Learning to Extrapolate: A Transductive Approach

Aviv Netanyahu* MIT

Abhishek Gupta* MIT

Max Simchowitz MIT

Kaiqing Zhang MIT

Pulkit Agrawal MIT

Abstract

Machine learning systems, especially overparameterized deep neural networks, can generalize to novel testing instances drawn from the same distribution as the training data. However, they fare poorly when evaluated on out-of-support testing points. In this work, we tackle the problem of developing machine learning systems that retain the power of overparametrized function approximators, while enabling extrapolation to out-of-support testing points when possible. This is accomplished by noting that under certain conditions, a "transductive" reparameterization can convert an out-of-support extrapolation problem into a problem of within-support combinatorial generalization. We propose a simple strategy based on bilinear embeddings to enable this type of combinatorial generalization, thereby addressing the out-of-support extrapolation problem. We instantiate a simple, practical algorithm applicable to various supervised learning problems and imitation learning tasks.

1 Introduction

Generalization is a central problem in machine learning. Typically, one expects generalization when the test data is sampled from *the same distribution* as the training set, i.e *out-of-sample* generalization. However, in many scenarios of interest, test data is sampled from a different distribution from the training set, i.e *out-of-distribution* (OOD). In some OOD scenarios, the test-distribution is assumed to be known during training – a common assumption made by meta-learning methods. Several works have tackled a more general scenario of "reweighted" distribution shift [13, 15] where the test distribution shares support with the training distribution, but has a different and unknown probability density; this setting can be tackled via distributional robustness approaches [20, 16]. We explore the scenario where test data is drawn from a distribution which has support *outside* that of the train distribution. Formally, assume the problem of learning function h: $\hat{y} = h_\theta(x)$ using data $\{(x_i, y_i)\}_{i=1}^N \sim \mathcal{D}_{\text{train}}$, where $x_i \in \mathcal{X}_{\text{train}}$, the train domain. We are interested in making accurate predictions $h(x)$ for $x \notin \mathcal{X}_{\text{train}}$. Consider an example task of predicting actions to reach a desired goal (Fig 1). During

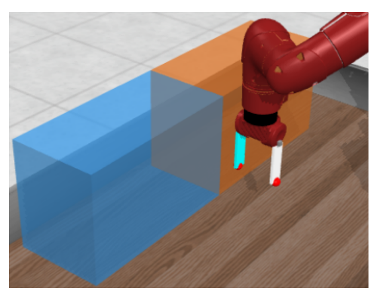


Figure 1: In the real-world the test distribution (orange) often has a *different support* than the training distribution (blue). Consider action prediction for reaching out-of-support goals. Conventional deep neural networks make accurate predictions for in-support inputs but fail out-of-support. We propose an algorithm that makes accurate out-of-support predictions under some assumptions.

*equal contribution. Correspondence to: Aviv Netanyahu <avivn@mit.edu>, Abhishek Gupta <abhgupta@cs.washington.edu>.

Workshop on Distribution Shifts, 36th Conference on Neural Information Processing Systems (NeurIPS 2022).

train, goals are provided from the blue cuboid ($x \in \mathcal{X}_{\text{train}}$), but test time goals are from the orange cuboid ($x \notin \mathcal{X}_{\text{train}}$). If f is modelled using a deep neural network, its predictions on test goals in the blue area are likely to be accurate, but in the orange area the performance can be arbitrarily poor. This challenge manifests itself in a variety of real world problems, ranging from object classification [6] to sequential decision making with reinforcement learning [12] and imitation learning [8]. Reliably deploying learning algorithms in unconstrained environments requires one to account for this type of "out-of-support" distribution shift.

If one can identify some *structure* in the training data that constrains the behavior of optimal predictors on novel data, then extrapolation may become possible. Several methods can extrapolate if the nature of distribution shift is known apriori: convolution neural networks are appropriate if a test-time training pattern appears at an out-of-distribution translation. Similarly, accurate predictions can be made for object point-clouds in out-of-support orientations by *building in* SE(3) equivariance [9, 18]. Another way to extrapolate is if the model class is known apriori: fitting a linear function to a linear problem will extrapolate. Similarly, methods like NeRF [26] use physics of image formation to learn a 3D model of a scene which can synthesize images from novel viewpoints.

In this work, we propose an alternative structural condition under which out-of-support extrapolation is feasible. Typical machine learning approaches are inductive: decision making rules are inferred from train data and employed for test predictions. An alternative to induction is *transduction* [10] or analogy-making where a test example is compared with training examples to make predictions. Our main insight is that in the transductive view of machine learning, out-of-support extrapolation can be reparameterized as a combinatorial generalization problem, which, under certain low-rank and coverage conditions [17, 2, 4, 3], admits a solution.

In this work we show how we can **(i)** re-parameterize out-of-support inputs $h(x_{test}) \to h(\Delta x, x')$, where $x' \in \mathcal{X}_{train}$, when provided a representation of measure of difference Δx between x_{test} and x'. **(ii)** Provide conditions under which $h(\Delta x, x')$ makes accurate predictions for unseen combinations $(\Delta x, x')$ **(iii)** based on a theoretically justified bilinear modeling approach: $h(\Delta x, x') \to f(\Delta x)^T g(x')$, where f and g map their inputs into same dimension vector spaces. **(iv)** Show empirical results demonstrating generality of extrapolation of our algorithm on: (a) regression for analytical functions and high-dimensional data; (b) sequential decision making tasks.

2 Setup

Notation. Given a space of inputs $\mathcal{X}$ and targets $\mathcal{Y}$, we aim to learn a predictor $h_{\boldsymbol{\theta}} : \mathcal{X} \to \mathcal{P}(\mathcal{Y})$[2] which best fits a ground truth function $h_\star : \mathcal{X} \to \mathcal{Y}$. Given some non-negative loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ on the outputs (e.g., square loss), and a distribution $\mathcal{D}$ over $\mathcal{X}$, *risk* is defined as

$$\mathcal{R}(h_{\boldsymbol{\theta}}; \mathcal{D}) := \mathbb{E}_{x\sim\mathcal{D}}\mathbb{E}_{y\sim h_{\boldsymbol{\theta}}(x)}\ell(y, h_\star(x)). \tag{2.1}$$

Various choices of train ($\mathcal{D}_{\text{train}}$) and test ($\mathcal{D}_{\text{test}}$) distributions yield different generalization settings:

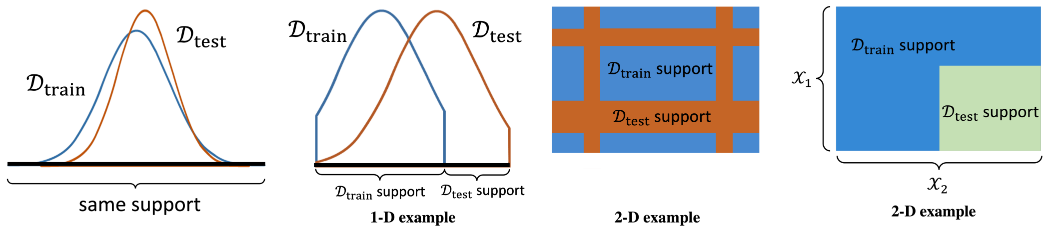
In-Distribution Generalization. This setting assumes $\mathcal{D}_{\text{test}} = \mathcal{D}_{\text{train}}$. The challenge is to ensure that with N samples from $\mathcal{D}_{\text{train}}$, the expected risk $\mathcal{R}(h_{\boldsymbol{\theta}}; \mathcal{D}_{\text{test}}) = \mathcal{R}(h_{\boldsymbol{\theta}}; \mathcal{D}_{\text{train}})$ is small. This is a common paradigm in both empirical supervised learning (e.g. [19]) and in standard statistical learning theory (e.g. [23]).

Out-of-Distribution (OOD). This is more challenging and requires accurate predictions when $\mathcal{D}_{\text{train}} \neq \mathcal{D}_{\text{test}}$. When the ratio between the density function of $\mathcal{D}_{\text{test}}$ to that of $\mathcal{D}_{\text{train}}$ is bounded, rigorous OOD extrapolation guarantees exist and are detailed in Appendix A.3. Such a situation arises when $\mathcal{D}_{\text{test}}$ shares support with $\mathcal{D}_{\text{train}}$ but is differently distributed as depicted in Fig 2a.

Out-of-Support (OOS). There are innumerable forms of distribution shift in which density ratios are not bounded. The most extreme case is when the support of $\mathcal{D}_{\text{test}}$ is not contained in that of $\mathcal{D}_{\text{train}}$. I.e., when there exists some $\mathcal{X}' \subset \mathcal{X}$ such that $\mathbb{P}_{x\sim\mathcal{D}_{\text{test}}}[x \in \mathcal{X}'] > 0$, but $\mathbb{P}_{x\sim\mathcal{D}_{\text{train}}}[x \in \mathcal{X}'] = 0$ (see Fig 2b). We term the problem of achieving low risk on such a $\mathcal{D}_{\text{test}}$ as OOS extrapolation.

Out-of-Combination (OOC). This is a special case of OOS. Let $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$ be the product of two spaces. Let $\mathcal{D}_{\text{train},\mathcal{X}_1}$, $\mathcal{D}_{\text{train},\mathcal{X}_2}$ denote the marginal distributions of $x_1 \in \mathcal{X}_1$, $x_2 \in \mathcal{X}_2$ under $\mathcal{D}_{\text{train}}$, and $\mathcal{D}_{\text{test},\mathcal{X}_1}$, $\mathcal{D}_{\text{test},\mathcal{X}_2}$ under $\mathcal{D}_{\text{test}}$. In OOC learning, $\mathcal{D}_{\text{test},\mathcal{X}_1}$, $\mathcal{D}_{\text{test},\mathcal{X}_2}$ are in the support of $\mathcal{D}_{\text{train},\mathcal{X}_1}$, $\mathcal{D}_{\text{train},\mathcal{X}_2}$, but the joint distributions $\mathcal{D}_{\text{test}}$ need not be in the support of $\mathcal{D}_{\text{train}}$.

[2]Throughout, we let $\mathcal{P}(\mathcal{Y})$ denote the set of distributions supported on $\mathcal{Y}$.



(a) OOD (b) General OOS (c) Structured OOS: OOC

Figure 2: Illustration of different learning settings. **(a)** in-support out-of-distribution (OOD) learning; **(b)** general out-of-support (OOS) learning (in 1-D and 2-D); **(c)** out-of-combination (OOC) learning.

3 Bilinear Transduction

To convert OOS to OOC, we require that $\mathcal{X}$ have a subtraction operator such that $x - x'$ is well-defined for $x, x' \in \mathcal{X}$. Let $\Delta\mathcal{X} := \{x - x' : x, x' \in \mathcal{X}\}$. We propose a *transductive re-parameterization* $h_{\boldsymbol{\theta}} : \mathcal{X} \to \mathcal{P}(\mathcal{Y})$ with a *deterministic* function $\bar{h}_{\boldsymbol{\theta}} : \Delta\mathcal{X} \times \mathcal{X} \to \mathcal{Y}$ as $h_{\boldsymbol{\theta}}(x) := \bar{h}_{\boldsymbol{\theta}}(x - x', x')$, where x' is referred to as an *anchor* point for a query point x.

Our basic proposal for OOS extrapolation is **unweighted transduction**, depicted in Algorithm 1: at train time, a predictor $\bar{h}_\theta$ is trained to make predictions for train points x_i drawn from the training set $\mathcal{D}_{\text{train}}$ based on their similarity with other points x_j also drawn from $\mathcal{D}_{\text{train}}$: $\bar{h}_\theta(x_i - x_j, x_j)$. The train pairs x_i, x_j are sampled uniformly from $\mathcal{D}_{\text{train}}$. At test time, for an OOS point x_{test}, we first select an anchor point x_i from the train set which has similarity with the test point $x_{\text{test}} - x_i$ that is within some radius ρ of the train similarities distribution $\Delta\mathcal{X}_{\text{train}}$. We then predict the value for x_{test} based on the anchor point x_i and the similarity of the test and anchor points: $\bar{h}_\theta(x_{\text{test}} - x_i, x_i)$.

Algorithm 1 Unweighted Transduction

Input: distance parameter ρ, training set $(x_1, y_1), \ldots, (x_n, y_n)$.
Train: Train $\boldsymbol{\theta}$ on loss $\mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^{n} \sum_{j \neq i} \ell(\bar{h}_{\boldsymbol{\theta}}(x_i - x_j, x_j), y_i)$
Test: for each new x_{test}, let $\mathcal{I}(x_{\text{test}}) := \{i : \inf_{\Delta x \in \Delta\mathcal{X}_{\text{train}}} \|x_{\text{test}} - x_i - \Delta x\| \leq \rho\}$, and predict

$$y = \bar{h}_{\boldsymbol{\theta}}(x_{\text{test}} - x_{\mathbf{i}}, x_{\mathbf{i}}), \text{ where } \mathbf{i} \sim \text{Uniform}(\mathcal{I}(x_{\text{test}}))$$

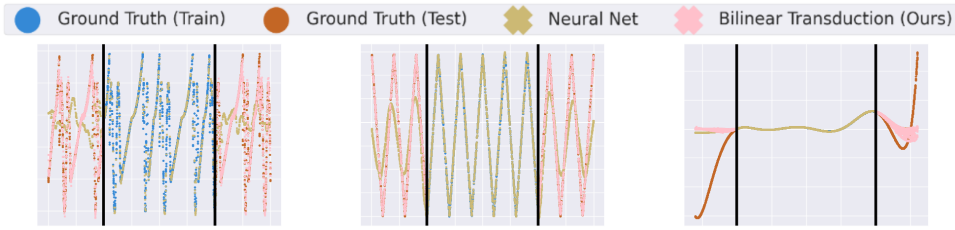
For the supervised regression setting, we compute differences directly between inputs $x_i, x_j \in \mathcal{X}$. For goal-conditioned imitation learning, we compute difference between states $x_i, x_j \in \mathcal{X}$ sampled uniformly over demonstration trajectories. At test time, we select an anchor *trajectory* based on the goal, and transduce each anchor state in the anchor trajectory to predict a sequence of actions for a test goal. In practice, we select ρ to be some percentile of differences $\|x_{\text{test}} - x_i - \Delta x\|$. We provide formal theoretical analysis of transductive bilinear predictors and conditions under which OOS extrapolation can be achieved in Appendix A.

4 Experiments

What types of problems satisfy the assumptions for extrapolation? We consider functions with different structure: a periodic function with mixed periods (Fig 3a), a sawtooth function (Fig 3b) and a polynomial function (Fig 3c). Standard deep networks (yellow) fit the training points well (blue), but fail to extrapolate to OOS inputs (orange). In comparison, our approach (pink) accurately extrapolates on periodic functions but is much less effective on polynomials. This is because the periodic functions have symmetries which induce low rank structure under the proposed re-parameterization.

Going Beyond Known Inductive Biases In Fig 4, we show that bilinear transduction is able to extrapolate even in cases that the ground truth function is not simply translation invariant, but is translation *equivariant*. Fig 5 demonstrates bilinear transduction extrapolation for a function *neither invariant nor equivariant* to translation, compared with an equivariant baseline (green).

How does the relationship between the training distribution and testing distribution affect extrapolation behavior? We show in Fig 6 that for a particular "width" of the training distribution



(a) Mixed periodic function (b) Sawtooth function (c) Degree-8 polynomial

Figure 3: Bilinear transduction behavior on 1-D regression problems. Bilinear transduction performs well on functions with repeated structure, whereas they struggle on arbitrary polynomials. Standard neural nets fail to extrapolate in most settings, even when provided periodic activations [22].

(size of the training set), OOS extrapolation only extends for one "width" beyond the training range since the conditions for $\Delta\mathcal{X}$ being in-support are no longer valid beyond this point.

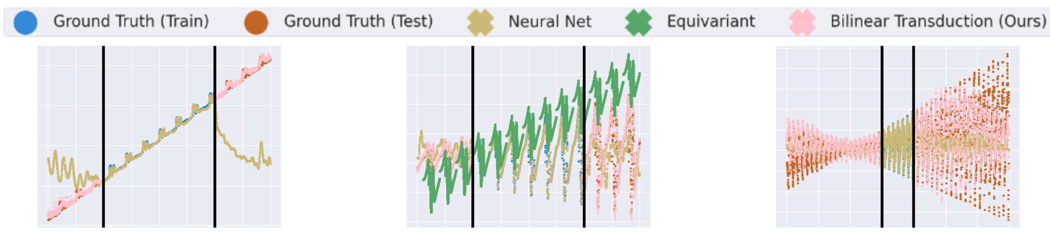


Figure 4: Function that displays affine equivariance.

Figure 5: Function that is neither invariant nor equivariant.

Figure 6: Predictions as test points go more and more OOS.

4.1 Analyzing OOS extrapolation on larger scale decision making problems

Baselines: Linear Model: linear function approximator to check whether linear models can solve the problem. **Neural Networks**: typical training of overparameterized neural network function approximators. Alternative Techniques with Neural Networks (**DeepSets** [25]): an alternative architecture for combining multiple inputs, that are meant to be permutation invariant and encourage a degree of generalization between different pairings of states and goals. Transductive Method without a mechanism for Structured Extrapolation (**Transduction**): transduction with no special structure, to check the impact of bilinear embeddings and low rank structure. This baseline uses reparameterization, and h_θ is a standard neural network.

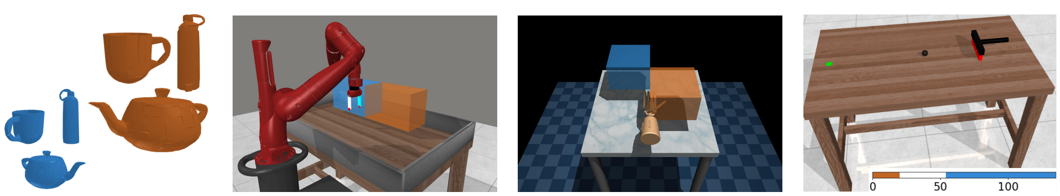


Figure 7: Evaluation domains at train (blue) and OOS (orange). (**Left to Right**:) grasp prediction for various objects and orientations, table-top robotic manipulation for reaching and pushing to various targets, dexterous manipulation for relocating objects to various targets, slider control for striking a ball of various mass

OOS Extrapolation in Sequential Decision Making:

• *Extrapolation to* OOS *Goals:* We consider two tasks from the Meta-World benchmark [24] where a simulated robotic agent needs to reach or push a target object to a goal location (Fig 7). Given a set of expert demonstrations reaching/pushing to goals in the blue box, we tested generalization to OOS goals in the orange box, using a simple extension of our method to perform transduction over trajectories rather than individual states. We quantify performance by measuring the distance between the conditioned and reached goal. Results in Table 1 show that on the easy task of reaching, training a typical linear or a neural network based predictor extrapolate as well as our method. However, for the more challenging task of pushing an object, our extrapolation is better by an order of magnitude than other baselines, showing the ability to generalize to goals in a completely different direction.

• *Extrapolation with large state and action space:* Next we tested our method on grasping and placing an object to OOS goal-locations in $\mathbb{R}^3$ with an anthropomorphic "Adroit" hand that has a much larger action ($\mathbb{R}^{30}$) and state ($\mathbb{R}^{39}$) space. Bilinear transduction is able to scale up to high dimensional state-action spaces as well and is naturally able to grasp the ball and move it to new target locations (with train and test distributions indicated in Fig 7).

• *Extrapolation to OOS Dynamics:* Lastly, we consider a slider task where the goal is to move a slider on a table to strike a ball such that it rolls to a fixed target position. The mass of the ball varies across episodes and is provided as input to the policy. We train and test on a range of masses (Fig 7). Bilinear transduction is able to successfully extrapolate to new masses and adjust behavior accordingly, showing the ability to extrapolate not just to goals, but also to varying dynamics.

OOS Extrapolation in Higher Dimensional Regression Problems. To scale up the dimension of the input space, we consider the problem of predicting valid grasping points (in $\mathbb{R}^3$) from point clouds of various objects (bottles, mugs and teapots) with different orientations, positions and scales (Fig 7). In this domain, we represent entire point clouds by a low dimensional representation of the point cloud obtained via PCA. We consider situations where the objects are not individually identified but instead a single grasp point predictor is trained on the entire set of bottles, mugs and teapots. We assume access to category labels at training time, but do not require this at test time.

Table 1: Mean and standard deviation over prediction (regression) or final state (sequential decision making) error for OOS samples and over a hyperparameter search.

Task	Expert	Linear	Neural Net	DeepSets	Transduction	Ours
Grasping		0.143 ± 0.116	0.118 ± 0.075		0.112 ± 0.08	$\mathbf{0.018 \pm 0.012}$
Reach	0.006 ± 0.008	0.007 ± 0.006	0.036 ± 0.054	0.19 ± 0.209	0.036 ± 0.048	$\mathbf{0.007 \pm 0.006}$
Push	0.012 ± 0.001	0.258 ± 0.063	0.258 ± 0.167	0.199 ± 0.114	0.159 ± 0.116	$\mathbf{0.02 \pm 0.017}$
Slider	0.105 ± 0.066	0.609 ± 0.07	0.469 ± 0.336	0.274 ± 0.262	0.495 ± 0.339	$\mathbf{0.149 \pm 0.113}$
Adroit	0.035 ± 0.015	0.337 ± 0.075	0.331 ± 0.203	0.521 ± 0.457	0.409 ± 0.32	$\mathbf{0.147 \pm 0.117}$

5 Discussion

In this work, we consider the problem of out-of-support extrapolation in regression and sequential decision making problems. We show that under some assumptions, extrapolation problems can be reparameterized using transduction to be viewed as combinatorial generalization problems. This allows us to leverage techniques from low-rank matrix completion in order to solve the combinatorial generalization problem. While our work serves as an initial study of the circumstances under which problem structure can be both *discovered* and *exploited* for understanding extrapolation, there are a number of natural questions for further research. First, can we classify which set of real-world domains fit our assumptions, beyond the domains we have demonstrated? Second, can we learn a latent space in which differences Δx are meaningful for high dimensional domains? And lastly, are there more effective schemes for selecting anchor points?

References

[1] A. Agarwal, A. Alomar, V. Alumootil, D. Shah, D. Shen, Z. Xu, and C. Yang. Persim: Data-efficient offline reinforcement learning with heterogeneous agents via personalized simulators. *Advances in Neural Information Processing Systems*, 34:18564–18576, 2021.

[2] A. Agarwal, M. Dahleh, D. Shah, and D. Shen. Causal matrix completion. *arXiv preprint arXiv:2109.15154*, 2021.

[3] J. Andreas. Measuring compositionality in representation learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[4] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Neural module networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 39–48. IEEE Computer Society, 2016.

[5] S. Athey, M. Bayati, N. Doudchenko, G. Imbens, and K. Khosravi. Matrix completion methods for causal panel data models. *Journal of the American Statistical Association*, 116(536):1716–1730, 2021.

[6] A. Barbu, D. Mayo, J. Alverio, W. Luo, C. Wang, D. Gutfreund, J. Tenenbaum, and B. Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. *Advances in neural information processing systems*, 32, 2019.

[7] E. Çinlar. *Probability and stochastics*, volume 261. Springer, 2011.

[8] P. de Haan, D. Jayaraman, and S. Levine. Causal confusion in imitation learning. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 11693–11704, 2019.

[9] C. Deng, O. Litany, Y. Duan, A. Poulenard, A. Tagliasacchi, and L. J. Guibas. Vector neurons: A general framework for so(3)-equivariant networks. *CoRR*, abs/2104.12229, 2021.

[10] A. Gammerman, V. Vovk, and V. Vapnik. Learning by transduction. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, page 148–155, 1998.

[11] A. Gittens and M. Mahoney. Revisiting the nystrom method for improved large-scale machine learning. In *International Conference on Machine Learning*, pages 567–575. PMLR, 2013.

[12] R. Kirk, A. Zhang, E. Grefenstette, and T. Rocktäschel. A survey of generalisation in deep reinforcement learning. *CoRR*, abs/2111.09794, 2021.

[13] P. W. Koh, S. Sagawa, H. Marklund, S. M. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R. L. Phillips, I. Gao, T. Lee, E. David, I. Stavness, W. Guo, B. Earnshaw, I. S. Haque, S. M. Beery, J. Leskovec, A. Kundaje, E. Pierson, S. Levine, C. Finn, and P. Liang. WILDS: A benchmark of in-the-wild distribution shifts. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 5637–5664. PMLR, 2021.

[14] L. Meng and B. Zheng. The optimal perturbation bounds of the moore–penrose inverse under the frobenius norm. *Linear algebra and its applications*, 432(4):956–963, 2010.

[15] J. Quinonero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset shift in machine learning*. Mit Press, 2008.

[16] H. Rahimian and S. Mehrotra. Distributionally robust optimization: A review, 2019.

[17] D. Shah, D. Song, Z. Xu, and Y. Yang. Sample efficient reinforcement learning via low-rank matrix estimation. *Advances in Neural Information Processing Systems*, 33:12092–12103, 2020.

[18] A. Simeonov, Y. Du, A. Tagliasacchi, J. B. Tenenbaum, A. Rodriguez, P. Agrawal, and V. Sitzmann. Neural descriptor fields: Se (3)-equivariant object representations for manipulation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6394–6400. IEEE, 2022.

[19] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[20] A. Sinha, H. Namkoong, and J. C. Duchi. Certifying some distributional robustness with principled adversarial training. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[21] G. W. Stewart. On the perturbation of pseudo-inverses, projections and linear least squares problems. *SIAM review*, 19(4):634–662, 1977.

[22] M. Tancik, P. P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. T. Barron, and R. Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *CoRR*, abs/2006.10739, 2020.

[23] V. Vapnik. *Estimation of dependences based on empirical data*. Springer Science & Business Media, 2006.

[24] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning*, pages 1094–1100. PMLR, 2020.

[25] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

[26] J. Zhang, Y. Zhang, H. Fu, X. Zhou, B. Cai, J. Huang, R. Jia, B. Zhao, and X. Tang. Ray priors through reprojection: Improving neural radiance fields for novel view extrapolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18376–18386, 2022.

A Appendix

A.1 Transductive Predictors: Converting OOS to OOC

To convert OOS to OOC, we require the input space $\mathcal{X}$ to have group structure, i.e. there are addition and subtraction operators such that $x + x', x - x'$ are well-defined for $x, x' \in \mathcal{X}$. Let $\Delta\mathcal{X} := \{x - x' : x, x' \in \mathcal{X}\}$. We propose a *transductive re-parameterization* $h_{\boldsymbol{\theta}} : \mathcal{X} \to \mathcal{P}(\mathcal{Y})$ with a *deterministic* function $\bar{h}_{\boldsymbol{\theta}} : \Delta\mathcal{X} \times \mathcal{X} \to \mathcal{Y}$ as

$$h_{\boldsymbol{\theta}}(x) := \bar{h}_{\boldsymbol{\theta}}(x - x', x') \tag{A.1}$$

where x' is referred to as an *anchor* point for a query point x. Under this re-parameterization, the training distribution can be rewritten as a *joint* distribution of train $\Delta x = x - x'$ and x' as follows

$$\mathbb{P}_{\bar{\mathcal{D}}_{\text{train}}}[(\Delta x, x') \in \cdot] := \Pr[(\Delta x, x') \in \cdot \ \mid x \sim \mathcal{D}_{\text{train}},\ x' \sim \mathcal{D}_{\text{train}}(x),\ \Delta x = x - x'] \tag{A.2}$$

This is just representing the prediction for every point from the training distribution in terms of it's relationship to other points in the training distribution.

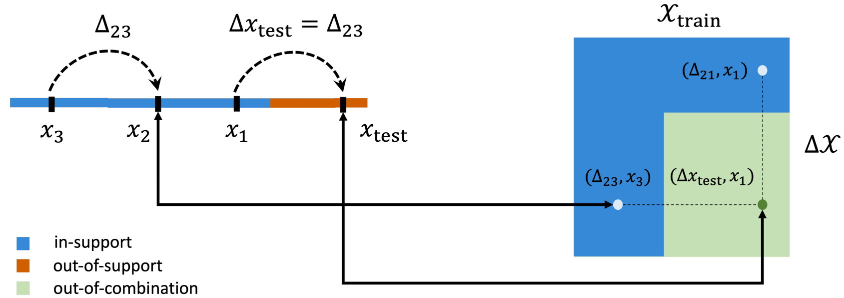


Figure 8: Illustration of converting OOS to OOC. (**Left**) Consider train points $x_1, x_2, x_3 \in \mathcal{X}_{\text{train}}$ and OOS test point x_{test}. During train, we predict $h_{\boldsymbol{\theta}}(x_2)$ by transducing x_3 to $h_{\boldsymbol{\theta}}(\Delta_{23}, x_3)$, where $\Delta_{23} = x_2 - x_3$. Similarly, at test time, we predict $h_{\boldsymbol{\theta}}(x_{\text{test}})$ by transducing train point x_1, via $h_{\boldsymbol{\theta}}(\Delta x_{\text{test}}, x_1)$, where $\Delta x_{\text{test}} = x_{\text{test}} - x_1$. In this example note that $\Delta_{23} = \Delta x_{\text{test}}$. (**Right**) This conversion yields an OOC generalization problem in space $\Delta\mathcal{X} \times \mathcal{X}_{\text{train}}$: marginal distributions $\Delta\mathcal{X}$ and $\mathcal{X}_{\text{train}}$ are covered by the train distribution, but their *combination* is not.

At test time, we are presented with point $x \sim \mathcal{D}_{\text{test}}$ that may be from an OOS distribution. To make a prediction on this OOS x, we make the observation that with a careful selection of an anchor point x', our reparameterization may be able to convert this OOS problem into a more manageable OOC one, since representing the test point x in terms of it's difference from training points can still be an "in-support" problem. To do so, we select an anchor point x' from $\mathcal{D}_{\text{train}}$ as follows. For a radius parameter $\rho > 0$, define the distribution of chosen anchor points $\mathcal{D}_{\text{trns}}(x)$ (referred to as a transducing distribution) as

$$\mathbb{P}_{\mathcal{D}_{\text{trns}}(x)}[x' \in \cdot] = \Pr[x' \in \cdot \mid x' \sim \mathcal{D}_{\text{train}},\ \inf_{\Delta x \in \Delta\mathcal{X}_{\text{train}}} \|(x - x') - \Delta x\| \leq \rho]. \tag{A.3}$$

where $\mathcal{X}_{\text{train}}$ denotes the set of x in our training set, and denote $\Delta\mathcal{X}_{\text{train}} := \{x_1 - x_2 : x_1, x_2 \in \mathcal{X}_{\text{train}}\}$. Intuitively, our choice of $\mathcal{D}_{\text{trns}}(x)$ selects anchor points x' to transduce from the training distribution, subject to the resulting differences $(x - x')$ being close to a "seen" $\Delta x \in \Delta\mathcal{X}_{\text{train}}$. In doing so, both the anchor point x' and the difference Δx have been seen individually at training time, albeit not in combination. This allows us to express the prediction for a OOS query point in terms of an in-support anchor point x' and an in-support difference Δx (but not jointly in support). This choice of anchor points induces a joint *test* distribution of $\Delta x = x - x'$ and x':

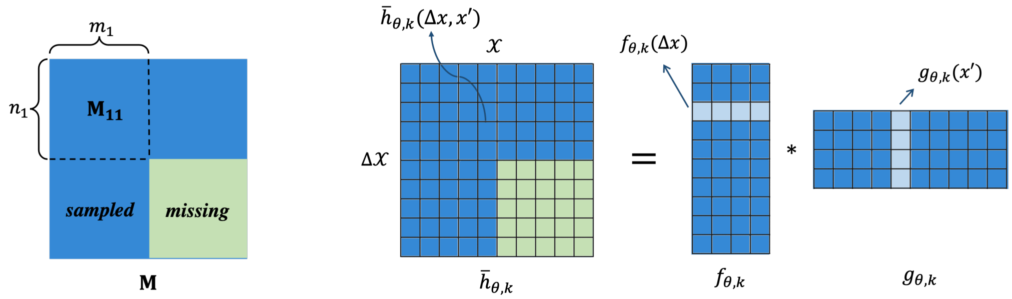
$$\mathbb{P}_{\bar{\mathcal{D}}_{\text{test}}}[(\Delta x, x') \in \cdot] := \Pr[(\Delta x, x') \in \cdot \ \mid x \sim \mathcal{D}_{\text{test}},\ x' \sim \mathcal{D}_{\text{trns}}(x),\ \Delta x = x - x']. \tag{A.4}$$

As seen from Fig 8, the marginals of Δx and x' under $\bar{\mathcal{D}}_{\text{test}}$, are individually in the support of those under $\bar{\mathcal{D}}_{\text{train}}$. Still, as Fig 8 reveals, since x_{test} is out-of-support, the joint distribution of $\bar{\mathcal{D}}_{\text{test}}$ is not covered by that of $\bar{\mathcal{D}}_{\text{train}}$ (i.e. the combination of x_1 and x_{test} have not been seen together before);

precisely the OOC regime. Moreover, if one tried to transduce *all* $x' \sim \mathcal{D}_{\text{train}}$ to $x \sim \mathcal{D}_{\text{test}}$ at test time (e.g. transduce point x_3 to x_{test} in the figure) then we would lose coverage of the Δx-marginal. By doing transduction to keep both the marginal x' and Δx in support, we are ensuring that we can convert difficult OOS problems into (potentially) more manageable OOC ones.

A.2 Bilinear representations for OOC learning

Without additional assumptions, OOC extrapolation may be just as challenging as OOS. However, with certain low-rank structure it can be feasible [17, 2, 5]. This is best illustrated in the case of matrix completion (see Fig 9a): let us consider a finite set of $x, \Delta x$, such that the OOC problem can be viewed as one of matrix completion (with rows and columns as Δx, x respectively). Consider a rank-p matrix $\mathbf{M} \in \mathbb{R}^{n \times m}$, and let $n_1 \le n$ and $m_1 \le m$ be such that the top-left $n_1 \times m_1$ block of $\mathbf{M}$, denoted $\mathbf{M}_{11}$, has rank p. Then, one can complete *entries* of $\mathbf{M}$, given only access to all entries (i, j) for which either $i \le n_1$ or $j \le m_1$. Such completion can be performed using SVD, where $\mathbf{M} = \mathbf{U}\Sigma\mathbf{V}^T$, where $\mathbf{U} \in \mathbb{R}^{n \times p}$, $\mathbf{V} \in \mathbb{R}^{m \times p}$, $\Sigma \in \mathbb{R}^{p \times p}$. $\mathbf{M}$ can we written as a bilinear function: $\mathbf{M} = \mathbf{U}'\mathbf{V}^T$, where $\mathbf{U}' = \mathbf{U}\Sigma$.



(a) Matrix completion

(b) Connecting OOC to matrix completion

Figure 9: Illustration of bilinear representations for OOC learning, and connection to matrix completion. **(a)** An example of low-rank matrix completion, where both $\mathbf{M}$ and $\mathbf{M}_{11}$ have rank-p. Blue: support where entries can be accessed, green: entries are missing. **(b)** An example that low-rank structure facilitates certain forms of OOC, i.e. for each $k \in [K]$, the predictor can be represented by bilinear embeddings as $\bar{h}_{\boldsymbol{\theta},k}(\Delta x, x') = \langle f_{\boldsymbol{\theta},k}(\Delta x), g_{\boldsymbol{\theta},k}(x') \rangle$.

Following [1], we recognize that this low-rank property can be leveraged *implicitly* for our reparameterized OOC problem even in the continuous case (where $x, \Delta x$ do not explicitly form a finite dimensional matrix) using a bilinear representation of the *transductive* predictor in Eq. (A.1), $\bar{h}_{\boldsymbol{\theta}}(\Delta x, x') = \langle f_{\theta_f}(\Delta x), g_{\theta_g}(x') \rangle$. Here $f_{\theta_f}, g_{\theta_g}$ map their respective inputs into a vector space of the same dimension (say p). If the output space is K dimensional, then we independently model the prediction for each dimension using a set of K bilinear embeddings:

$$\bar{h}_{\boldsymbol{\theta}}(\Delta x, x') = (\bar{h}_{\boldsymbol{\theta},1}(\Delta x, x'), \ldots, \bar{h}_{\boldsymbol{\theta},K}(\Delta x, x')); \bar{h}_{\boldsymbol{\theta},k}(\Delta x, x') = \langle f_{\boldsymbol{\theta},k}(\Delta x), g_{\boldsymbol{\theta},k}(x') \rangle. \quad \text{(A.5)}$$

While $\bar{h}_{\boldsymbol{\theta},k}$ are bilinear in embeddings $f_{\boldsymbol{\theta},k}, g_{\boldsymbol{\theta},k}$, the embeddings themselves may be parameterized by general function approximators. The effective "rank" of the transductive predictor is controlled by the dimension of the continuous embeddings $f_{\boldsymbol{\theta},k}(\Delta x), g_{\boldsymbol{\theta},k}(x')$. To illustrate the connection to matrix completion, we can imagine our predictors in Eq. (A.5) as matrices defining large look-up tables for each $(\Delta x, x')$ pair. See Fig 9b and a more detailed exposition in Section A.4. Leveraging the analysis of matrix completion in [17], we next provide formal theoretical analysis of transductive bilinear predictors and conditions under which OOS extrapolation can be achieved.

A.3 Generalization under bounded density ratio

The following gives a robust, quantitative notion of when one distribution is in the support of another. For generality, we state this condition in terms of general positive measures μ_1, μ_2, which need not be normalized and sum to one.

Definition A.1 (κ-bounded density ratio)**.** Let μ_1, μ_2 be two measures over a space Ω. We say μ_1 has κ-bounded density with respect to μ_2, which we denote $\mu_1 \ll_\kappa \mu_2$, if for all measurable[3] $A \subset \Omega$, $\mu_1[A] \le \kappa\mu_2[A]$.

[3]For simplicity, we omit concrete discussion of measurability concerns throughout.

Stating Definition A.1 for general probability affords us the flexibility to write example, $\mathbb{P}_1 \ll_\kappa \mathbb{P}_2 + \mathbb{P}_3$, as $\mathbb{P}_2 + \mathbb{P}_3$ is a nonnegative measure with total mass $1 + 1 = 2$.

The following lemma motivates the use of Definition A.1. Its proof is standard but included for completeness.

Lemma A.1. *Let μ_1, μ_2 be measures on the same measurable space Ω, and suppose that $\mu_2 \ll_\kappa \mu_1$. Then, for any nonnegative function ϕ, $\mu_2[\phi] \le \kappa\mu_1[\phi]$.[4] In particular, if $\mathcal{D}_{\text{test}} \ll_\kappa \mathcal{D}_{\text{train}}$, then as long as our loss function $\ell(\cdot,\cdot)$ is nonnegative,*

$$\mathcal{R}(h_{\boldsymbol{\theta}}; \mathcal{D}_{\text{test}}) \le \kappa \mathcal{R}(h_{\boldsymbol{\theta}}; \mathcal{D}_{\text{train}}).$$

Thus, up to a κ-factor, $\mathcal{R}(h_{\boldsymbol{\theta}}; \mathcal{D}_{\text{test}})$ inherits any in-distribution generalization guarantees for $\mathcal{R}(h_{\boldsymbol{\theta}}; \mathcal{D}_{\text{train}})$.

Proof. As in standard measure theory (c.f. [7, Chpater 1]), we can approximate any $\phi \ge 0$ by a sequence of *simple functions* $\phi_n \uparrow \phi$, where $\phi_n(\omega) = \sum_{i=1}^{k_n} c_{n,i}\mathbb{I}\{\omega \in A_{n,i}\}$, with $A_{n,i} \subset \Omega$ and $c_{n,i} \ge 0$. For each ϕ_n, we have

$$\mu_2[\phi_n] = \sum_{i=1}^{k_n} c_{n,i}\mu_2[A_{n,i}] \le \kappa \sum_{i=1}^{k_n} c_{n,i}\mu_1[A_{n,i}] = \mu_1[\phi_n].$$

The result now follows from the monotone convergence theorem. To derive the special case for $\mathcal{D}_{\text{test}}$ and $\mathcal{D}_{\text{train}}$, apply the general result with nonnegative function $\phi(x) = \mathbb{E}_{y \sim h_{\boldsymbol{\theta}}(x)}\ell(y, h_\star(x))$ (recall $\ell(\cdot,\cdot) \ge 0$ by assumption), $\mu_1 = \mathcal{D}_{\text{train}}$ and $\mu_2 = \mathcal{D}_{\text{test}}$. □

A.4 Extrapolation for Matrix Completion

In what follows, we derive a simple extrapolation guarantee for matrix completion. The following is in the spirit of the Nyström column approximation (see e.g. [11]), and our proof follows the analysis due to [17]. Throughout, consider

$$\hat{\mathbf{M}} = \begin{bmatrix} \hat{\mathbf{M}}_{11} & \hat{\mathbf{M}}_{12} \\ \hat{\mathbf{M}}_{21} & \hat{\mathbf{M}}_{22} \end{bmatrix}, \quad \mathbf{M}^\star = \begin{bmatrix} \mathbf{M}^\star_{11} & \mathbf{M}^\star_{12} \\ \mathbf{M}^\star_{21} & \mathbf{M}^\star_{22} \end{bmatrix},$$

where we decompose $\hat{\mathbf{M}}, \mathbf{M}^\star$ into blocks $(i,j) \in \{1,2\}^2$ for dimension $n_i \times m_j$.

Lemma A.2. *Suppose that $\hat{\mathbf{M}}$ is rank at most p, $\mathbf{M}^\star$ is rank p, and*

$$\forall (i,j) \ne (2,2), \quad \|\hat{\mathbf{M}}_{i,j} - \mathbf{M}^\star_{i,j}\|_{\text{F}} \le \epsilon, \quad \text{and } \|\mathbf{M}^\star_{i,j}\|_{\text{F}} \le M,$$

where $\epsilon \le \sigma_p(\mathbf{M}^\star_{11})/2$. Then,

$$\|\hat{\mathbf{M}}_{22} - \mathbf{M}^\star_{22}\|_{\text{F}} \le 8\epsilon \frac{M^2}{\sigma_p(\mathbf{M}^\star_{11})^2}.$$

Proof. The proof mirrors that of [17, Proposition 13]. We shall show below that $\hat{\mathbf{M}}$ is of rank exactly p. Hence, [17, Lemma 12] gives the following exact expression for the bottom-right blocks,

$$\hat{\mathbf{M}}_{22} = \hat{\mathbf{M}}_{21}\hat{\mathbf{M}}_{11}^\dagger\hat{\mathbf{M}}_{12}, \quad \mathbf{M}^\star_{22} = \mathbf{M}^\star_{21}(\mathbf{M}^\star_{11})^\dagger\mathbf{M}^\star_{12},$$

where above $(\cdot)^\dagger$ denotes the Moore-Penrose pseudoinverse. Since $\|\hat{\mathbf{M}}_{11} - \mathbf{M}^\star_{11}\|_{\text{op}} \le \|\hat{\mathbf{M}}_{11} - \mathbf{M}^\star_{11}\|_{\text{F}} \le \epsilon \le \sigma_p(\mathbf{M}^\star_{11})/2$, Weyls inequality implies that $\hat{\mathbf{M}}_{11}$ is rank p (as promised), and $\|\hat{\mathbf{M}}_{11}^\dagger\|_{\text{op}} \le 2\sigma_p(\mathbf{M}^\star_{11})^{-1}$. Similarly, as $\|\hat{\mathbf{M}}_{12} - \mathbf{M}^\star_{12}\|_{\text{op}} \le \sigma_p(\mathbf{M}^\star_{11})/2 \le M/2$, so $\|\hat{\mathbf{M}}_{12}\|_{\text{op}} \le \frac{3}{2}M$. Thus,

$$\begin{aligned}\|\hat{\mathbf{M}}_{22} - \mathbf{M}^\star_{22}\|_{\text{F}} &\le \|\hat{\mathbf{M}}_{21} - \mathbf{M}^\star_{21}\|_{\text{F}}\|\hat{\mathbf{M}}_{11}^\dagger\|_{\text{op}}\|\hat{\mathbf{M}}_{12}\|_{\text{op}} + \|\mathbf{M}^\star_{21}\|_{\text{op}}\|\hat{\mathbf{M}}_{11}^\dagger\|_{\text{op}}\|\mathbf{M}^\star_{12} - \hat{\mathbf{M}}_{12}\|_{\text{F}} \\ &\quad \|\mathbf{M}^\star_{12}\|_{\text{op}}\|\mathbf{M}^\star_{21}\|_{\text{op}}\|\hat{\mathbf{M}}_{11}^\dagger - (\mathbf{M}^\star_{11})^\dagger\|_{\text{F}} \\ &\le \frac{5\epsilon M}{2\sigma_p(\mathbf{M}^\star)} + M^2\|\hat{\mathbf{M}}_{11}^\dagger - (\mathbf{M}^\star_{11})^\dagger\|_{\text{F}}.\end{aligned}$$

[4]Here, $\mu[\phi] := \int \phi(\omega)\mathrm{d}\mu(\omega)$ denotes the integration with respect to μ.

Next, using a perturbation bound on the pseudoinverse[5] due to [14, Theorem 2.1],

$$\begin{aligned}\|\hat{\mathbf{M}}_{11}^{\dagger} - (\mathbf{M}_{11}^{\star})^{\dagger}\|_{\mathrm{F}} &\le \|\hat{\mathbf{M}}_{11} - \mathbf{M}_{11}^{\star}\|_{\mathrm{F}} \max\{\|\hat{\mathbf{M}}_{11}^{\dagger}\|_{\mathrm{op}}^2, \|(\mathbf{M}_{11}^{\star})^{\dagger}\|_{\mathrm{op}}^2\} \\ &\le \epsilon \cdot 4\sigma_p(\mathbf{M}_{11}^{\star})^{-2}.\end{aligned}$$

Therefore, we conclude

$$\|\hat{\mathbf{M}}_{22} - \mathbf{M}_{22}^{\star}\|_{\mathrm{F}} \le \frac{5\epsilon M}{2\sigma_p(\mathbf{M}^{\star})} + \epsilon \frac{4M^2}{\sigma_p(\mathbf{M}_{11}^{\star})^2} \le 8\epsilon \frac{M^2}{\sigma_p(\mathbf{M}_{11}^{\star})^2}.$$

□

A.5 General Analysis for Combinatioral Extrapolation

We now provide our general analysis for combinatorial extrapolation. To avoid excessive subscripts, we write $\mathcal{X} = \mathcal{W} \times \mathcal{V}$ rather than $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$ as in the main body. We consider extrapolation under the following definition of combinatorial support.

Definition A.2 (Bounded combinatorial density ratio, generic definition)**.** We say $\mathcal{D}_{\text{test}}$ has κ-*bounded combinatorial density ratio* with respect to $\mathcal{D}_{\text{train}}$, written as $\mathcal{D}_{\text{test}} \overset{\text{comb}}{\ll_{\kappa}} \mathcal{D}_{\text{train}}$, if there exist distributions $\mathcal{D}_{\mathcal{W},i}$ and $\mathcal{D}_{\mathcal{V},j}$, $i, j \in \{1, 2\}$, over $\mathcal{W}$ and $\mathcal{V}$, respectively, such that $\mathcal{D}_{i\otimes j} := \mathcal{D}_{\mathcal{W},i} \otimes \mathcal{D}_{\mathcal{V},j}$ satisfy

$$\sum_{(i,j)\neq(2,2)} \mathcal{D}_{i\otimes j} \ll_{\kappa} \mathcal{D}_{\text{train}}, \quad \text{and} \quad \mathcal{D}_{\text{test}} \ll_{\kappa} \sum_{i,j=1,2} \mathcal{D}_{i\otimes j},$$

where we recall the definition of κ-bounded density ratio notation $\ll_{\kappa}$ in Definition A.1.

Remark A.1 (Connection to matrix completion)**.** This definition of combinatorial support is the distributional equivalent of the four-block matrix completion setting depicted in Fig 9, where $\mathcal{D}_{\text{train}}$ covers the top-left, bottom-left, and top-right blocks (corresponding to $\mathcal{D}_{i,j}$ for $(i, j) \neq (2, 2)$, but $\mathcal{D}_{\text{test}}$ may also include samples from the bottom-right block as well (corresponding to $\mathcal{D}_{2\times 2}$). It is this structure that allows us to leverage the matrix-completion guarantee Lemma A.2 from the previous section to establish a combinatorial extrapolation guarantee below.

For simplicity, we consider scalar predictors, as the general result for vector valued estimators can be obtained by stacking the components. Specifically, we consider a ground-truth predictor $h_{\star}$ and estimator $\hat{h}$ of the form

$$h_{\star} = \langle f_{\star}, g_{\star}\rangle, \quad \hat{h} = \langle \hat{f}, \hat{g}\rangle, \quad f_{\star}, \hat{f} : \mathcal{W} \to \mathbb{R}^p,\ g_{\star}, \hat{g} : \mathcal{V} \to \mathbb{R}^p. \tag{A.6}$$

Lastly, we choose the (scalar) square-loss, yielding the following risk

$$\mathcal{R}(\hat{h}; \mathcal{D}) := \mathbb{E}_{(w,v)\sim\mathcal{D}}[(h_{\star}(w, v) - \hat{h}(w, v))^2].$$

Throughout, we assume that all expectations that arise are finite. Our main guarantee is as follows.

Theorem 1. *Define the effective singular value*

$$\sigma_{\star}^2 := \sigma_p(\mathbb{E}_{\mathcal{D}_{\mathcal{W},1}}[f_{\star}(w)f_{\star}(w)^{\top}])\sigma_p\left(\mathbb{E}_{\mathcal{D}_{\mathcal{V},1}}[g_{\star}(v)g_{\star}(v)^{\top}]\right), \tag{A.7}$$

and suppose that $\max_{1\le i,j\le 2} \mathbb{E}_{\mathcal{D}_{i\otimes j}} |h_{\star}(w, v)|^2 \le M_{\star}^2$*. Then, if* $\mathcal{R}(\hat{h}; \mathcal{D}_{\text{train}}) \le \frac{\sigma_{\star}^2}{4\kappa}$*,*

$$\mathcal{R}(\hat{h}; \mathcal{D}_{\text{test}}) \le \mathcal{R}(\hat{h}; \mathcal{D}_{\text{train}}) \cdot \kappa^2 \left(1 + 64\frac{M_{\star}^4}{\sigma_{\star}^4}\right) = \mathcal{R}(\hat{h}; \mathcal{D}_{\text{train}}) \cdot \mathrm{poly}\left(\kappa, \frac{M_{\star}}{\sigma_{\star}}\right).$$

A.5.1 Proof of Theorem 1

First, let us assume the following two conditions hold; we shall derive these conditions from the conditions of Theorem 1 at the end of the proof:[6]

$$\forall (i, j) \neq (2, 2), \quad \mathcal{R}(\hat{h}; \mathcal{D}_{i\otimes j}) \le \epsilon^2, \quad \mathbb{E}_{\mathcal{D}_{i\otimes j}}[h_{\star}(w, v)^2] \le M_{\star}^2, \quad \epsilon < \sigma_{\star}/2. \tag{A.8}$$

[5] Unlike [17], we are interested in the Frobenius norm error, so we elect for the slightly sharper bound of [14] above than the classical operator norm bound of [21].

[6] Notice that here we take $M_{\star}^2$ as an upper bound of $\mathbb{E}_{\mathcal{D}_{i\otimes j}}[h_{\star}(w, v)^2]$, rather than a pointwise upper bound in Theorem 1. This is for convenience in a limiting argument below.

Our strategy is first to prove a version of Theorem 1 for when $\mathcal{W}$ and $\mathcal{V}$ have finite cardinality by reduction to the analysis of matrix completion in Lemma A.2, and then extend to arbitrary domains via a limiting argument.

Lemma A.3. *Suppose that Eq. (A.8) hold, and in addition, that $\mathcal{W}$ and $\mathcal{V}$ have finite cardinality. Then,*

$$\mathcal{R}(\hat{h}; \mathcal{D}_{2\otimes 2}) = \|\hat{\mathbf{M}}_{22} - \mathbf{M}^{\star}_{22}\|_{\mathrm{F}}^2 \le 64\epsilon^2 \frac{M_{\star}^4}{\sigma_{\star}^4}.$$

Proof Lemma A.3. By adding additional null elements to either $\mathcal{W}$ or $\mathcal{V}$, we may assume without loss of generality that $|\mathcal{W}| = |\mathcal{V}| = d$, and enumerate their elements $\{w_1, \dots, w_d\}$ and $\{v_1, \dots, v_d\}$. Let $\mathsf{p}_{i,a} = \Pr_{w \sim \mathcal{D}_{\mathcal{W},i}}[w = w_a]$ and $\mathsf{q}_{j,b} = \Pr_{v \sim \mathcal{D}_{\mathcal{V},j}}[v = v_b]$. Consider matrices $\hat{\mathbf{M}}, \mathbf{M}^{\star} \in \mathbb{R}^{2d \times 2d}$, with $d \times d$ blocks

$$(\hat{\mathbf{M}}_{ij})_{ab} = \sqrt{\mathsf{p}_{i,a}\mathsf{q}_{j,b}} \cdot \hat{h}(w_a, v_b), \quad (\mathbf{M}^{\star}_{ij})_{ab} = \sqrt{\mathsf{p}_{i,a}\mathsf{q}_{j,b}} \cdot h_{\star}(w_a, v_b).$$

We then verify that

$$\begin{aligned} \|\hat{\mathbf{M}}_{ij} - \mathbf{M}^{\star}_{ij}\|_{\mathrm{F}}^2 &= \sum_{a,b=1}^{d} \mathsf{p}_{i,a}\mathsf{q}_{j,b}(\hat{h}(w_a, v_b) - h_{\star}(w_a, v_b))^2 \\ &= \mathbb{E}_{\mathcal{D}_{i\otimes j}}[(\hat{h}(w, v) - h_{\star}(w, v))^2] = \mathcal{R}(\hat{h}; \mathcal{D}_{i\otimes j}), \end{aligned} \tag{A.9}$$

and thus $\|\hat{\mathbf{M}}_{ij} - \mathbf{M}^{\star}_{ij}\|_{\mathrm{F}}^2 \le \epsilon^2$ for $(i, j) \neq (2, 2)$. Furthermore, define the matrices $\hat{\mathbf{A}}_i, \hat{\mathbf{B}}_j$ via

$$(\hat{\mathbf{A}}_i)_a := \sqrt{\mathsf{p}_{i,a}}\hat{f}(w_a)^{\top}, \quad (\hat{\mathbf{B}}_j)_b := \sqrt{\mathsf{q}_{j,b}}\hat{g}(v_b)^{\top},$$

and define $\mathbf{A}^{\star}_i, \mathbf{B}^{\star}_j$ similarly. Then,

$$\hat{\mathbf{M}} = \begin{bmatrix} \hat{\mathbf{A}}_1 \\ \hat{\mathbf{A}}_2 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{B}}_1 \\ \hat{\mathbf{B}}_2 \end{bmatrix}^{\top}, \quad \mathbf{M}^{\star} = \begin{bmatrix} \mathbf{A}^{\star}_1 \\ \mathbf{A}^{\star}_2 \end{bmatrix} \begin{bmatrix} \mathbf{B}^{\star}_1 \\ \mathbf{B}^{\star}_2 \end{bmatrix}^{\top},$$

showing that $\mathrm{rank}(\hat{\mathbf{M}}_1), \mathrm{rank}(\hat{\mathbf{M}}_2) \le p$. Finally, by Eq. (A.7),

$$\begin{aligned} \sigma_p(\mathbf{M}^{\star}_{11})^2 &= \sigma_p(\mathbf{A}^{\star}_1(\mathbf{B}^{\star}_1)^{\top})^2 \ge \sigma_p^2(\mathbf{A}^{\star}_1)\sigma_p^2(\mathbf{B}^{\star}_1) \\ &= \sigma_p\left((\mathbf{A}^{\star}_1)^{\top}\mathbf{A}^{\star}_1\right)\sigma_p\left((\mathbf{B}^{\star}_1)^{\top}\mathbf{B}^{\star}_1\right) \\ &= \sigma_p\left(\sum_{a=1}^{d} \mathsf{p}_{1,a}\hat{f}(w_a)\hat{f}(w_a)^{\top}\right)\sigma_p\left(\sum_{b=1}^{d} \mathsf{q}_{1,b}\hat{g}(v_b)\hat{g}(v_b)^{\top}\right) \\ &= \sigma_p(\mathbb{E}_{\mathcal{D}_{\mathcal{W},1}}[\hat{f}(w)\hat{f}(w)^{\top}])\sigma_p\left(\mathbb{E}_{\mathcal{D}_{\mathcal{V},1}}[\hat{g}(v)\hat{g}(v)^{\top}]\right) = \sigma_{\star}^2. \end{aligned}$$

Lastly, we have

$$\|\mathbf{M}^{\star}_{i,j}\|_{\mathrm{F}}^2 = \sum_{a,b} \mathsf{p}_{i,a}\mathsf{q}_{j,b}h_{\star}(w_a, v_b)^2 = \mathbb{E}_{\mathcal{D}_{i\otimes j}}h_{\star}(w, v)^2 \le M_{\star}^2.$$

Thus, Eq. (A.9) and Lemma A.2 imply that

$$\mathcal{R}(\hat{h}; \mathcal{D}_{2\otimes 2}) = \|\hat{\mathbf{M}}_{22} - \mathbf{M}^{\star}_{22}\|_{\mathrm{F}}^2 \le 64\epsilon^2 \frac{M_{\star}^4}{\sigma_{\star}^4}.$$

$\square$

Lemma A.4. *Suppose that Eq. (A.8) hold, but unlike Lemma A.4, $\mathcal{W}$ and $\mathcal{V}$ need not be finite spaces. Then, still, it holds that*

$$\mathcal{R}(\hat{h}; \mathcal{D}_{2\otimes 2}) = \|\hat{\mathbf{M}}_{22} - \mathbf{M}^{\star}_{22}\|_{\mathrm{F}}^2 \le 64\epsilon^2 \frac{M_{\star}^4}{\sigma_{\star}^4}.$$

Proof of Lemma A.4. For $n \in \mathbb{N}$, define $h_{\star,n} = \langle f_{\star,n}, g_{\star,n}\rangle$ and $\hat{h}_n = \langle \hat{f}_n, \hat{g}_n\rangle$, where $f_{\star,n}, \hat{f}_n, \hat{g}_n, g_{\star,n}$ are simple functions (i.e. finite range, see the proof of Lemma A.1) converging to $f_\star, \hat{f}, g_\star, \hat{g}$. Define

$$\begin{aligned}
\sigma_{\star,n}^2 &= \sigma_p(\mathbb{E}_{\mathcal{D}_{\mathcal{W},1}}[f_{\star,n}(w) f_{f_\star,n}(w)^\top])\sigma_p\left(\mathbb{E}_{\mathcal{D}_{\mathcal{V},1}} g_{\star,n}(v) g_{\star,n}(v)^\top]\right),\\
M_{\star,n}^2 &= \max_{i,j\neq(2,2)} \mathbb{E}_{\mathcal{D}_{i\otimes j}}[h_\star(w,v)^2]\\
\epsilon_n^2 &= \max_{i,j\neq(2,2)} \mathcal{R}(\hat{h}_n; \mathcal{D}_{i\otimes j}).
\end{aligned}$$

By the dominated convergence theorem[7],

$$\liminf_{n\geq 1} \sigma_{\star,n}^2 \geq \sigma_\star^2, \quad \limsup_{n\geq 1} M_{\star,n}^2 \leq M_\star^2, \quad \limsup_{n\geq 1} \epsilon_n^2 \leq \epsilon^2.$$

In particular, as $\epsilon^2 \leq \sigma^2/4$, then applying Lemma A.3 for n sufficiently large,

$$\mathcal{R}(\hat{h}_n; \mathcal{D}_{2\otimes 2}) \leq 64\frac{M_{\star,n}^4}{\sigma_{\star,n}^4}\epsilon_n^2.$$

Indeed, for any fixed n, all of $\hat{f}_n, \hat{g}_n, f_{\star,n}, g_{\star,n}$ are simple functions, so we can partition $\mathcal{W}$ and $\mathcal{V}$ into sets on which these embeddings are constant, and thus treat $\mathcal{W}$ and $\mathcal{V}$ as finite domains; this enables the application of Lemma A.3 applies. Finally, using the dominated covergence theorem one last time,

$$\mathcal{R}(\hat{h}; \mathcal{D}_{2\otimes 2}) = \lim_{n\to\infty} \mathcal{R}(\hat{h}_n; \mathcal{D}_{2\otimes 2}) \leq \limsup_{n\geq 1} 64\frac{M_{\star,n}^4}{\sigma_{\star,n}^4}\epsilon_n^2 \leq 64\frac{M_\star^4}{\sigma_\star^4}\epsilon^2.$$

□

We can now conclude the proof of our proposition.

Proof of Theorem 1. As $\mathcal{D}_{\text{test}} \ll_\kappa \sum_{i,j} \mathcal{D}_{i\otimes j}$ and $\sum_{i,j\neq(2,2)} \mathcal{D}_{i\otimes j} \ll_\kappa \mathcal{D}_{\text{train}}$, Lemma A.1 and additivity of the integral implies

$$\begin{aligned}
\mathcal{R}(\hat{h}; \mathcal{D}_{\text{test}}) &\leq \kappa\mathcal{R}(\hat{h}; \mathcal{D}_{2\otimes 2}) + \kappa \sum_{(i,j)\neq(2,2)} \mathcal{R}(\hat{h}; \mathcal{D}_{i\otimes j})\\
&\leq \kappa\mathcal{R}(\hat{h}; \mathcal{D}_{2\otimes 2}) + \kappa^2\mathcal{R}(\hat{h}; \mathcal{D}_{\text{train}}). \qquad \text{(A.10)}
\end{aligned}$$

Moreover, setting $\epsilon^2 := \kappa\mathcal{R}(\hat{h}; \mathcal{D}_{\text{train}})$, we have

$$\max_{(i,j)\neq(2,2)} \mathcal{R}(\hat{h}; \mathcal{D}_{i\otimes j}) \leq \sum_{(i,j)\neq(2,2)} \mathcal{R}(\hat{h}; \mathcal{D}_{i\otimes j}) \leq \kappa\mathcal{R}(\hat{h}; \mathcal{D}_{\text{train}}) := \epsilon^2.$$

Thus, for $\mathcal{R}(\hat{h}; \mathcal{D}_{\text{train}}) < \frac{\sigma_\star^2}{4\kappa}$, Eq. (A.8) holds and thus Lemma A.4 entails

$$\mathcal{R}(\hat{h}; \mathcal{D}_{2\otimes 2}) \leq 64\epsilon^2\frac{M_\star^4}{\sigma_\star^4} = 64\kappa\mathcal{R}(\hat{h}; \mathcal{D}_{\text{train}})\frac{M_\star^4}{\sigma_\star^4}.$$

Thus, combining with Eq. (A.10),

$$\mathcal{R}(\hat{h}; \mathcal{D}_{\text{test}}) \leq \kappa^2\mathcal{R}(\hat{h}; \mathcal{D}_{\text{train}}) \cdot \left(1 + 64\frac{M_\star^4}{\sigma_\star^4}\right),$$

completing the proof. □

[7]Via standard arguments, one can construct the limiting embeddings $f_{\star,n}, \hat{f}_n, \hat{g}_n, g_{\star,n}$ in such a way that their norms are dominated by integrable functions.

A.6 Extrapolation for Transduction

Leveraging Theorem 1, this section provides a formal theoretical justification for predictors of the form Eq. (A.5).

We begin by stipulating the requisite conditions. First, we require *well-specification*: that $h_\star(\cdot)$ can also be expressed in the form Eqs. (A.1) and (A.5); to ensure $\bar{h}_\star(\cdot,\cdot)$ is well-defined as a deterministic predictor (whereas $h_{\boldsymbol{\theta}}(\cdot)$ need not be), we need the following, rather strong condition on $h_\star(\cdot)$.

Assumption A.1. We assume that $h_\star$ is *bilinearly transducible*; that is, there exists $f_{\star,k} : \Delta\mathcal{X} \to \mathbb{R}^p$ and $g_{\star,k} : \mathcal{X} \to \mathbb{R}^p$ such that for all $x \in \mathcal{X}$, the following holds with probability 1 over $x' \sim \mathcal{D}_{\mathrm{trns}}(x)$:

$$h_{\star,k}(x) = \bar{h}_{\star,k}(\Delta x, x') := \langle f_{\star,k}(\Delta x), g_{\star,k}(x')\rangle, \quad \text{where } \Delta x = x - x'.$$

Assumption A.1 means that for any feature x, any feature x' in the support of $\mathcal{D}_{\mathrm{trns}}(x)$ be transduced to x via bilinear embeddings.

Next, our theory requires that the distributions $\bar{\mathcal{D}}_{\mathrm{train}}, \bar{\mathcal{D}}_{\mathrm{test}}$ defined in Eqs. (A.2) and (A.4) satisfy the notion of combinatorial support given in the previous section.

Definition A.3 (Bounded combinatorial density ratio, specialized to transduction)**.** We say that $\bar{\mathcal{D}}_{\mathrm{test}}$ has κ*-bounded combinatorial density ratio* with respect to $\bar{\mathcal{D}}_{\mathrm{train}}$, written as $\bar{\mathcal{D}}_{\mathrm{test}} \overset{\mathrm{comb}}{\ll_\kappa} \bar{\mathcal{D}}_{\mathrm{train}}$, if it abides by Definition A.2. That is, there exists distributions $\mathcal{D}_{\Delta\mathcal{X},i}$ and $\mathcal{D}_{\mathcal{X},j}$, $i, j \in \{1, 2\}$, over $\Delta\mathcal{X}$ and $\mathcal{X}$, respectively, such that $\mathcal{D}_{i\otimes j} := \mathcal{D}_{\Delta\mathcal{X},i} \otimes \mathcal{D}_{\mathcal{X},j}$ satisfy

$$\sum_{(i,j)\neq(2,2)} \mathcal{D}_{i\otimes j} \ll_\kappa \bar{\mathcal{D}}_{\mathrm{train}}, \quad \text{and } \bar{\mathcal{D}}_{\mathrm{test}} \ll_\kappa \sum_{i,j=1,2} \mathcal{D}_{i\otimes j},$$

where we recall the definition of κ-bounded density ratio notation $\ll_\kappa$ in Definition A.1.

Let us recall the discussion of Remark A.1. Building upon Definition A.1, Definition A.3 introduces a notion of bounded density ratio between $\bar{\mathcal{D}}_{\mathrm{train}}$ and $\bar{\mathcal{D}}_{\mathrm{test}}$ in the OOC setting. Take the discrete case of matrix completion as an example, as illustrated in Fig 9, the training distribution of $(\Delta x, x')$ covers the support of the $(1, 1)$, $(1, 2)$, $(2, 1)$ blocks of the matrix, while the testing distribution of $(\Delta x, x')$ might be covered by any product of the marginals of the 2×2 blocks. With this connection in mind, it is possible to establish the OOC guarantees on $\bar{\mathcal{D}}_{\mathrm{test}}$ as in matrix completion tasks, if the bilinear embedding admits some low-rank structure.

Theorem 2. *Suppose that $h_\star$ is bilinearly transducible (Assumption A.1), $h_{\boldsymbol{\theta}}$ takes the form of Eqs. (A.1) and (A.5), and for each $k \in [K]$, the embeddings $f_{\star,k}, f_{\boldsymbol{\theta},k}, g_{\star,k}, g_{\boldsymbol{\theta},k}$ are all of dimension p. Further, suppose there exist $\kappa \geq 1$ and $M \geq \sigma > 0$ such that $\bar{\mathcal{D}}_{\mathrm{test}} \overset{\mathrm{comb}}{\ll_\kappa} \bar{\mathcal{D}}_{\mathrm{train}}$, and for all $k \in [K]$,*

$$\sigma_p(\mathbb{E}_{\mathcal{D}_{\Delta\mathcal{X},1}}[f_{\star,k}f_{\star,k}^\top])\sigma_p(\mathbb{E}_{\mathcal{D}_{\mathcal{X},1}}[g_{\star,k}g_{\star,k}^\top]) \geq \sigma^2, \quad \sup_{\Delta x, x'} |\bar{h}_{\star,k}(\Delta x, x')| \leq M. \tag{A.11}$$

Finally, suppose the loss $\ell(\cdot,\cdot)$ is the square loss. Then, if $\mathcal{R}(h_{\boldsymbol{\theta}}; \mathcal{D}_{\mathrm{train}}) \leq \frac{\sigma^2}{4\kappa}$, we have

$$\mathcal{R}(h_{\boldsymbol{\theta}}; \mathcal{D}_{\mathrm{test}}) \leq \mathcal{R}(h_{\boldsymbol{\theta}}; \mathcal{D}_{\mathrm{train}}) \cdot \kappa^2 \left(1 + 64\frac{M^4}{\sigma^4}\right) = \mathcal{R}(h_{\boldsymbol{\theta}}; \mathcal{D}_{\mathrm{train}}) \cdot \mathrm{poly}(\kappa, \frac{M}{\sigma}).$$

The additional conditions of Theorem 2 beyond those stated in Assumption A.1 and Definition A.3 are discussed at the end of the section.

Proof of Theorem 2. We argue by reducing to Theorem 1. The parameterization of the stochastic predictor $h_{\boldsymbol{\theta}}$ in Eq. (A.1), followed by Assumption A.1 allows us to write

$$\begin{aligned}
\mathcal{R}(h_{\boldsymbol{\theta}}; \mathcal{D}_{\mathrm{train}}) &= \mathbb{E}_{x\sim\mathcal{D}_{\mathrm{train}}}\mathbb{E}_{y\sim h_{\boldsymbol{\theta}}(x)}\ell(y, h_\star(x)) \\
&= \mathbb{E}_{x\sim\mathcal{D}_{\mathrm{train}}}\mathbb{E}_{x'\sim\mathcal{D}_{\mathrm{trns}}(x)}\ell(\bar{h}_{\boldsymbol{\theta}}(x - x', x'), h_\star(x)) \\
&= \mathbb{E}_{x\sim\mathcal{D}_{\mathrm{train}}}\mathbb{E}_{x'\sim\mathcal{D}_{\mathrm{trns}}(x)}\ell(\bar{h}_{\boldsymbol{\theta}}(x - x', x'), \bar{h}_\star(x - x', x')).
\end{aligned}$$

In the above display, the joint distribution of $(x - x', x')$ is precisely given by $\bar{\mathcal{D}}_{\text{train}}$ (see Eq. (A.2)). Hence,

$$\mathcal{R}(h_{\boldsymbol{\theta}}; \mathcal{D}_{\text{train}}) = \mathbb{E}_{\bar{\mathcal{D}}_{\text{train}}} \ell(\bar{h}_{\boldsymbol{\theta}}(\Delta x, x'), \bar{h}_{\star}(\Delta x, x')).$$

Further, as $\ell(y, y') = \|y - y'\|^2$ is the square loss and decomposes across coordinates,

$$\mathcal{R}(h_{\boldsymbol{\theta}}; \mathcal{D}_{\text{train}}) = \sum_{k=1}^{K} \mathbb{E}_{\bar{\mathcal{D}}_{\text{train}}} (\bar{h}_{\boldsymbol{\theta},k}(\Delta x, x') - \bar{h}_{\star,k}(\Delta x, x'))^2. \tag{A.12}$$

By the same token,

$$\mathcal{R}(h_{\boldsymbol{\theta}}; \mathcal{D}_{\text{test}}) = \sum_{k=1}^{K} \mathbb{E}_{\bar{\mathcal{D}}_{\text{test}}} (\bar{h}_{\boldsymbol{\theta},k}(\Delta x, x') - \bar{h}_{\star,k}(\Delta x, x'))^2.$$

To conclude the proof, we remain to show that for all $k \in [K]$, we have

$$\begin{aligned} &\mathbb{E}_{\bar{\mathcal{D}}_{\text{test}}} (\bar{h}_{\boldsymbol{\theta},k}(\Delta x, x') - \bar{h}_{\star,k}(\Delta x, x'))^2 \le C_{\text{prob}} \cdot \mathbb{E}_{\bar{\mathcal{D}}_{\text{train}}} (\bar{h}_{\boldsymbol{\theta},k}(\Delta x, x') - \bar{h}_{\star,k}(\Delta x, x'))^2, \\ &\quad \text{where } C_{\text{prob}} = \kappa^2 \left(1 + 64 \frac{M^4}{\sigma^4}\right). \end{aligned} \tag{A.13}$$

Indeed, for each $k \in [K]$, we have

$$\mathbb{E}_{\bar{\mathcal{D}}_{\text{train}}} (\bar{h}_{\boldsymbol{\theta},k}(\Delta x, x') - \bar{h}_{\star,k}(\Delta x, x'))^2 \overset{\text{(Eq. (A.12))}}{\le} \mathcal{R}(h_{\boldsymbol{\theta}}; \mathcal{D}_{\text{train}}) \overset{\text{(by assumption)}}{\le} \frac{\sigma^2}{4\kappa}.$$

Hence Eq. (A.13) holds by invoking Theorem 1 with the correspondences $\mathcal{W} \leftarrow \Delta\mathcal{X}$, $\mathcal{V} \leftarrow \mathcal{X}$, $\sigma_\star \leftarrow \sigma$, $M_\star \leftarrow M$ and $\kappa \leftarrow \kappa$. This concludes the proof. $\square$

Remarks on additional conditions. We comment on the three additional conditions of Theorem 2.

- The singular value condition, $\sigma_p(\mathbb{E}_{\mathcal{D}_{\Delta\mathcal{X},1}}[f_{\star,k} f_{\star,k}^\top]) \cdot \sigma_p(\mathbb{E}_{\mathcal{D}_{\mathcal{X},1}}[g_{\star,k} g_{\star,k}^\top])) \ge \sigma^2 > 0$, mirrors non-degeneracy conditions given in past work in matrix completion (c.f. [17]).
- The support condition $\sup_{\Delta x, x'} |\bar{h}_{\star,k}(\Delta x, x')| \le M$ is a mild boundedness condition, which (in light of Theorem 1) can be weakened further to

$$\max_{1 \le i,j \le 2} \mathbb{E}_{\mathcal{D}_{i \otimes j}} [\bar{h}_{\star,k}(\Delta x, x')^2] \le M^2,$$

  where $\mathcal{D}_{i \otimes j}$ are the contituent distributions witnessing $\bar{\mathcal{D}}_{\text{test}} \overset{\text{comb}}{\ll_\kappa} \bar{\mathcal{D}}_{\text{train}}$.
- The final condition, $\mathcal{R}(h_{\boldsymbol{\theta}}; \mathcal{D}_{\text{train}}) \le \frac{\sigma^2}{4\kappa}$, is mostly for convenience. Indeed, as $M \ge \sigma$ and $\kappa \ge 1$, then as soon as $\mathcal{R}(h_{\boldsymbol{\theta}}; \mathcal{D}_{\text{train}}) > \frac{\sigma^2}{4\kappa}$, our upper-bound on $\mathcal{R}(h_{\boldsymbol{\theta}}; \mathcal{D}_{\text{test}})$ is no better than

$$\kappa M^2 \cdot \frac{64}{4} \cdot \frac{M^2}{\sigma^2} \ge 6M^2,$$

  which is essentially vacuous. Indeed, if we also inflate M and stipulate that $\sup_{\Delta x, x'} |\bar{h}_{\boldsymbol{\theta}}(\Delta x, x')| \le \sqrt{6}M$, we can remove the condition $\mathcal{R}(h_{\boldsymbol{\theta}}; \mathcal{D}_{\text{train}}) \le \frac{\sigma^2}{4\kappa}$ altogether.