0)^\top\boldsymbol{\Psi}^\top(\boldsymbol{\Psi}\boldsymbol{\Psi}^\top+\delta\boldsymbol{I})^{-1}\mathfrak{e}\right\|^2\right] \\
&\le 2\,\mathbb{E}_{\boldsymbol{z}_0}\left[\left\|\boldsymbol{\psi}(\boldsymbol{z}_0)^\top[I - \boldsymbol{\Psi}^\top(\boldsymbol{\Psi}\boldsymbol{\Psi}^\top+\delta\boldsymbol{I})^{-1}\boldsymbol{\Psi}]\boldsymbol{\beta}^*\right\|^2\right] + 2\,\mathbb{E}_{\boldsymbol{z}_0}\left[\left\|\boldsymbol{\psi}(\boldsymbol{z}_0)^\top\boldsymbol{\Psi}^\top(\boldsymbol{\Psi}\boldsymbol{\Psi}^\top+\delta\boldsymbol{I})^{-1}\mathfrak{e}\right\|^2\right] \\
&= 2\,\mathbb{E}_{\boldsymbol{z}_0}\left[\boldsymbol{\beta}^{*\top}[I - \boldsymbol{\Psi}^\top(\boldsymbol{\Psi}\boldsymbol{\Psi}^\top+\delta\boldsymbol{I})^{-1}\boldsymbol{\Psi}]\boldsymbol{\psi}(\boldsymbol{z}_0)\boldsymbol{\psi}(\boldsymbol{z}_0)^\top[I - \boldsymbol{\Psi}^\top(\boldsymbol{\Psi}\boldsymbol{\Psi}^\top+\delta\boldsymbol{I})^{-1}\boldsymbol{\Psi}]\boldsymbol{\beta}^*\right] \\
&\quad + 2\,\mathbb{E}_{\boldsymbol{z}_0}\left[\mathfrak{e}^\top(\boldsymbol{\Psi}\boldsymbol{\Psi}^\top+\delta\boldsymbol{I})^{-1}\boldsymbol{\Psi}\boldsymbol{\psi}(\boldsymbol{z}_0)\boldsymbol{\psi}(\boldsymbol{z}_0)^\top\boldsymbol{\Psi}^\top(\boldsymbol{\Psi}\boldsymbol{\Psi}^\top+\delta\boldsymbol{I})^{-1}\mathfrak{e}\right].
\end{aligned}
$$

Therefore, by analogously defining $\boldsymbol{B}_\delta$ and $\boldsymbol{C}_\delta$, Lemma 5.2 follows immediately. □

E.2 Upper bound of the variance term in Lemma 5.2

Previous analyses [8, 61] relied critically on the assumption that the noise variables $\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_n$ are i.i.d. in order to derive a high-probability bound for $\mathfrak{e}^\top\boldsymbol{C}_\delta\mathfrak{e}$. Since this assumption fails in our setting, their argument does not directly apply.

Let us elaborate on Assumption 5.1, or more precisely, on how we adopt the standard Gaussian universality type ansatz on the feature distribution that underlies the analyses of Simon et al. [56, 57]. Informally, for the purpose of computing the risk estimate, we replace the evaluations of the eigenfunctions by an idealized surrogate with i.i.d. Gaussian entries, so that, within this surrogate, the residuals may be treated as independent of the design. We conjecture that this effective independence could be further supported by delocalization results for eigenvectors of random matrices, and we leave a detailed analysis of this direction to future work. Nonetheless, the Gaussian feature surrogate is known to give accurate predictions for empirical risk curves in related settings [56, 57], and our experiments in Appendix G indicate that our estimate, based on this more general noise model, enjoys a comparable level of agreement. Hence, we use this interpretation of Assumption 5.1 as an idealized surrogate throughout this section and subsequently also in Section F.2.

Recently, in the non-i.i.d. setting, Lee and Lee [38] presented a decomposition of the expectation of $\mathfrak{e}^\top\boldsymbol{C}_\delta\mathfrak{e} = \mathrm{tr}(\boldsymbol{C}_\delta\mathfrak{e}\mathfrak{e}^\top)$ into $\mathbb{E}[\mathrm{tr}(\boldsymbol{C}_\delta)]$ and $\mathbb{E}[\mathrm{tr}(\mathfrak{e}\mathfrak{e}^\top)]$, under the assumption that the matrix $\boldsymbol{\Psi}$ is left spherical. Our Assumption 5.1 in fact implies this left spherical property, so we have the following.

Lemma E.2. *Under Assumption 5.1, it holds that* $\mathbb{E}[\mathfrak{e}^\top\boldsymbol{C}_\delta\mathfrak{e}] = \frac{1}{nm}\mathbb{E}[\mathrm{tr}(\boldsymbol{C}_\delta)]\mathbb{E}[\mathrm{tr}(\mathfrak{e}\mathfrak{e}^\top)]$.

Proof. The proof is mostly identical to the derivations presented in Appendix E.1 of [38], but with some modifications that are necessary in order to apply them to our setting. Let us define

$$\boldsymbol{B} \coloneqq \boldsymbol{\Sigma}^{1/2}\boldsymbol{\Psi}^\top(\boldsymbol{\Psi}\boldsymbol{\Psi}^\top + \delta\boldsymbol{I})^{-1} \in \mathbb{R}^{p\times nm}$$

and consider its compact singular value decomposition $\boldsymbol{B} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top$, so that $\boldsymbol{U} \in \mathbb{R}^{p\times\mathrm{rank}(\boldsymbol{B})}$ satisfies $\boldsymbol{U}^\top\boldsymbol{U} = \boldsymbol{I}$. Let $\Omega \coloneqq \mathbb{E}[\mathfrak{e}\mathfrak{e}^\top] \in \mathbb{R}^{nm\times nm}$ and $\boldsymbol{T} \coloneqq \Omega^{1/2}$. Then, the orthogonal invariance of Frobenius norms yield the identity

$$\|\boldsymbol{B}\boldsymbol{T}\|_F^2 = \|\boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top\boldsymbol{T}\|_F^2 = \|\boldsymbol{D}\boldsymbol{V}^\top\boldsymbol{T}\|_F^2.$$

Thus, following the steps and the notations of the proof of Theorem E.1 of Lee and Lee [38], we obtain $\|\boldsymbol{B}\boldsymbol{T}\|_F^2 = a(\boldsymbol{\Psi};\delta)^\top\Gamma(\boldsymbol{\Psi};\delta)b$. Here, Assumption 5.1 allows us to leverage the isotropic Gaussian nature of the features. This grants left-spherical symmetry and independence between the features and the noise, which are properties implied by the universality equivalence to an independent Gaussian. The proof is then completed by following the remaining derivation of Theorem E.1 in Lee and Lee [38]. □

In [38], $\mathbb{E}[\mathrm{tr}(\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top)]$ is directly assumed to be bounded, and stronger conditions are imposed to analyze the asymptotic behavior of the risk. In our setting, $\mathbb{E}[\mathrm{tr}(\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top)]$ is bounded above under Assumption 4.1,

as we show in Proposition D.4. We note that a positive ridge parameter is crucial for bounding $\mathbb{E}[\mathrm{tr}(\boldsymbol{C}_\delta)]$, as it allows a uniform bound on $\boldsymbol{C}_\delta$. Otherwise, in the ridgeless $\delta = 0$ case, $\mathbb{E}[\mathrm{tr}(\boldsymbol{C}_\delta)]$ would diverge when $\boldsymbol{\Psi}\boldsymbol{\Psi}^\top$ has near-zero eigenvalues. We leave the ridgeless limit for future work, since controlling the minimum eigenvalue near zero is extremely challenging, as also discussed by Tsigler and Bartlett [61].

E.3 Proof of Theorem 5.3

We analyze an expectation bound to effectively decouple noise from features as in Lee and Lee [38], bypassing the intractable noise structure beyond a bound on its expectation. We note that this analysis is not a direct extension of high-probability bounds as is typically the case (*e.g.*, Lemma A.3 in [55]), because under the ridgeless regression setting, small nonzero singular values of $\boldsymbol{\Psi}$ can lead to an unbounded expectation. To mitigate this, we focus on ridge regression with $\delta > 0$.[6] Consequently, following the decomposition in Lemma 5.2, our task reduces to establishing upper bounds for both bias and variance terms.

E.3.1 Upper bound of bias term

We first establish an upper bound on the bias term $\boldsymbol{B}_\delta$. The following technical lemma is essential for our derivation.

Lemma E.3. *For any $\delta > 0$, the followings hold.*

(i) $\left\|\boldsymbol{I} - \boldsymbol{\Psi}^\top(\boldsymbol{\Psi}\boldsymbol{\Psi}^\top + \delta\boldsymbol{I})^{-1}\boldsymbol{\Psi}\right\|_{L^2} \leq 1$

(ii) $\left\|\boldsymbol{\Psi}^\top(\boldsymbol{\Psi}\boldsymbol{\Psi}^\top + \delta\boldsymbol{I})^{-2}\boldsymbol{\Psi}\right\|_{L^2} \leq \dfrac{1}{4\delta}$

Proof. To establish the bound (i), we analyze the spectrum of the operator. For simplicity, following the notation in Lemma 5.2, let us denote $\boldsymbol{M}_\delta \coloneqq \boldsymbol{I} - \boldsymbol{\Psi}^\top(\boldsymbol{\Psi}\boldsymbol{\Psi}^\top + \delta\boldsymbol{I})^{-1}\boldsymbol{\Psi}$. The Woodbury matrix identity gives

$$\boldsymbol{\Psi}^\top(\boldsymbol{\Psi}\boldsymbol{\Psi}^\top + \delta\boldsymbol{I})^{-1} = (\boldsymbol{\Psi}^\top\boldsymbol{\Psi} + \delta\boldsymbol{I})^{-1}\boldsymbol{\Psi}^\top, \tag{46}$$

from which we obtain that

$$\begin{aligned}
\boldsymbol{M}_\delta &= \boldsymbol{I} - \boldsymbol{\Psi}^\top(\boldsymbol{\Psi}\boldsymbol{\Psi}^\top + \delta\boldsymbol{I})^{-1}\boldsymbol{\Psi} \\
&= \boldsymbol{I} - (\boldsymbol{\Psi}^\top\boldsymbol{\Psi} + \delta\boldsymbol{I})^{-1}\boldsymbol{\Psi}^\top\boldsymbol{\Psi} \\
&= (\boldsymbol{\Psi}^\top\boldsymbol{\Psi} + \delta\boldsymbol{I})^{-1}\big((\boldsymbol{\Psi}^\top\boldsymbol{\Psi} + \delta\boldsymbol{I}) - \boldsymbol{\Psi}^\top\boldsymbol{\Psi}\big) \\
&= \delta(\boldsymbol{\Psi}^\top\boldsymbol{\Psi} + \delta\boldsymbol{I})^{-1}.
\end{aligned} \tag{47}$$

Since $\boldsymbol{\Psi}^\top\boldsymbol{\Psi}$ is positive semidefinite, each eigenvalue of $\boldsymbol{M}_\delta$ is of the form $\delta/(a+\delta)$ for some eigenvalue $a \geq 0$ of $\boldsymbol{\Psi}^\top\boldsymbol{\Psi}$. Here, notice that for any $\alpha \geq 0$, regardless of whether α is an eigenvalue of $\boldsymbol{\Psi}^\top\boldsymbol{\Psi}$ or not, whenever $\delta > 0$ it holds that

$$0 < \frac{\delta}{\alpha + \delta} \leq 1.$$

Therefore, every eigenvalue of $\boldsymbol{M}_\delta$ lies in the interval $(0, 1]$, and the assertion is immediate.

To show the bound (ii), let us first define $\boldsymbol{Q} \coloneqq \boldsymbol{\Psi}^\top(\boldsymbol{\Psi}\boldsymbol{\Psi}^\top + \delta\boldsymbol{I})^{-2}\boldsymbol{\Psi}$. Once again by the Woodbury matrix identity, we similarly obtain

$$\boldsymbol{Q} = \boldsymbol{\Psi}^\top\boldsymbol{\Psi}(\boldsymbol{\Psi}^\top\boldsymbol{\Psi} + \delta\boldsymbol{I})^{-2}.$$

Thus, each eigenvalue of $\boldsymbol{Q}$ is of the form $a/(a+\delta)^2$ for some eigenvalue $a \geq 0$ of $\boldsymbol{\Psi}^\top\boldsymbol{\Psi}$. To bound this quantity, observe that the AM–GM inequality gives $(a+\delta)^2 \geq 4a\delta$, and hence

$$0 \leq \frac{a}{(a+\delta)^2} \leq \frac{1}{4\delta}.$$

Since this holds for every eigenvalue of $\boldsymbol{Q}$, the assertion follows. □

[6] While Tsigler and Bartlett [61] also consider the ridge regression setting, their work focuses on different generalized assumptions and potentially negative ridge parameters, complicating the expectation analysis.

We are now ready to derive an upper bound for the bias term $\boldsymbol{B}_\delta$. Let us start by observing that (47) and (46) together give

$$\begin{aligned}\boldsymbol{M}_\delta \boldsymbol{\Psi}^\top &= \delta(\boldsymbol{\Psi}^\top \boldsymbol{\Psi} + \delta \boldsymbol{I})^{-1} \boldsymbol{\Psi}^\top \\ &= \delta \boldsymbol{\Psi}^\top (\boldsymbol{\Psi} \boldsymbol{\Psi}^\top + \delta \boldsymbol{I})^{-1}.\end{aligned}$$

As $\boldsymbol{B}_\delta = \boldsymbol{M}_\delta \boldsymbol{\Sigma} \boldsymbol{M}_\delta$, writing $\boldsymbol{\Sigma} = (\boldsymbol{\Sigma} - \frac{1}{n}\boldsymbol{\Psi}^\top \boldsymbol{\Psi}) + \frac{1}{n}\boldsymbol{\Psi}^\top \boldsymbol{\Psi}$, we obtain the identity

$$\begin{aligned}\boldsymbol{B}_\delta &= \boldsymbol{M}_\delta \left(\boldsymbol{\Sigma} - \frac{1}{n}\boldsymbol{\Psi}^\top \boldsymbol{\Psi} \right) \boldsymbol{M}_\delta + \frac{1}{n} (\boldsymbol{M}_\delta \boldsymbol{\Psi}^\top)(\boldsymbol{M}_\delta \boldsymbol{\Psi}^\top)^\top \\ &= \boldsymbol{M}_\delta \left(\boldsymbol{\Sigma} - \frac{1}{n}\boldsymbol{\Psi}^\top \boldsymbol{\Psi} \right) \boldsymbol{M}_\delta + \frac{\delta^2}{n} \boldsymbol{\Psi}^\top (\boldsymbol{\Psi} \boldsymbol{\Psi}^\top + \delta \boldsymbol{I})^{-2} \boldsymbol{\Psi}.\end{aligned}$$

Then, we first establish the bound for $\boldsymbol{\beta}^{*\top} \boldsymbol{B}_\delta \boldsymbol{\beta}^*$. Applying the operator norm bounds derived in Lemma E.3, we get

$$\begin{aligned}\boldsymbol{\beta}^{*\top} \boldsymbol{B}_\delta \boldsymbol{\beta}^* &\leq \|\boldsymbol{\beta}^*\|^2 \|\boldsymbol{M}_\delta\|^2 \left\| \boldsymbol{\Sigma} - \frac{1}{n}\boldsymbol{\Psi}^\top \boldsymbol{\Psi} \right\| + \frac{\delta^2}{n} \|\boldsymbol{\beta}^*\|^2 \left\| \boldsymbol{\Psi}^\top (\boldsymbol{\Psi} \boldsymbol{\Psi}^\top + \delta \boldsymbol{I})^{-2} \boldsymbol{\Psi} \right\| \\ &\leq \|\boldsymbol{\beta}^*\|^2 \left(\left\| \boldsymbol{\Sigma} - \frac{1}{n}\boldsymbol{\Psi}^\top \boldsymbol{\Psi} \right\| + \frac{\delta^2}{n} \cdot \frac{1}{4\delta} \right) \\ &= \|\boldsymbol{\beta}^*\|^2 \left(\left\| \boldsymbol{\Sigma} - \frac{1}{n}\boldsymbol{\Psi}^\top \boldsymbol{\Psi} \right\| + \frac{\delta}{4n} \right).\end{aligned}$$

Finally, taking the expectation yields:

$$\begin{aligned}\mathbb{E}[\boldsymbol{\beta}^{*\top} \boldsymbol{B}_\delta \boldsymbol{\beta}^*] &\leq \mathbb{E}\left[\left\| \boldsymbol{\Sigma} - \frac{1}{n}\boldsymbol{\Psi}^\top \boldsymbol{\Psi} \right\| \right] \|\boldsymbol{\beta}^*\|^2 + \frac{\delta}{4n} \|\boldsymbol{\beta}^*\|^2 \\ &\leq C \|\boldsymbol{\Sigma}\| \|\boldsymbol{\beta}^*\|^2 \max\left(\sqrt{\frac{r_0(\boldsymbol{\Sigma})}{n}}, \frac{r_0(\boldsymbol{\Sigma})}{n} \right) + \frac{\delta}{4n} \|\boldsymbol{\beta}^*\|^2, \qquad (48)\end{aligned}$$

where the last inequality follows from the covariance concentration bound of Theorem 4 of Koltchinskii and Lounici [31].

E.3.2 Upper bound of the variance term

In this section, we next derive an upper bound for the expectation of $\boldsymbol{C}_\delta$. Let us first briefly sketch the proof. We consider an event A_λ whose probability is at most $c_1 e^{-nm/c_2}$ for some universal constants c_1 and c_2, and use the law of total expectation to write

$$\mathbb{E}[\mathrm{tr}(\boldsymbol{C}_\delta)] = \mathbb{E}[\mathrm{tr}(\boldsymbol{C}_\delta) \mid A_\lambda^{\mathsf{c}}] \, \mathbb{P}(A_\lambda^{\mathsf{c}}) + \mathbb{E}[\mathrm{tr}(\boldsymbol{C}_\delta) \mid A_\lambda] \, \mathbb{P}(A_\lambda). \qquad (49)$$

We then demonstrate that, conditional on the high-probability event A_λ^{c}, the expectation of the trace vanishes as $n \to \infty$, thereby controlling the first term on the right-hand side of (49). For the second term, we show that the conditional expectation $\mathbb{E}[\mathrm{tr}(\boldsymbol{C}_\delta) \mid A_\lambda]$ grows at most polynomially in n. Since $\mathbb{P}(A_\lambda)$ decays exponentially with respect to n, it follows that the second term in the right hand side of (49) also tends to zero as $n \to \infty$.

To this end, we should first define a suitable event A_λ and derive the upper bound for $\mathrm{tr}(\boldsymbol{C}_\delta)$ conditioned on the complement of that event. This is done in Lemma E.5, by utilizing the following technical result.

Corollary E.4. *Consider the setting of Lemma 5 in Bartlett et al. [8] with $\delta > 0$. If*

$$\hat{r}_k(\boldsymbol{\Sigma}) \coloneqq \frac{m\delta + \sum_{i>k} \lambda_i}{\lambda_{k+1}} \geq bmn,$$

then the following two inequalities hold:

(i) $\mu_{k+1}(\boldsymbol{A}_{-i} + \delta \boldsymbol{I}) \leq \dfrac{c(1+b)}{bm} \lambda_{k+1} \hat{r}_k(\boldsymbol{\Sigma})$,

(ii) $\mu_{mn}(\boldsymbol{A}_{-i} + \delta \boldsymbol{I}) \geq \dfrac{1}{m}\left(\dfrac{1}{c} - \dfrac{c}{b}\right)\lambda_{k+1}\hat{r}_k(\boldsymbol{\Sigma})$.

Proof. From part 1 of Lemma 5 in [8], we have

$$\begin{aligned}
\mu_{k+1}(\boldsymbol{A}_{-i} + \delta \boldsymbol{I}) &= \mu_{k+1}(\boldsymbol{A}_{-i}) + \delta \\
&\leq c\left(\frac{1}{m}\sum_{j>k}\lambda_j + \delta + \lambda_{k+1}n\right) \\
&\leq \frac{c}{m}\left(\sum_{j>k}\lambda_j + m\delta + \lambda_{k+1}nm\right) \\
&\leq \frac{c}{m}\left(1 + \frac{1}{b}\right)\lambda_{k+1}\hat{r}_k(\boldsymbol{\Sigma})
\end{aligned}$$

where the last line follows from our assumption that $\hat{r}_k(\boldsymbol{\Sigma}) \geq bn$. Similarly, invoking part 2) of Lemma 5 in [8], we obtain

$$\begin{aligned}
\mu_n(\boldsymbol{A}_{-i} + \delta \boldsymbol{I}) &= \mu_n(\boldsymbol{A}_{-i}) + \delta \\
&\geq \frac{1}{cm}\sum_{j>k}\lambda_j - c\lambda_{k+1}n + \delta \\
&\geq \frac{1}{cm}\left(\sum_{j>k}\lambda_j + m\delta\right) - c\lambda_{k+1}\cdot\frac{\hat{r}_k(\boldsymbol{\Sigma})}{bm} \\
&\geq \frac{1}{m}\left(\frac{1}{c} - \frac{c}{b}\right)\lambda_{k+1}\hat{r}_k(\boldsymbol{\Sigma}).
\end{aligned}$$

where the third line again follows from that $\hat{r}_k(\boldsymbol{\Sigma}) \geq bnm$. This completes the proof. □

Lemma E.5. *There are constants $b, c \geq 1$ such that, if $0 \leq l \leq k \leq nm/c$, $\hat{r}_k(\boldsymbol{\Sigma}) \geq bnm$, and $\delta > 0$, then with probability at least $1 - 7e^{-nm/c}$,*

$$\operatorname{tr}(\boldsymbol{C}_\delta) \leq c\left(\frac{l}{n} + \frac{nm^2\sum_{i>l}\lambda_i^2}{(m\delta + \sum_{i>k}\lambda_i)^2}\right).$$

Proof of Lemma E.5. For $\boldsymbol{\Sigma} = \sum_{j=1}^p \lambda_j v_j v_j^\top$, where $\lambda_1 \geq \cdots \geq \lambda_p \geq 0$ are the eigenvalues of $\boldsymbol{\Sigma}$ and $v_1, \ldots, v_p$ form an orthonormal basis of $\mathbb{R}^p$, let $z_i \coloneqq \frac{\boldsymbol{\Psi} v_i}{\sqrt{\lambda_i}}$ for each i such that $\lambda_i > 0$. If $\lambda_i = 0$, then $\boldsymbol{\Psi} v_i = \mathbf{0}$ almost surely; in this case, we set $z_i \coloneqq \mathbf{0}$. Then, we have

- $\boldsymbol{A} \coloneqq \boldsymbol{\Psi}\boldsymbol{\Psi}^\top = \boldsymbol{\Psi}\boldsymbol{I}\boldsymbol{\Psi}^\top = \boldsymbol{\Psi}\sum_j v_j v_j^\top \boldsymbol{\Psi}^\top = \sum_j (\boldsymbol{\Psi} v_j)(\boldsymbol{\Psi} v_j)^\top = \sum_j \lambda_j z_j z_j^\top$,
- $\boldsymbol{\Psi}\boldsymbol{\Sigma}\boldsymbol{\Psi}^\top = \boldsymbol{\Psi}\sum_j \lambda_j v_j v_j^\top \boldsymbol{\Psi}^\top = \sum_j \lambda_j (\boldsymbol{\Psi} v_j)(\boldsymbol{\Psi} v_j)^\top = \sum_j \lambda_j^2 z_j z_j^\top$

and this implies that

$$\begin{aligned}
\operatorname{tr}(\boldsymbol{C}_\delta) &= \operatorname{tr}\left((\boldsymbol{\Psi}\boldsymbol{\Psi}^\top + \delta \boldsymbol{I})^{-2}\boldsymbol{\Psi}\boldsymbol{\Sigma}\boldsymbol{\Psi}^\top\right) \\
&= \operatorname{tr}\left((\boldsymbol{A} + \delta \boldsymbol{I})^{-2}\sum_j \lambda_j^2 z_j z_j^\top\right) \\
&= \sum_j \operatorname{tr}\left(\lambda_j^2 z_j^\top (\boldsymbol{A} + \delta \boldsymbol{I})^{-2} z_j\right) \\
&= \sum_{j\le l} \lambda_j^2 z_j^\top (\boldsymbol{A} + \delta \boldsymbol{I})^{-2} z_j + \sum_{j>l} \lambda_j^2 z_j^\top (\boldsymbol{A} + \delta \boldsymbol{I})^{-2} z_j \\
&= \underbrace{\sum_{j\le l} \frac{\lambda_j^2 z_j^\top (\boldsymbol{A}_{-j} + \delta \boldsymbol{I})^{-2} z_j}{(1 + \lambda_j z_j^\top (\boldsymbol{A}_{-j} + \delta \boldsymbol{I})^{-1} z_j)^2}}_{:=L} + \underbrace{\sum_{j>l} \lambda_j^2 z_j^\top (\boldsymbol{A} + \delta \boldsymbol{I})^{-2} z_j}_{:=S}
\end{aligned}$$

where $\boldsymbol{A}_{-j} \coloneqq \sum_{i\neq j} \lambda_i z_i z_i^\top$ and the last equality comes from Lemma S.3 of Bartlett et al. [8] with $k = 1$ and $Z = \sqrt{\lambda_j} z_j$ for $j \le l$.

The remaining proof provides upper bound of each terms L and S, respectively. We begin the analysis of L by adapting the arguments and notation in the proof of Lemma 6 of Bartlett et al. [8]. Let $\mathcal{L}_j$ denote the span of the eigenvectors corresponding to the smallest $nm - k$ eigenvalues of $\boldsymbol{A}_{-j}$; this subspace is independent of z_j. Substituting the eigenvalue bounds derived in Corollary E.4 yields the following inequalities with probability at least $1 - 2e^{-nm/c_1}$:

$$z^\top (\boldsymbol{A}_{-j} + \delta \boldsymbol{I})^{-2} z \le \frac{c_1^2 m^2}{\lambda_{k+1}^2 \hat{r}_k(\boldsymbol{\Sigma})^2} \left\|z\right\|^2,$$

and the upper bounds on $\mu_{k+1}(\boldsymbol{A}_{-j} + \delta \boldsymbol{I})$ yield

$$z^\top (\boldsymbol{A}_{-j} + \delta \boldsymbol{I})^{-1} z \ge (\Pi_{L_j} z)^\top (\boldsymbol{A}_{-j} + \delta \boldsymbol{I})^{-1} \Pi_{L_j} z \ge \frac{m \left\|\Pi_{L_j} z\right\|^2}{c_1 \lambda_{k+1} \hat{r}_k(\boldsymbol{\Sigma})}.$$

Then, for $l \le k$, we have

$$\begin{aligned}
L &\le \sum_{j\le l} \frac{z_j^\top (\boldsymbol{A}_{-j} + \delta \boldsymbol{I})^{-2} z_j}{(z_j^\top (\boldsymbol{A}_{-j} + \delta \boldsymbol{I})^{-1} z_j)^2} \\
&\le c_1^4 \sum_{j\le l} \frac{\left\|z_j\right\|^2}{\left\|\Pi_{L_j} z_j\right\|^4}.
\end{aligned}$$

Analogously to Bartlett et al. [8], we apply Corollary 1 in Bartlett et al. [8] l times combined with a union bound to show that, with probability at least $1 - 3e^{-t}$,

$$\begin{aligned}
\left\|\boldsymbol{z}_i\right\|^2 &\le n + \frac{a\sigma_x^2}{m}\left(t + \ln k + \sqrt{nm(t + \ln k)}\right) \le c_2 n, \\
\left\|\Pi_{L_i} \boldsymbol{z}_i\right\|^2 &\ge n - \frac{a\sigma_x^2}{m}\left(k + t + \ln k + \sqrt{nm(t + \ln k)}\right) \ge \frac{n}{c_3},
\end{aligned}$$

hold for all $1 \le i \le l$. This holds provided $t < nm/c_0$ and $c > c_0$ for sufficiently large c_0. We note that c_2 and c_3 are determined solely by c_0, a, and σ_x, independent of the choice of sufficiently large c.

Combining these results, with probability at least $1 - 5e^{-nm/c_0}$, we have

$$L \le c_1^4 \sum_{j\le l} \frac{\left\|z_j\right\|^2}{\left\|\Pi_{L_j} z_j\right\|^4} \le c_4 \frac{l}{n}$$

Next, we analyze $S = \sum_{j>l} \lambda_j^2 z_j^\top (\boldsymbol{A} + \delta \boldsymbol{I})^{-2} z_j$. Bartlett et al. [8, Lemma 5-2] By Bartlett et al. [8, Lemma 5-2] and the argument used in the proof of Corollary E.4, we have, for sufficiently large c_1,

$$\mu_{nm}(\boldsymbol{A} + \delta \boldsymbol{I}) \ge \frac{1}{m}\left(\frac{1}{c} - \frac{c}{b}\right) \lambda_{k+1} \hat{r}_k(\boldsymbol{\Sigma}) \ge \frac{1}{c_1 m} \lambda_{k+1} \hat{r}_k(\boldsymbol{\Sigma}).$$

Therefore, we have

$$S = \sum_{j>l} \lambda_j^2 z_j^\top (\boldsymbol{A} + \delta \boldsymbol{I})^{-2} z_j \leq \frac{c_1^2 m^2 \sum_{j>l} \lambda_j^2 \left\| z_j \right\|^2}{(\lambda_{k+1} \hat{r}_k(\boldsymbol{\Sigma}))^2} \leq \frac{c_5 n m^2 \sum_{j>l} \lambda_j^2}{(\lambda_{k+1} \hat{r}_k(\boldsymbol{\Sigma}))^2}.$$

where the second inequality holds, by Lemma 7 in Bartlett et al. [8], with probability $1 - 2e^{-nm/c_0}$. This completes the proof. □

Lemma E.6. *For any $\delta > 0$, it holds that*

$$\operatorname{tr}(\boldsymbol{C}_\delta) \leq \frac{nm \left\| \boldsymbol{\Sigma} \right\|}{\delta}.$$

Proof. Recall the definition $\boldsymbol{C}_\delta = (\boldsymbol{\Psi}\boldsymbol{\Psi}^\top + \delta \boldsymbol{I})^{-1} \boldsymbol{\Psi}\boldsymbol{\Sigma}\boldsymbol{\Psi}^\top (\boldsymbol{\Psi}\boldsymbol{\Psi}^\top + \delta \boldsymbol{I})^{-1}$. Since $\boldsymbol{\Sigma} \preceq \left\| \boldsymbol{\Sigma} \right\| \boldsymbol{I}$, we have

$$\begin{aligned}
\operatorname{tr}(\boldsymbol{C}_\delta) &\leq \left\| \boldsymbol{\Sigma} \right\| \operatorname{tr}\left((\boldsymbol{\Psi}\boldsymbol{\Psi}^\top + \delta \boldsymbol{I})^{-2} \boldsymbol{\Psi}\boldsymbol{\Psi}^\top \right) \\
&= \left\| \boldsymbol{\Sigma} \right\| \operatorname{tr}\left((\boldsymbol{\Psi}\boldsymbol{\Psi}^\top + \delta \boldsymbol{I})^{-2} [(\boldsymbol{\Psi}\boldsymbol{\Psi}^\top + \delta \boldsymbol{I}) - \delta \boldsymbol{I}] \right) \\
&= \left\| \boldsymbol{\Sigma} \right\| \operatorname{tr}\left((\boldsymbol{\Psi}\boldsymbol{\Psi}^\top + \delta \boldsymbol{I})^{-1} - \delta (\boldsymbol{\Psi}\boldsymbol{\Psi}^\top + \delta \boldsymbol{I})^{-2} \right).
\end{aligned}$$

Observing that $\delta(\boldsymbol{\Psi}\boldsymbol{\Psi}^\top + \delta \boldsymbol{I})^{-2}$ is positive semi-definite, we can drop the negative term:

$$\operatorname{tr}(\boldsymbol{C}_\delta) \leq \left\| \boldsymbol{\Sigma} \right\| \operatorname{tr}\left((\boldsymbol{\Psi}\boldsymbol{\Psi}^\top + \delta \boldsymbol{I})^{-1} \right).$$

Since the eigenvalues of $(\boldsymbol{\Psi}\boldsymbol{\Psi}^\top + \delta \boldsymbol{I})^{-1}$ are bounded by $1/\delta$, the trace of $(\boldsymbol{\Psi}\boldsymbol{\Psi}^\top + \delta \boldsymbol{I})^{-1}$ is bounded by nm/δ. The conclusion thus follows from the inequality above. □

We now combine this bound with the high-probability bound derived in Lemma E.5. Let A_λ denote the event where the bound in Lemma E.5 holds, such that $\mathbb{P}(A_\lambda^\mathsf{c}) \leq 7e^{-nm/c_0}$. By the law of total expectation,

$$\begin{aligned}
\mathbb{E}[\operatorname{tr}(\boldsymbol{C}_\delta)] &\leq \mathbb{E}[\operatorname{tr}(\boldsymbol{C}_\delta) \mid A_\lambda^\mathsf{c}] \, \mathbb{P}(A_\lambda^\mathsf{c}) + \mathbb{E}[\operatorname{tr}(\boldsymbol{C}_\delta) \mid A_\lambda] \, \mathbb{P}(A_\lambda) \\
&\leq c \left(\frac{l}{n} + \frac{nm^2 \sum_{i>l} \lambda_i^2}{(m\delta + \sum_{i>k} \lambda_i)^2} \right) + \frac{nm \left\| \boldsymbol{\Sigma} \right\|}{\delta} \cdot 7e^{-nm/c_0}. \qquad (50)
\end{aligned}$$

E.3.3 Benign condition for ridge regression

The effective rank condition in Lemma E.5 is $\hat{r}_k(\boldsymbol{\Sigma}) \geq bnm$, which is satisfied whenever $r_k(\boldsymbol{\Sigma}) \geq bn$. For fixed $\delta > 0$, define

$$\hat{R}_k(\boldsymbol{\Sigma}) \coloneqq \frac{\left(m\delta + \sum_{i>k} \lambda_i \right)^2}{\sum_{i>k} \lambda_i^2}.$$

We retain the original definition of $R_k(\boldsymbol{\Sigma})$; the quantity $\hat{R}_k(\boldsymbol{\Sigma})$ includes the ridge parameter in its numerator. We show below that the original benign condition introduced in Definition 4 of Bartlett et al. [8] is sufficient for the regularized variance bound to converge to zero.

We first establish the following auxiliary result.

Lemma E.7. *Let $\{\lambda_i\}_{i \geq 1}$ be the eigenvalues of $\boldsymbol{\Sigma}$, sorted in non-increasing order. Given constants $b > 1$, $n \in \mathbb{N}$, and $\delta > 0$, let*

$$k^* \coloneqq \min \left\{ k \geq 0 : \frac{m\delta + \sum_{i>k} \lambda_i}{\lambda_{k+1}} \geq bnm \right\}.$$

Then, a function $f : \{0, 1, \dots\} \to \mathbb{R}$ defined as

$$f(l) \coloneqq \frac{l}{bnm} + \frac{bnm \sum_{i>l} \lambda_i^2}{\left(m\delta + \sum_{i>k^*} \lambda_i \right)^2}$$

is minimized at $l = k^*$.

Proof. For convenience, let us define

$$C \coloneqq \frac{bnm}{(m\delta + \sum_{i>k^*} \lambda_i)^2} \quad \text{and} \quad \tau \coloneqq \frac{m\delta + \sum_{i>k^*} \lambda_i}{bnm}.$$

Considering the discrete difference $\Delta f(l) \coloneqq f(l) - f(l-1)$, we have

$$\Delta f(l) = \frac{1}{bnm} - C\lambda_l^2 = C\left(\tau^2 - \lambda_l^2\right).$$

Since λ_i is non-increasing, the sign of $\Delta f(l)$ depends on the magnitude of λ_l relative to the threshold τ. In particular, there are two cases:

- If $l \leq k^*$: By the definition of k^*, $\frac{m\delta + \sum_{i>k^*-1} \lambda_i}{\lambda_{k^*}} < bnm$. Rearranging this yields $(bnm-1)\lambda_{k^*} > m\delta + \sum_{i>k^*} \lambda_i$. Since $bnm > 1$, we have

  $$\lambda_l \geq \lambda_{k^*} > \frac{m\delta + \sum_{i>k^*} \lambda_i}{bnm - 1} > \frac{m\delta + \sum_{i>k^*} \lambda_i}{bnm} = \tau.$$

  Thus, $\Delta f(l) < 0$, meaning $f(l)$ is strictly decreasing for $l \leq k^*$.

- If $l > k^*$: By the definition of k^*, we have $\frac{m\delta + \sum_{i>k^*} \lambda_i}{\lambda_{k^*+1}} \geq bnm$, which implies $\lambda_{k^*+1} \leq \tau$. Consequently, for any $l > k^*$,

  $$\lambda_l \leq \lambda_{k^*+1} \leq \tau.$$

  Thus, $\Delta f(l) \geq 0$, meaning $f(l)$ is non-decreasing for $l > k^*$.

Therefore, $f(l)$ has its global minimum at $l = k^*$. □

Lemma E.8. *Fix $b > 1$ and $\delta > 0$, and define $k_n^* \coloneqq \min\{k \geq 0 : r_k(\mathbf{\Sigma}_n) \geq bnm\}$ and $\hat{k}_n^* \coloneqq \min\{k \geq 0 : \hat{r}_k(\mathbf{\Sigma}_n) \geq bnm\}$. Any covariance operator sequence $\mathbf{\Sigma}_n$ satisfying the benign condition*

$$\lim_{n\to\infty} \frac{r_0(\mathbf{\Sigma}_n)}{nm} = \lim_{n\to\infty} \frac{k_n^*}{nm} = \lim_{n\to\infty} \frac{nm}{R_{k_n^*}(\mathbf{\Sigma}_n)} = 0$$

also satisfies

$$\lim_{n\to\infty} \frac{r_0(\mathbf{\Sigma}_n)}{nm} = \lim_{n\to\infty} \frac{\hat{k}_n^*}{nm} = \lim_{n\to\infty} \frac{nm}{\hat{R}_{\hat{k}_n^*}(\mathbf{\Sigma}_n)} = 0.$$

Proof. Fix n and write $k^* = k_n^*$ and $\hat{k}^* = \hat{k}_n^*$. Let $\{\lambda_i\}$ be the eigenvalues of $\mathbf{\Sigma}_n$. Since $\hat{r}_k(\mathbf{\Sigma}_n) \geq r_k(\mathbf{\Sigma}_n)$, we have $\hat{k}^* \leq k^*$, and hence

$$\sum_{i>\hat{k}^*} \lambda_i \geq \sum_{i>k^*} \lambda_i.$$

Define

$$\tilde{f}(l) \coloneqq \frac{l}{bnm} + \frac{bnm \sum_{i>l} \lambda_i^2}{\left(m\delta + \sum_{i>\hat{k}^*} \lambda_i\right)^2}.$$

By Lemma E.7 with the term $\hat{k}^*$, we have $\tilde{f}(\hat{k}^*) \leq \tilde{f}(k^*)$. Therefore,

$$\begin{aligned}
\frac{k^*}{bnm} + \frac{bnm}{R_{k^*}(\mathbf{\Sigma}_n)} &= \frac{k^*}{bnm} + \frac{bnm \sum_{i>k^*} \lambda_i^2}{\left(\sum_{i>k^*} \lambda_i\right)^2} \\
&\geq \frac{k^*}{bnm} + \frac{bnm \sum_{i>k^*} \lambda_i^2}{\left(m\delta + \sum_{i>k^*} \lambda_i\right)^2} \\
&\geq \tilde{f}(k^*) \\
&\geq \tilde{f}(\hat{k}^*) \\
&= \frac{\hat{k}^*}{bnm} + \frac{bnm}{\hat{R}_{\hat{k}^*}(\mathbf{\Sigma}_n)}.
\end{aligned}$$

The first expression converges to zero under the benign condition. Since $b > 1$ is fixed and all terms are nonnegative,

$$\frac{\hat{k}_n^*}{nm} \longrightarrow 0, \qquad \frac{nm}{\hat{R}_{\hat{k}_n^*}(\boldsymbol{\Sigma}_n)} \longrightarrow 0.$$

The assertion then follows. □

Based on the established results, we are now ready to prove the Theorem 5.3.

Theorem E.9 (Theorem 5.3)**.** *Under Assumptions 4.1 and 5.1, if $\boldsymbol{\Sigma}$ has benign eigenvalues, then for any $\delta > 0$, the expected risk satisfies $\mathbb{E}[\mathcal{R}(\hat{\boldsymbol{\beta}}(\delta))] \to 0$ as $n \to \infty$.*

Proof of Theorem 5.3. Combining the bounds derived in (48) and (50), for any $\delta > 0$, one can obtain an upper bound on the expected risk as

$$\begin{aligned}\mathbb{E}[\mathcal{R}(\hat{\boldsymbol{\beta}})] &\leq 2\,\mathbb{E}[\boldsymbol{\beta}^{*\top}\boldsymbol{B}\boldsymbol{\beta}^*] + 2\,\mathbb{E}[\boldsymbol{\epsilon}^\top \boldsymbol{C}_\delta \boldsymbol{\epsilon}] \\ &\leq C\,\|\boldsymbol{\Sigma}\|\,\|\boldsymbol{\beta}^*\|^2 \max\left(\sqrt{\frac{r_0(\boldsymbol{\Sigma})}{nm}}, \frac{r_0(\boldsymbol{\Sigma})}{nm}\right) + \frac{\delta}{2n}\|\boldsymbol{\beta}^*\|^2 \\ &\quad + c\left(\frac{l}{nm} + \frac{nm\sum_{i>l}\lambda_i^2}{(m\delta + \sum_{i>k}\lambda_i)^2} + \frac{n\,\|\boldsymbol{\Sigma}\|}{\delta} e^{-nm/c_0}\right).\end{aligned}$$

Under the condition that $\boldsymbol{\Sigma}$ has benign eigenvalues, Lemma E.8 implies that the risk converges to zero; that is, $\mathbb{E}[\mathcal{R}(\hat{\boldsymbol{\beta}})] \to 0$, as $n \to \infty$. □

F Missing details for Section 5.3

F.1 Population risk and the Wasserstein distance

For two probability measures ξ and ν defined on the same Euclidean space $\mathbb{R}^m$, let $\Gamma(\xi, \nu)$ denote the collection of all *couplings* of ξ and ν; that is, probability measures on $\mathbb{R}^m \times \mathbb{R}^m$ whose marginal distributions on the first and second coordinates are ξ and ν, respectively. Recall that the 2-Wasserstein distance $\mathcal{W}_2(\xi, \nu)$ is defined as the square root of

$$\min_{\gamma \in \Gamma(\xi,\nu)} \int \|\boldsymbol{z} - \boldsymbol{x}\|^2 \,\mathrm{d}\gamma(\boldsymbol{z}, \boldsymbol{x}). \tag{51}$$

The 2-Wasserstein distance is indeed a metric, on the space of probability distributions with finite second moments. For further details, see, *e.g.*, [54, Chapter 5]. When ξ is absolutely continuous with respect to the Lebesgue measure, it is well known that (51) and the problem (8) with a quadratic cost $c(\boldsymbol{z}, \boldsymbol{x}) = \|\boldsymbol{z} - \boldsymbol{x}\|^2$ have the same optimal value [50].

Proposition F.1. *For the population risk $\mathcal{R}(\hat{\boldsymbol{\beta}})$ defined in* (14)*, it holds that*

$$\mathcal{W}_2^2(\hat{G}_\sharp \xi, \nu) \leq \mathcal{R}(\hat{\boldsymbol{\beta}}).$$

Proof. Notice that the population risk can be written as

$$\begin{aligned}\mathcal{R}(\hat{\boldsymbol{\beta}}) &= \mathbb{E}_{\boldsymbol{z}\sim\xi}\left[\left\|\hat{G}(\boldsymbol{z}) - G^*(\boldsymbol{z})\right\|^2\right] \\ &= \int \left\|\hat{G}(\boldsymbol{z}) - G^*(\boldsymbol{z})\right\|^2 \mathrm{d}\xi(\boldsymbol{z}) \\ &= \int \|\boldsymbol{x} - \boldsymbol{y}\|^2 \,\mathrm{d}\gamma(\boldsymbol{x}, \boldsymbol{y})\end{aligned}$$

where $\gamma \coloneqq (\hat{G}, G^*)_\sharp \xi$. As G^* is an OT map from ξ to ν, we know that γ is a joint distribution of $\hat{G}_\sharp \xi$ and ν. The inequality is now immediate from the definition of the 2-Wasserstein distance (51). □

F.2 Incorporating the noise component into the eigenlearning framework

The eigenlearning framework [56] and the random feature eigenframework [57], in which the labels are assumed to be scalar-valued, model the labels as $x_i = f(\boldsymbol{z}_i) + \epsilon_i$, where the label noises ϵ_i are i.i.d. with $\mathbb{E}[\epsilon_i] = 0$, $\mathrm{Var}(\epsilon_i) = \sigma^2$, and are independent of the inputs. In those frameworks, the effect of label noise is incorporated by representing it as additional target mass placed in modes with zero eigenvalue of the kernel operator, *i.e.*, as a component of L^2 orthogonal to the eigenfunctions ϕ_i with nonzero eigenvalues; see, *e.g.*, Simon et al. [56, Appendix I.3]. Such components are said to be of learnability zero, in the sense that the kernel does not fit them. Under the i.i.d. and homoscedastic noise assumption, the contribution of this "unlearnable" component to the test risk reduces to an additive correction term of the scalar noise variance σ^2.

In our setting, the residuals $\boldsymbol{\epsilon}_i$ arising from the optimal transport pairing are generally not independent, and they may further exhibit dependence on the inputs. Therefore, while desired, it is unclear whether there is a way to directly translate to our setting the interpretation of the scalar σ^2 from the eigenlearning framework as the "noise level". If this were possible, then a natural candidate for the noise level in our setting would be the marginal mean squared norm $\mathbb{E}\left\|\boldsymbol{\epsilon}_i\right\|^2$. In the remainder of this section, we argue that, under assumptions analogous in spirit to those used by Simon et al. [56, 57], $\mathbb{E}\left\|\boldsymbol{\epsilon}_i\right\|^2$ is in fact a theoretically justified choice for the noise-level correction term for our purposes.

To make contact with the risk formulae, we now adopt the interpretation of Assumption 5.1 described in Section E.2. Under this interpretation, let us begin by examining how the noise term arises in the eigenlearning framework. Recall (22), which tells us how $\hat{\boldsymbol{v}}$ is determined in the noiseless setting. The rightmost factors in the right hand side of (22) being $\boldsymbol{\Phi}^\top \boldsymbol{v}$ is because in the noiseless setting the observed labels are $\boldsymbol{f}(\boldsymbol{z}_j) = \sum_i v_i \boldsymbol{\phi}_i(\boldsymbol{z}_j)$. Thus, when the observations are noisy, we should replace $\boldsymbol{\Phi}^\top \boldsymbol{v}$ with $\boldsymbol{\Phi}^\top \boldsymbol{v} + \mathfrak{e}$, where $\mathfrak{e}$ is a concatenation of $\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_n$ into an nm-dimensional vector. This yields a new formula for $\hat{\boldsymbol{v}}$, namely

$$\hat{\boldsymbol{v}} = \boldsymbol{\Lambda}\boldsymbol{\Phi}(\boldsymbol{\Phi}^\top \boldsymbol{\Lambda}\boldsymbol{\Phi} + \delta \boldsymbol{I})^{-1}\boldsymbol{\Phi}^\top \boldsymbol{v} + \boldsymbol{\Lambda}\boldsymbol{\Phi}(\boldsymbol{\Phi}^\top \boldsymbol{\Lambda}\boldsymbol{\Phi} + \delta \boldsymbol{I})^{-1}\mathfrak{e}. \tag{52}$$

The mean squared error of $\hat{\boldsymbol{f}} = \sum_i \hat{v}_i \boldsymbol{\phi}_i$ compared to the ground truth function $\boldsymbol{f}$ is $\left\|\hat{\boldsymbol{v}} - \boldsymbol{v}\right\|^2$. Taking a conditional expectation and applying a bias–variance type decomposition, we get

$$\begin{aligned}
\mathbb{E}\left[\left\|\hat{\boldsymbol{v}} - \boldsymbol{v}\right\|^2 \middle| \boldsymbol{\Phi}\right] &= \mathbb{E}\left[\left\|(\hat{\boldsymbol{v}} - \mathbb{E}[\hat{\boldsymbol{v}}|\boldsymbol{\Phi}]) + (\mathbb{E}[\hat{\boldsymbol{v}}|\boldsymbol{\Phi}] - \boldsymbol{v})\right\|^2 \middle| \boldsymbol{\Phi}\right] \\
&= \mathbb{E}\left[\left\|\hat{\boldsymbol{v}} - \mathbb{E}[\hat{\boldsymbol{v}}|\boldsymbol{\Phi}]\right\|^2 \middle| \boldsymbol{\Phi}\right] + \mathbb{E}\left[\left\|\mathbb{E}[\hat{\boldsymbol{v}}|\boldsymbol{\Phi}] - \boldsymbol{v}\right\|^2 \middle| \boldsymbol{\Phi}\right] + 2\,\mathbb{E}\left[\left\langle \hat{\boldsymbol{v}} - \mathbb{E}[\hat{\boldsymbol{v}}|\boldsymbol{\Phi}], \mathbb{E}[\hat{\boldsymbol{v}}|\boldsymbol{\Phi}] - \boldsymbol{v}\right\rangle \middle| \boldsymbol{\Phi}\right] \\
&= \mathbb{E}\left[\left\|\hat{\boldsymbol{v}} - \mathbb{E}[\hat{\boldsymbol{v}}|\boldsymbol{\Phi}]\right\|^2 \middle| \boldsymbol{\Phi}\right] + \mathbb{E}\left[\left\|\mathbb{E}[\hat{\boldsymbol{v}}|\boldsymbol{\Phi}] - \boldsymbol{v}\right\|^2 \middle| \boldsymbol{\Phi}\right]
\end{aligned}$$

where in the last line we used the fact that $\mathbb{E}[\hat{\boldsymbol{v}}|\boldsymbol{\Phi}] - \boldsymbol{v}$ is a constant when conditioned on $\boldsymbol{\Phi}$. By the law of total expectation, we end up with

$$\mathbb{E}\left[\left\|\hat{\boldsymbol{v}} - \boldsymbol{v}\right\|^2\right] = \mathbb{E}\left[\left\|\hat{\boldsymbol{v}} - \mathbb{E}[\hat{\boldsymbol{v}}|\boldsymbol{\Phi}]\right\|^2\right] + \mathbb{E}\left[\left\|\mathbb{E}[\hat{\boldsymbol{v}}|\boldsymbol{\Phi}] - \boldsymbol{v}\right\|^2\right]. \tag{53}$$

Proposition D.3 asserts that $\mathbb{E}[\mathfrak{e}] = \boldsymbol{0}$. Thus, as we consider the idealized surrogate model discussed above, we have

$$\mathbb{E}[\hat{\boldsymbol{v}}|\boldsymbol{\Phi}] = \boldsymbol{\Lambda}\boldsymbol{\Phi}(\boldsymbol{\Phi}^\top \boldsymbol{\Lambda}\boldsymbol{\Phi} + \delta \boldsymbol{I})^{-1}\boldsymbol{\Phi}^\top \boldsymbol{v}.$$

Notice that the right hand side of the above is exactly the right hand side of (52) when there is no noise; *i.e.*, $\mathfrak{e} = \boldsymbol{0}$. Hence, the risk estimate established in Observation C.3 in the noiseless setting directly applies, allowing us to put

$$\mathbb{E}\left[\left\|\mathbb{E}[\hat{\boldsymbol{v}}|\boldsymbol{\Phi}] - \boldsymbol{v}\right\|^2\right] \approx \frac{1}{1 - \frac{q}{nm}} \left(\sum_i \frac{\kappa^2 v_i^2}{(\lambda_i + \kappa)^2}\right) \tag{54}$$

for q and κ defined therein.

It remains to estimate

$$\begin{aligned}
\mathbb{E}\left[\left\|\hat{\boldsymbol{v}} - \mathbb{E}[\hat{\boldsymbol{v}}|\boldsymbol{\Phi}]\right\|^2\right] &= \mathbb{E}\left[\left\|\boldsymbol{\Lambda}\boldsymbol{\Phi}(\boldsymbol{\Phi}^\top \boldsymbol{\Lambda}\boldsymbol{\Phi} + \delta \boldsymbol{I})^{-1}\mathfrak{e}\right\|^2\right] \\
&= \mathbb{E}\left[\mathfrak{e}^\top (\boldsymbol{\Phi}^\top \boldsymbol{\Lambda}\boldsymbol{\Phi} + \delta \boldsymbol{I})^{-1}\boldsymbol{\Phi}^\top \boldsymbol{\Lambda}^2 \boldsymbol{\Phi}(\boldsymbol{\Phi}^\top \boldsymbol{\Lambda}\boldsymbol{\Phi} + \delta \boldsymbol{I})^{-1}\mathfrak{e}\right] \\
&= \mathbb{E}\left[\mathrm{tr}\left((\boldsymbol{\Phi}^\top \boldsymbol{\Lambda}\boldsymbol{\Phi} + \delta \boldsymbol{I})^{-1}\boldsymbol{\Phi}^\top \boldsymbol{\Lambda}^2 \boldsymbol{\Phi}(\boldsymbol{\Phi}^\top \boldsymbol{\Lambda}\boldsymbol{\Phi} + \delta \boldsymbol{I})^{-1}\mathfrak{e}\mathfrak{e}^\top\right)\right]
\end{aligned} \tag{55}$$

where in the last line we used the cyclic property of the trace function. For convenience, let us define $\boldsymbol{S}(\boldsymbol{\Phi}) \coloneqq \boldsymbol{\Lambda}\boldsymbol{\Phi}(\boldsymbol{\Phi}^\top\boldsymbol{\Lambda}\boldsymbol{\Phi} + \delta\boldsymbol{I})^{-1}$, so that the expectation in the last line of (55) can be compactly written as $\mathbb{E}[\mathrm{tr}(\boldsymbol{S}(\boldsymbol{\Phi})^\top\boldsymbol{S}(\boldsymbol{\Phi})\mathfrak{e}\mathfrak{e}^\top)]$.

Let $\boldsymbol{Q} \in \mathbb{R}^{nm\times nm}$ be any orthogonal matrix. According to the Gaussian universality ansatz, $\boldsymbol{\Phi}$ and $\boldsymbol{\Phi Q}$ are identically distributed. Now, observe that

$$\begin{aligned}\boldsymbol{S}(\boldsymbol{\Phi Q}) &= \boldsymbol{\Lambda}\boldsymbol{\Phi Q}(\boldsymbol{Q}^\top\boldsymbol{\Phi}^\top\boldsymbol{\Lambda}\boldsymbol{\Phi Q} + \delta\boldsymbol{I})^{-1}\\ &= \boldsymbol{\Lambda}\boldsymbol{\Phi Q}(\boldsymbol{Q}^\top(\boldsymbol{\Phi}^\top\boldsymbol{\Lambda}\boldsymbol{\Phi} + \delta\boldsymbol{I})\boldsymbol{Q})^{-1}\\ &= \boldsymbol{\Lambda}\boldsymbol{\Phi}\boldsymbol{Q}\boldsymbol{Q}^\top(\boldsymbol{\Phi}^\top\boldsymbol{\Lambda}\boldsymbol{\Phi} + \delta\boldsymbol{I})^{-1}\boldsymbol{Q}\\ &= \boldsymbol{S}(\boldsymbol{\Phi})\boldsymbol{Q}.\end{aligned}$$

Thus, $\boldsymbol{S}(\boldsymbol{\Phi})^\top\boldsymbol{S}(\boldsymbol{\Phi})$ and $\boldsymbol{S}(\boldsymbol{\Phi Q})^\top\boldsymbol{S}(\boldsymbol{\Phi Q}) = \boldsymbol{Q}^\top\boldsymbol{S}(\boldsymbol{\Phi})^\top\boldsymbol{S}(\boldsymbol{\Phi})\boldsymbol{Q}$ are also identically distributed. In other words, the distribution of $\boldsymbol{S}(\boldsymbol{\Phi})^\top\boldsymbol{S}(\boldsymbol{\Phi})$ is orthogonally invariant. It is a well known fact[7] that such a matrix is, for $\boldsymbol{U}$ distributed according to the Haar measure on the orthogonal group, equal in distribution to $\boldsymbol{UDU}^\top$ for some random diagonal matrix $\boldsymbol{D} = \mathrm{diag}\{d_1, \ldots, d_{nm}\}$ independent of $\boldsymbol{U}$. From this we can derive a distributional equivalence

$$\begin{aligned}\mathrm{tr}(\boldsymbol{S}(\boldsymbol{\Phi})^\top\boldsymbol{S}(\boldsymbol{\Phi})\mathfrak{e}\mathfrak{e}^\top) \overset{\mathrm{d}}{=} \mathrm{tr}(\boldsymbol{UDU}^\top\mathfrak{e}\mathfrak{e}^\top) &= \mathrm{tr}(\boldsymbol{DU}^\top\mathfrak{e}\mathfrak{e}^\top\boldsymbol{U})\\ &= \left\langle \boldsymbol{D}, \boldsymbol{U}^\top\mathfrak{e}\mathfrak{e}^\top\boldsymbol{U}\right\rangle_F\\ &= \sum_{i=1}^{nm} d_{ii}\boldsymbol{u}_i^\top\mathfrak{e}\mathfrak{e}^\top\boldsymbol{u}_i,\end{aligned}$$

where $\left\langle \cdot, \cdot \right\rangle_F$ denotes the Frobenius inner product and $\boldsymbol{u}_i$ denotes the ith column vector of $\boldsymbol{U}$. As $\boldsymbol{U}$ is Haar-distributed on the orthogonal group, each $\boldsymbol{u}_i$ is uniformly distributed on the unit sphere in $\mathbb{R}^{nm}$. Hence, they satisfy $\mathbb{E}[\boldsymbol{u}_i\boldsymbol{u}_i^\top] = \frac{1}{nm}\boldsymbol{I}$; see, *e.g.*, Proposition 3.3.8 in [63]. It follows that

$$\begin{aligned}\mathbb{E}\left[\mathrm{tr}(\boldsymbol{S}(\boldsymbol{\Phi})^\top\boldsymbol{S}(\boldsymbol{\Phi})\mathfrak{e}\mathfrak{e}^\top)\right] &= \sum_{i=1}^{nm}\mathbb{E}\left[d_{ii}\boldsymbol{u}_i^\top\mathfrak{e}\mathfrak{e}^\top\boldsymbol{u}_i\right]\\ &= \sum_{i=1}^{nm}\mathbb{E}\left[d_{ii}\,\mathrm{tr}(\mathfrak{e}\mathfrak{e}^\top\boldsymbol{u}_i\boldsymbol{u}_i^\top)\right]\\ &= \sum_{i=1}^{nm}\mathbb{E}[d_{ii}]\,\mathrm{tr}\left(\mathbb{E}\left[\mathfrak{e}\mathfrak{e}^\top\right]\mathbb{E}\left[\boldsymbol{u}_i\boldsymbol{u}_i^\top\right]\right)\\ &= \frac{1}{nm}\sum_{i=1}^{nm}\mathbb{E}[d_{ii}]\,\mathrm{tr}\left(\mathbb{E}\left[\mathfrak{e}\mathfrak{e}^\top\right]\right)\\ &= \frac{1}{nm}\,\mathbb{E}\left[\mathrm{tr}\left(\boldsymbol{D}\right)\right]\mathrm{tr}\left(\mathbb{E}\left[\mathfrak{e}\mathfrak{e}^\top\right]\right).\end{aligned} \tag{56}$$

As $\mathrm{tr}(\boldsymbol{D})$ is the sum of the eigenvalues of $\boldsymbol{S}(\boldsymbol{\Phi})^\top\boldsymbol{S}(\boldsymbol{\Phi})$, so it is equal to $\mathrm{tr}(\boldsymbol{S}(\boldsymbol{\Phi})^\top\boldsymbol{S}(\boldsymbol{\Phi}))$.

Now, in order to determine the expected value of $\mathrm{tr}(\boldsymbol{S}(\boldsymbol{\Phi})^\top\boldsymbol{S}(\boldsymbol{\Phi}))$, suppose for the moment that we were doing a kernel regression with the labels being pure white noise, *i.e.*, the true signal function $\boldsymbol{f}$ is an identically zero function and the label noise is independent of the inputs and i.i.d. $\mathcal{N}(0, \boldsymbol{I})$. Then we would have $\boldsymbol{v} = \boldsymbol{0}$ and $\mathbb{E}[\mathfrak{e}\mathfrak{e}^\top] = \boldsymbol{I} \in \mathbb{R}^{nm\times nm}$, so on one hand, from (52) and (56) we obtain

$$\begin{aligned}\mathbb{E}\left[\left\|\hat{\boldsymbol{v}} - \boldsymbol{v}\right\|^2\right] = \mathbb{E}\left[\left\|\hat{\boldsymbol{v}}\right\|^2\right] &= \mathbb{E}\left[\mathrm{tr}(\boldsymbol{S}(\boldsymbol{\Phi})^\top\boldsymbol{S}(\boldsymbol{\Phi})\mathfrak{e}\mathfrak{e}^\top)\right]\\ &= \mathrm{tr}\left(\mathbb{E}\left[\boldsymbol{S}(\boldsymbol{\Phi})^\top\boldsymbol{S}(\boldsymbol{\Phi})\right]\mathbb{E}\left[\mathfrak{e}\mathfrak{e}^\top\right]\right)\\ &= \mathrm{tr}\left(\mathbb{E}\left[\boldsymbol{S}(\boldsymbol{\Phi})^\top\boldsymbol{S}(\boldsymbol{\Phi})\right]\right)\\ &= \mathbb{E}\left[\mathrm{tr}\left(\boldsymbol{D}\right)\right].\end{aligned}$$

[7] See, *e.g.*, Corollary 2.5.4 in [4], where the statement is in terms of a Gaussian orthogonal ensemble but the proof applies to any orthogonally invariant random real symmetric matrices.

On the other hand, one can also get an estimate of $\mathbb{E}\left[\|\hat{\boldsymbol{v}}-\boldsymbol{v}\|^2\right]$ from (37), but with a caveat as we discussed in Remark C.4. Indeed, plugging in $\boldsymbol{v}=\mathbf{0}$ and $\sigma^2=1$ into $\mathcal{E}_{\text{te}}-m\sigma^2$ where $\mathcal{E}_{\text{te}}$ is from (37), we get

$$\mathbb{E}\left[\operatorname{tr}\left(\boldsymbol{D}\right)\right] \approx \frac{m}{1-\frac{q}{nm}} - m \tag{57}$$

where q is the quantity defined in Observation C.3.

Resuming from (56) with the observation (57) in hand, we get

$$\begin{aligned}
\mathbb{E}\left[\operatorname{tr}(\boldsymbol{S}(\boldsymbol{\Phi})^\top \boldsymbol{S}(\boldsymbol{\Phi})\mathfrak{e}\mathfrak{e}^\top)\right] &= \frac{1}{nm}\,\mathbb{E}\left[\operatorname{tr}\left(\boldsymbol{D}\right)\right]\operatorname{tr}\left(\mathbb{E}\left[\mathfrak{e}\mathfrak{e}^\top\right]\right) \\
&\approx \frac{1}{n}\left(\frac{1}{1-\frac{q}{nm}}-1\right)\mathbb{E}\left[\operatorname{tr}\left(\mathfrak{e}\mathfrak{e}^\top\right)\right] \\
&= \frac{1}{n}\left(\frac{1}{1-\frac{q}{nm}}-1\right)\mathbb{E}\left[\mathfrak{e}^\top\mathfrak{e}\right].
\end{aligned}$$

Here, as $\mathfrak{e}$ is a concatenation of $\boldsymbol{\epsilon}_1,\dots,\boldsymbol{\epsilon}_n$ which are all identically distributed, we have

$$\mathbb{E}\left[\mathfrak{e}^\top\mathfrak{e}\right] = \mathbb{E}\left[\|\mathfrak{e}\|^2\right] = \mathbb{E}\left[\sum_{i=1}^n \|\boldsymbol{\epsilon}_i\|^2\right] = n\,\mathbb{E}\left[\|\boldsymbol{\epsilon}_1\|^2\right],$$

and thus we obtain

$$\mathbb{E}\left[\operatorname{tr}(\boldsymbol{S}(\boldsymbol{\Phi})^\top \boldsymbol{S}(\boldsymbol{\Phi})\mathfrak{e}\mathfrak{e}^\top)\right] \approx \left(\frac{1}{1-\frac{q}{nm}}-1\right)\mathbb{E}\left[\|\boldsymbol{\epsilon}_1\|^2\right]. \tag{58}$$

Substituting the results from (54) and (58) back to (53), we obtain the risk estimate, as follows.

Observation F.2. *Assume that the multivariate Gaussian universality ansatz (Assumption 5.1) holds. Consider a regression model as in* (10)*, and suppose we perform a vector-valued kernel ridge regression to obtain a regressor $\hat{\boldsymbol{f}}$. Let $s \coloneqq \sum_i \frac{\lambda_i}{\lambda_i+\kappa}$ and $q \coloneqq \sum_i \frac{\lambda_i^2}{(\lambda_i+\kappa)^2}$, where $\kappa \geq 0$ is the unique nonnegative scalar such that*

$$nm = s + \frac{m\delta}{\kappa}.$$

Then, we can estimate the expected population risk as

$$\mathbb{E}_{\mathcal{D}}\big[\mathcal{R}(\hat{\boldsymbol{\beta}})\big] = \mathbb{E}\left[\|\hat{\boldsymbol{v}}-\boldsymbol{v}\|^2\right] \approx \frac{1}{1-\frac{q}{nm}}\left(\sum_i \frac{\kappa^2 v_i^2}{(\lambda_i+\kappa)^2} + \mathbb{E}\left[\|\boldsymbol{\epsilon}_1\|^2\right]\right) - \mathbb{E}\left[\|\boldsymbol{\epsilon}_1\|^2\right]. \tag{59}$$

Comparing (59) with $\mathcal{E}_{\text{te}}-m\sigma^2$ for $\mathcal{E}_{\text{te}}$ from (37), the population risk estimate on a clean test data without noise in the kernel ridge regression setting, we see that the only difference is that the term $m\sigma^2 = \mathbb{E}\left[\|\boldsymbol{\omega}_1\|^2\right]$ is replaced by $\mathbb{E}\left[\|\boldsymbol{\epsilon}_1\|^2\right]$. This observation supports our earlier claim that $\mathbb{E}\left[\|\boldsymbol{\epsilon}_1\|^2\right]$ can be interpreted as the *noise level*.

Finally, notice that in the derivation of (59), the only property of $\boldsymbol{\Lambda}$ we have used is that is symmetric. The risk estimate for the random feature regression is obtained by replacing $\boldsymbol{\Lambda}$ with $\tilde{\boldsymbol{\Lambda}}$ defined in (32), which is also a symmetric matrix. Hence, for the purpose of approximating the test risk in the random feature regression case, the same logic as above allows us to simply replace $m\sigma^2$ by $\mathbb{E}\left[\|\boldsymbol{\epsilon}_1\|^2\right]$ in the expression $\mathcal{E}_{\text{te}}-m\sigma^2$, where $\mathcal{E}_{\text{te}}$ is given in (36).

Observation F.3. *Assume that the multivariate Gaussian universality ansatz (Assumption 5.1) holds. Consider a regression model as in* (10)*, and suppose we perform a vector-valued random feature regression to obtain a regressor $\hat{\boldsymbol{f}}$. Let $s \coloneqq \sum_i \frac{\lambda_i}{\lambda_i+\gamma}$ and $q \coloneqq \sum_i \frac{\lambda_i^2}{(\lambda_i+\gamma)^2}$, where $\kappa,\gamma \geq 0$ are the unique nonnegative scalars such that*

$$nm = s + \frac{m\delta}{p\kappa} \qquad \textit{and} \qquad p = s + \frac{p\kappa}{\gamma}.$$

Then, we can estimate the expected population risk as

$$\mathbb{E}_{\mathcal{D}}\big[\mathcal{R}(\hat{\boldsymbol{\beta}})\big] \approx \frac{1}{1-\frac{q(p-2s)+s^2}{nm(p-q)}}\left(\sum_i\left(\frac{\gamma}{\lambda_i+\gamma}-\frac{\kappa\lambda_i}{(\lambda_i+\gamma)^2}\frac{p}{p-q}\right)v_i^2+\mathbb{E}\left[\|\boldsymbol{\epsilon}_1\|^2\right]\right)-\mathbb{E}\left[\|\boldsymbol{\epsilon}_1\|^2\right]. \tag{60}$$

F.3 Proof of Theorem 5.4

Theorem F.4 (Theorem 5.4)**.** *Let $\mathcal{R}_{\text{te}}(n,p,\delta)$ denote the value of $\mathcal{R}_{\text{te}}$ with dataset size n, number of features p, and ridge parameter δ. Suppose that $p \le p'$. Then, it holds that*

$$\min_\delta \mathcal{R}_{\text{te}}(n,p',\delta) \le \min_\delta \mathcal{R}_{\text{te}}(n,p,\delta). \tag{61}$$

Moreover, if we further assume that $p > nm$, then denoting $\mathcal{R}_{\text{te},0} = \lim_{\delta\to 0+}\mathcal{R}_{\text{te}}$, it holds that

$$\mathcal{R}_{\text{te},0}(n,p') \le \mathcal{R}_{\text{te},0}(n,p). \tag{62}$$

Proof. For convenience, let $\varsigma^2 \coloneqq \mathbb{E}\big[\|\boldsymbol{\epsilon}_1\|^2\big]$, and let $\mathcal{E}_{\text{te}}(n,p,\delta,\sigma^2)$ denote the quantity $\mathcal{E}_{\text{te}}$, defined in (36), evaluated with dataset size n, number of features p, ridge parameter δ, and noise level σ^2. Then, as discussed in the previous section, we have $\mathcal{R}_{\text{te}}(n,p,\delta) = \mathcal{E}_{\text{te}}(n,p,\delta,\frac{\varsigma^2}{m}) - \varsigma^2$. Moreover, if we denote by $\mathcal{E}'_{\text{te}}(n,p,\delta,\sigma^2)$ the corresponding $\mathcal{E}_{\text{te}}$ quantity in the one-dimensional label setting, then, as also discussed in the previous section, we have $\mathcal{E}_{\text{te}}(n,p,\delta,\sigma^2) = \mathcal{E}'_{\text{te}}(nm,p,\delta m,m\sigma^2)$, yielding the identity

$$\mathcal{R}_{\text{te}}(n,p,\delta) = \mathcal{E}'_{\text{te}}(nm,p,\delta m,\varsigma^2) - \varsigma^2. \tag{63}$$

We claim that the first part of the statement follows from Theorem 1 of Simon et al. [57], which states that $p \mapsto \min_\delta \mathcal{E}'_{\text{te}}(n,p,\delta,\sigma^2)$ is nonincreasing. Indeed, in our regression model, ς^2 is induced by the regression (10); hence it may depend on n, but it clearly does not depend on the number of features p. Therefore, as long as n is kept fixed, it follows directly from (63) that the inequality (61) holds.

Similarly, the second part of the statement follows from Proposition 2 of Simon et al. [57] which, denoting $\mathcal{E}'_{\text{te},0} = \lim_{\delta\to 0+}\mathcal{E}'_{\text{te}}$, states that $\frac{\partial \mathcal{E}'_{\text{te},0}}{\partial p} \le 0$ as long as $p > n$. Indeed, from (63) we have

$$\frac{\partial}{\partial p}\mathcal{R}_{\text{te},0}(n,p) = \frac{\partial}{\partial p}\left(\lim_{\delta\to 0+}\mathcal{E}'_{\text{te}}(nm,p,\delta m,\varsigma^2) - \varsigma^2\right) = \frac{\partial}{\partial p}\mathcal{E}'_{\text{te},0}(nm,p,\varsigma^2).$$

Therefore, whenever $p > nm$, we get $\frac{\partial}{\partial p}\mathcal{R}_{\text{te},0}(n,p) \le 0$, which implies the inequality (62). □

F.4 More features is better also for neural-network-like random feature models

Theorem 5.4 states that increasing the number of features is beneficial in reducing the population risk in random feature regression. The following theorem shows that the same phenomenon occurs for neural-network-like random feature models, introduced in Section C.5.

Theorem F.5. *Consider neural-network-like random feature models. Let $\mathcal{R}_{\text{te}}(n,p,\delta)$ denote the right hand side of* (40) *with dataset size n, number of features p, and ridge parameter δ. Suppose that $p \le p'$. Then, it holds that*

$$\min_\delta \mathcal{R}_{\text{te}}(n,p',\delta) \le \min_\delta \mathcal{R}_{\text{te}}(n,p,\delta).$$

Moreover, if we further assume that $p > n$, then denoting $\mathcal{R}_{\text{te},0} = \lim_{\delta\to 0+}\mathcal{R}_{\text{te}}$, it holds that

$$\mathcal{R}_{\text{te},0}(n,p') \le \mathcal{R}_{\text{te},0}(n,p).$$

Proof. Recall that $\mathcal{R}_{\text{te}}(n,p,\delta)$ for neural-network-like models is the sum of coordinatewise population risk estimates. More precisely, observe, from (40), that $\mathcal{R}_{\text{te}}(n,p,\delta)$ in this case is

$$\mathbb{E}_{\mathcal{D}}[\mathcal{R}(\hat{\beta})] \approx \frac{1}{1-\frac{q(p-2s)+s^2}{n(p-q)}}\left(\sum_j\left(\frac{\gamma}{\lambda_j+\gamma}-\frac{\kappa\lambda_j}{(\lambda_j+\gamma)^2}\frac{p}{p-q}\right)\sum_{i=1}^m v_{ij}^2+\varsigma^2\right)-\varsigma^2.$$

Therefore, the inequalities are immediate from Theorem 1 and Proposition 2 of [57], similar to the proof of Theorem 5.4. □

G Experimental Results on Random Feature Models

In this section, we discuss the claims regarding random feature models, including those concerning the approximation of the population risk by $\mathcal{R}_{\text{te}}$ as stated in (15), and support them with empirical results.

G.1 Random feature models on Gaussian distributions

As an empirical validation of our extension of the theories to the setting of multidimensional targets, we consider the setting where both the latent and the target distributions are multivariate Gaussians, $\xi = \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ and $\nu = \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$.

One advantage of working with Gaussians is that we have an explicit formula for the OT map, allowing us to seamlessly apply our theories.

Theorem G.1 ([48], Remark 2.31)**.** *For $\xi = \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ and $\nu = \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ with $\boldsymbol{\Sigma}_0$ invertible, the optimal transport map G^* from ξ to ν is an affine function of the form*

$$G^*(\boldsymbol{z}) = \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_0^{-1/2}\Big(\boldsymbol{\Sigma}_0^{1/2}\boldsymbol{\Sigma}_1\boldsymbol{\Sigma}_0^{1/2}\Big)^{1/2}\boldsymbol{\Sigma}_0^{-1/2}(\boldsymbol{z} - \boldsymbol{\mu}_0).$$

As an instantiation of the manifold hypothesis, we set $\boldsymbol{\Sigma}_1$ to be degenerate, so that ν is supported on an affine subspace whose dimensionality is lower than its ambient space.

The noise part, *i.e.*, the term $\mathbb{E}\big[\|\boldsymbol{\epsilon}_1\|^2\big]$ in the formula (15) of $\mathcal{R}_{\text{te}}$, is estimated based on (43). More precisely, given samples $\boldsymbol{z}_1, \dots, \boldsymbol{z}_n \overset{\text{i.i.d.}}{\sim} \xi$ and $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \overset{\text{i.i.d.}}{\sim} \nu$, let $\tilde{\pi} \in S_n$ be the permutation such that the optimal transport map from $\hat{\xi}$ to $\hat{\nu}$ is $\boldsymbol{z}_i \mapsto \boldsymbol{x}_{\tilde{\pi}(i)}$. Then (43) tells us that

$$\begin{aligned}\mathbb{E}[\|\boldsymbol{\epsilon}_1\|^2] &= \frac{1}{n}\sum_{j=1}^{n}\mathbb{E}\big[\|\check{\boldsymbol{\epsilon}}_j\|^2\big] \\ &= \frac{1}{n}\sum_{j=1}^{n}\mathbb{E}\big[\|\boldsymbol{x}_{\tilde{\pi}(j)} - G^*(\boldsymbol{z}_j)\|^2\big].\end{aligned}$$

Hence, in the experiments, a Monte Carlo estimate of $\frac{1}{n}\sum_{j=1}^{n}\|\boldsymbol{x}_{\tilde{\pi}(j)} - G^*(\boldsymbol{z}_j)\|^2$, computed using freshly sampled datasets that are not used during the regression phase, may be used to approximate $\mathbb{E}[\|\boldsymbol{\epsilon}_1\|^2]$.

For the function $\boldsymbol{g}$ determining the (random) feature map $\boldsymbol{\psi}$ in (11), we chose

$$\boldsymbol{g}(\boldsymbol{w}, \boldsymbol{x}) = \begin{bmatrix} \sin(\boldsymbol{\omega}_1^\top \boldsymbol{x} + b_1) \\ \vdots \\ \sin({\boldsymbol{\omega}_m}^\top \boldsymbol{x} + b_m) \end{bmatrix}$$

where $\boldsymbol{w}$ here is used to denote the collection of all the weights $\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_m \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\boldsymbol{0}, 4^2\boldsymbol{I})$ and the biases $b_1, \dots, b_m \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1)$. We remark that this is essentially ρ being a Gaussian distribution on $\mathbb{R}^{dm+m}$.

Figure 5a is a result when the ambient space is $\mathbb{R}^2$, with

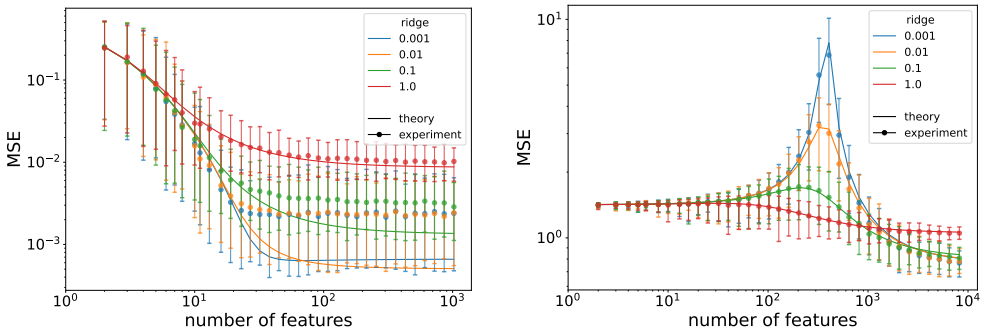
$$\boldsymbol{\mu}_0 = \begin{bmatrix}0.5\\0.5\end{bmatrix}, \quad \boldsymbol{\Sigma}_0^{1/2} = \begin{bmatrix}0.16 & 0\\0 & 0.08\end{bmatrix},$$

$$\boldsymbol{\mu}_1 = \begin{bmatrix}-0.5\\-0.5\end{bmatrix}, \quad \boldsymbol{\Sigma}_1^{1/2} = \begin{bmatrix}\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\end{bmatrix}\begin{bmatrix}0.16 & 0\\0 & 0\end{bmatrix}\begin{bmatrix}\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\end{bmatrix}^\top.$$

In this case, we set $n = 64$. In each of the Monte Carlo integrations to compute the "true" population risks, 256 samples were used.

Figure 5b is a result when the ambient space is $\mathbb{R}^3$, setting $\xi = \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ a standard normal distribution and $\nu = \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ with

$$\boldsymbol{\mu}_1 = \begin{bmatrix}1\\\frac{1}{2}\\\frac{1}{3}\end{bmatrix} \quad \text{and} \quad \boldsymbol{\Sigma}_1^{1/2} = \boldsymbol{R}\begin{bmatrix}0.1 & 0 & 0\\0 & 0.2 & 0\\0 & 0 & 0\end{bmatrix}\boldsymbol{R}^\top,$$



(a) Between 2D Gaussians (b) Between 3D Gaussians

Figure 5: **Learning optimal transport maps with random feature models.** We plot the computed theoretical and experimental population risks. The meaning of the curves and scatter plots are explained in Section G.1.

where $\boldsymbol{R}$ is a rotation matrix

$$\boldsymbol{R} = \begin{bmatrix} \frac{1}{2} & 0 & -\frac{\sqrt{3}}{2} \\ 0 & 1 & 0 \\ \frac{\sqrt{3}}{2} & 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

In this case, we set $n = 128$. We chose the latent distribution to be a standard normal distribution, mimicking the standard practice in modern generative models. In each of the Monte Carlo integrations to compute the "true" population risks, 1024 samples were used.

Figure 5 shows the plots of the results. In the figure, the curves drawn with solid lines, labeled theory, are the plots of the estimated population risk $\mathcal{R}_{\text{te}}$, computed by (15). The scatter plots with vertical error bars, labeled experiment, represent the "true" population risks computed by 200 iterations of Monte Carlo integration. The circle markers are the means, and the error bars range from the 2.5th percentile to the 97.5th percentile of the results from those 200 iterations. The plots demonstrate that $\mathcal{R}_{\text{te}}$ in (15) reasonably approximates the expected population risk $\mathbb{E}_{\mathcal{D}}\big[\mathcal{R}(\hat{\boldsymbol{\beta}})\big]$, concurrently capturing the overall descending tendency with respect to the number of features. These results demonstrate that $\mathcal{R}_{\text{te}}$ serves as a reliable proxy for the expected population risk, thereby empirically supporting the "more features is better" principle derived via $\mathcal{R}_{\text{te}}$.

Both results in Figure 5 show that $\mathcal{R}_{\text{te}}$ is a fairly accurate approximation of the population risk $\mathbb{E}_{\mathcal{D}}\big[\mathcal{R}(\hat{\boldsymbol{\beta}})\big]$. This empirically validates the discussions we made in Section C.

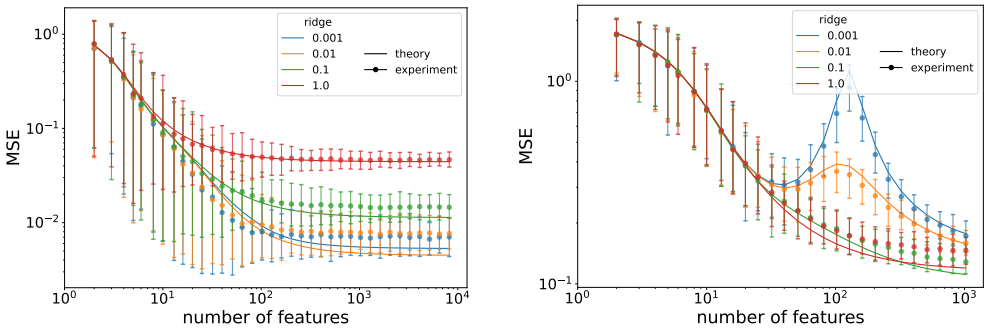
G.2 Neural-network-like model experiments

We also experimented with neural-network-like models, testing the potency of the formula (40) as the estimator of the population risk.

We performed experiments in two different settings; once in $\mathbb{R}^3$, and once in $\mathbb{R}^8$. The target and latent distributions in the $\mathbb{R}^3$ experiment are the same as Section G.1. For the experiment in $\mathbb{R}^8$, since neural-network-like models are inspired by architectures used in practice, we again adopt the standard normal distribution as the latent distribution ξ. For the target distribution, similar to the case in $\mathbb{R}^3$, we set $\nu = \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$, where

$$\boldsymbol{\mu}_1 = \left(1, \frac{1}{2}, \ldots, \frac{1}{8}\right)$$

and $\boldsymbol{\Sigma}_1 = \boldsymbol{Q}\boldsymbol{D}\boldsymbol{Q}^\top$ with $\boldsymbol{Q}$ a randomly generated orthogonal matrix obtained by a QR decomposition of a random matrix with i.i.d. standard normal entries and $\boldsymbol{D} = \operatorname{diag}(0.1, 0.2, \ldots, 0.5, 0, 0, 0)$. The last three entries of $\boldsymbol{D}$ are deliberately set to zero so that $\boldsymbol{\Sigma}_1$ is of rank 5 and consequently $\boldsymbol{\mu}_1$ is supported on a five-dimensional manifold in $\mathbb{R}^8$, as again an instantiation of the manifold hypothesis. We set $n = 128$ and use 1024 samples in each of the Monte Carlo integrations to compute the "true" population risks, as in the $\mathbb{R}^3$ setting.



(a) Between 3D Gaussians

(b) Between 8D Gaussians

Figure 6: **Learning the optimal transport maps with neural-network-like models.** We plot the computed theoretical and experimental population risks.

Meanwhile, we use a different feature map to better resemble neural networks used in practice. Specifically, we replace the nonlinearity with the hyperbolic tangent function and take

$$\boldsymbol{g}(\boldsymbol{w}, \boldsymbol{x}) = \begin{bmatrix} \tanh(\boldsymbol{\omega}_1^\top \boldsymbol{x} + b_1) \\ \vdots \\ \tanh(\boldsymbol{\omega}_m{}^\top \boldsymbol{x} + b_m) \end{bmatrix},$$

where $\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_m \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ and $b_1, \ldots, b_m \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1)$, with the random feature map $\boldsymbol{\psi}$ following the configuration of (38). Other details remain identical to Section G.1.

Figure 6 shows the plots of the results, depicted in the same way with Figure 5. The curves and the scatter plots are drawn in the exact same way as in the previous experiments. Again, we can observe an agreement between actual experimental results and the theoretical estimation, which this time is computed using (40).

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the paper's claims and contributions. Both the theoretical and experimental results are consistent with these claims.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss the limitations of our work in Section 6.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: We specify the assumptions imposed in the each of the statements, and their complete proofs are presented in the appendices.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The experiments are numerical computations, so revealing the settings and details is sufficient for reproducibility. Details of the experiments are provided in Section 3 and Section B.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: The exact codes used for the experiments are attached as supplementary materials.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We specify our experimental setup, including the use of flow matching models and other relevant settings.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: The experimental results in the main paper are reported with error bars, including how the reader should interpret those. For the experiments in the appendices, we include error bars where appropriate, and whenever they are omitted we provide a justification.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We report the computational environment and resource requirements in Section B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research fully complies with the NeurIPS Code of Ethics in all respects.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: Since our work focuses on the theoretical aspects of generative models, we believe it does not have a direct societal impact.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: Our paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: Packages used in the experiments, such as `NumPy`, `SciPy`, `Matplotlib`, and `POT`, are cited. Also, we use the publicly available MNIST and CIFAR-10 datasets and cite their references.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: Our paper does not introduce new assets, such as a new architecture.

    Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

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

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

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

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method in this paper does not involve LLMs; we use LLMs for grammar checking and improving sentence flow, if necessary.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.