Predicting is not Understanding: Recognizing and Addressing Underspecification in Machine Learning

Damien Teney[1,3] Maxime Peyrard[2] Ehsan Abbasnejad[3]

[1]Idiap Research Institute [2]EPFL [3]Australian Inst. for Machine Learning
firstname.lastname@{idiap.ch,epfl.ch,adelaide.edu.au}

Abstract. Machine learning (ML) models are typically optimized for their accuracy on a given dataset. However, this predictive criterion rarely captures all desirable properties of a model, in particular how well it matches a domain expert's *understanding* of a task. Underspecification [12] refers to the existence of multiple models that are indistinguishable in their in-domain accuracy, even though they differ in other desirable properties such as out-of-distribution (OOD) performance. Identifying these situations is critical for assessing the reliability of ML models. We formalize the concept of underspecification and propose a method to identify and partially address it. We train multiple models with an independence constraint that forces them to implement different functions. They discover predictive features that are otherwise ignored by standard empirical risk minimization (ERM), which we then distill into a global model with superior OOD performance. Importantly, we constrain the models to align with the data manifold to ensure that they discover meaningful features. We demonstrate the method on multiple datasets in computer vision (collages, WILDS-Camelyon17, GQA) and discuss general implications of underspecification. Most notably, in-domain performance cannot serve for OOD model selection without additional assumptions.[1]

1 Introduction

Is data all you need? A finite set of i.i.d. examples is almost never sufficient to learn a task. Inductive biases have long been known to be necessary for in-domain generalization [49,82]. OOD[2] generalization complicates things further since one also needs to determine which predictive patterns of the training data will remain relevant at test time. Correlations between inputs and labels that are important for the task may be indistinguishable from spurious ones that result from dataset-specific artefacts such as selection biases.

An example in image recognition. Image labels are often correlated with objects and the backgrounds they appear in (*e.g.* cars in cities, birds in nature). Recognizing either often suffice to predict correct labels. However, robust OOD

[1] See https://arxiv.org/abs/2207.02598 for the full-length version of this work.
[2] In this paper, OOD means there is a covariate shift between training and test data [67].

generalization (*e.g.* correctly labeling images of birds in street scenes) requires to rely on shapes and to ignore the background. When this requirement cannot be deduced from the data (because both features leave a similar signature in the joint training distribution), the task is said to be underspecified. In this example, the task requires the additional knowledge that labels refer to object shapes rather than background textures [20]. Such knowledge is often task-specific. For example, the opposite choice of prioritizing color or texture over shape would be sensible for recognizing traffic signs or segmenting medical images.

Underspecification gap: the difference between the information provided in a dataset and the information required to perform *as desired* on a task.

The qualifier "*as desired*" captures the fact that different use cases require different properties such as adversarial robustness, interpretability, fairness, or OOD generalization. The latter is the focus of this paper. Underspecification arises because these properties do not necessarily correlate with the ERM objective [77] typically used to train models.

This paper argues that **identifying underspecification** is important for assessing the reliability of ML models, their reliance on hidden assumptions, and for identifying the information missing for OOD robustness. We identify underspecification by **discovering multiple understandings** of the data. We learn multiple predictive models compatible with a given dataset and hypothesis class (low in-domain risk). We force them to rely on different predictive features by encouraging orthogonality of their input gradients. We also ensure that these features remain semantically meaningful by constraining the input gradients to the data manifold. Training multiple models stands in contrast with the standard practice of optimizing a single solution to a learning problem – which hides the existence of underspecification. With our method, we discover predictive features otherwise ignored by standard ERM. This alone produces candidate models with superior OOD performance. In addition, we show how to distill selected features from multiple candidate models into one that is robust across a wider range of distribution shifts. In all cases, a selection strategy must be provided (see Section C) such as an OOD validation set, domain expertise, task-specific heuristics, etc.

Experiments. We apply the method to controlled data (collages [66,72]) and computer vision benchmarks (WILDS-Camelyon17 [41], GQA [32,37]). On visual question answering, we show that multiple models can produce similar answers while relying on different visual features (Figure 1).

Implications. Our work complements other studies [12,47] in formalizing underspecification as a root cause of multiple challenges in ML including shortcut learning, distribution shifts, and even adversarial vulnerabilities (an extreme case of OOD inputs). Our formalization of underspecification makes it obvious that ID and OOD performance are not necessarily coupled. Therefore, without further assumptions, in-domain validation performance is not a reliable model selection strategy for OOD performance despite contradictory suggestions made in the literature [24,48]. The prevalence of underspecification [12] also suggests that

Fig. 1: Example of underspecification in visual question answering. Our method trains multiple models that each discover different predictive features. We obtain three models producing identical answers on most training and validation data, even though they rely on different visual clues (evidenced by grad-CAM visualizations over object proposals [64]). Each model reflects a **different understanding of the task** compatible with the data (possibly incomprehensible to humans) which reveals ambiguity in its specification.

task-specific knowledge and assumptions are often necessary to build robust ML models, since they cannot emerge from simply scaling up data and architectures. We summarize our contributions as follows. See Appendix A for related work.

1. We propose a mathematical framework for quantifying and addressing underspecification.
2. We derive a method to learn a set of models compatible with a given dataset that exhibit distinct OOD behaviour. We force the models to rely on different features (independence objective) that are nonetheless semantically meaningful (on-manifold constraint).
3. We use the method for (1) highlighting underspecification in given dataset/architecture pairs, and (2) building models with superior OOD performance on collages [66,72], WILDS-Camelyon17 [41], and GQA [32,37].

2 Formalizing underspecification

Let us focus on binary classification tasks. A dataset provides **labeled examples** $\mathcal{D}_{\text{tr}} = \{(\boldsymbol{x}_i, y_i)\}_i$ with $\boldsymbol{x} \in \mathbb{R}^{d_{\text{in}}}$, $y_i \in \{0,1\}$. The goal of a learning algorithm is to identify a **predictor** $f : \mathbb{R}^{d_{\text{in}}} \to \mathbb{R}$ to estimate labels[3] of examples from a test set $\mathcal{D}_{\text{test}} = \{\boldsymbol{x}_i\}_i$. While the input data $\boldsymbol{x}$ is typically high-dimensional (*e.g.* vectorized images), natural data (*e.g.* photographs) occupies only a fraction of the input space assumed to form a low-dimensional **manifold** [81] $\mathcal{M} \subset \mathbb{R}^{d_{\text{in}}}$. The dimensionality d_{manifold} ($< d_{\text{in}}$) is known as the **intrinsic dimensionality** of the data. Training and test data are drawn from a distribution on this manifold P_{ID} (in-domain examples) while unbiased natural data (free of dataset-specific selection biases) is drawn from a distribution P_{OOD} of typically broader support.

[3] We define f to output logits. A binary prediction $\hat{y}$ is obtained as $\hat{y} = \text{round}\big(\sigma\big(f(\boldsymbol{x})\big)\big)$.

Inductive biases are the properties of a learning algorithm that determine what model $f_{\theta^\star}$ is returned for a dataset $\mathcal{D}$ from a hypothesis class $\mathcal{H} = \{f_{\boldsymbol{\theta}}, \forall\, \boldsymbol{\theta}\}$ where $f_{\boldsymbol{\theta}}$ is a model with free parameters θ. Inductive biases enable generalization from finite data [49] by encoding assumptions on the relation between $\mathcal{D}$ and $\mathcal{D}_{\text{test}}$. In particular, classical learning theory assumes that $\mathcal{D}$ and $\mathcal{D}_{\text{test}}$ contain i.i.d. samples from the same distribution. For completeness, we summarize a standard training workflow.

1. Randomly split the data into training and validation sets: $\mathcal{D} = \mathcal{D}_{\text{tr}} \cup \mathcal{D}_{\text{val}}$.
2. A **hypothesis class** $\mathcal{H} = \{f_{\boldsymbol{\theta}}, \forall\, \boldsymbol{\theta}\}$ is chosen *e.g.* by defining a neural architecture f.
3. **Empirical risk minimization** serves to optimize the free parameters of f as $\boldsymbol{\theta}^\star = \text{argmin}_{\boldsymbol{\theta}}\ \mathcal{R}\left(f_{\boldsymbol{\theta}}, \mathcal{D}_{\text{tr}}\right)$ where the empirical risk is defined as $\mathcal{R}(f, \mathcal{D}) = \Sigma_{(\boldsymbol{x},y)\in\mathcal{D}}\ \mathcal{L}_{\text{pred}}\big(y, \sigma\big(f(\boldsymbol{x})\big)\big)\ /\ |\mathcal{D}|$,
and $\mathcal{L}_{\text{pred}}$ is a predictive loss such as binary cross-entropy.
4. **Validation performance** serves to refine various choices (architecture, regularizers, …) by trial and error, *i.e.* loosely solving $f'_{\boldsymbol{\theta}^{\star\prime}} = \text{argmin}_{f,\ldots}\ \mathcal{R}(f_{\boldsymbol{\theta}^\star}, \mathcal{D}_{\text{val}})$ where $\mathcal{R}$ is often substituted with a task-specific metric such as the error rate.

There is often a multitude of models satisfying the above procedure, not all are equally desirable because they differ in properties that the procedure does not constrain. The degree of underspecification indicates the importance of arbitrary and stochastic factors in the outcome of the learning process.

This paper focuses on **differences in OOD performance** among predictive models. OOD performance is the predictive performance of a model (in terms of risk, accuracy, or another task-specific metric) on test data drawn from a distribution $\text{P}_{\text{OOD}} \neq \text{P}_{\text{ID}}$. On OOD data, features that were predictive in the training data may become irrelevant or misleading, causing a drop in performance of a model that relies on them. By definition, OOD performance is underspecified by the ERM objective, since the empirical risk is estimated on in-domain data.

To capture variability in OOD performance, we propose a definition of underspecification based on the number of ways to fit the data with the above procedure and produce different OOD predictions.[4]

Definition 1. *The* ***degree of underspecification*** *of a dataset* $\mathcal{D} = \mathcal{D}_{\text{tr}} \cup \mathcal{D}_{\text{val}}$*, input manifold* $\mathcal{M}$*, and hypothesis class* $\mathcal{H} = \{f_{\boldsymbol{\theta}}, \forall\, \boldsymbol{\theta}\}$ *is the ratio of volumes* $vol(\mathcal{H}')/\, vol(\mathcal{H})$ *of the largest subset of models* $\mathcal{H}' \subset \mathcal{H}$ *such that its elements* $\{f_{\boldsymbol{\theta}_m}\}_m$ *all have, for small constants* ϵ_{tr}*,* ϵ_{val}*:*

- *A low training risk:* $\mathcal{R}(f_{\boldsymbol{\theta}_m}, \mathcal{D}_{\text{tr}}) < \epsilon_{tr},\ \forall\, f_{\boldsymbol{\theta}} \in \mathcal{H}'$,
- *A low validation risk:* $\mathcal{R}(f_{\boldsymbol{\theta}_m}, \mathcal{D}_{\text{val}}) < \epsilon_{val},\ \forall\, f_{\boldsymbol{\theta}} \in \mathcal{H}'$,
- *Distinct OOD predictions:* $P\big(\text{round}(\sigma(f_{\boldsymbol{\theta}1}(\boldsymbol{x})) \neq \text{round}(\sigma(f_{\boldsymbol{\theta}2}(\boldsymbol{x}))\big) \approx 1$, $\forall\ f_{\boldsymbol{\theta}1}, f_{\boldsymbol{\theta}2} \in \mathcal{H}',\quad f_{\boldsymbol{\theta}1} \neq f_{\boldsymbol{\theta}2},\quad \boldsymbol{x} \sim P_{\text{OOD}}$.

The next section derives a method to learn a set of models with these properties.

[4] Previously, [31,65] used volumes of hypothesis spaces to define Rashomon sets.

3 Proposed method

Overview. We train multiple models with the same architecture and data while enforcing them to represent different functions and use different features. The models use different initializations, but this does not always suffice to produce significantly-different models. We add two regularizers that enforce (1) independence of the models (mutually-orthogonal input gradients) and (2) alignment with the data manifold such that the models learn meaningful features.

Since the constraints follow from Definition 1, the number of models trainable to satisfy them indicates the degree of underspecification. The only existence of multiple such models thus highlights cases of underspecification. The models also discover some predictive features missed by standard ERM, which can be combined by distillation into a predictor with superior OOD performance. In the next sections, we implement the two constraints as differentiable regularizers.

3.1 Independent models

To optimize for distinct OOD predictions, we turn the criteria of Definition 1 into a differentiable objective using the concept of independent models [60,61].

Definition 2. *A pair of predictors* $f_{\boldsymbol{\theta}_1}$, $f_{\boldsymbol{\theta}_2}$ *are* ***locally independent*** *at* $\boldsymbol{x}$ *iff their predictions are statistically independent for Gaussian perturbations around* $\boldsymbol{x}$*:* $f_{\boldsymbol{\theta}_1}(\widetilde{\boldsymbol{x}}) \perp f_{\boldsymbol{\theta}_2}(\widetilde{\boldsymbol{x}})$, $\widetilde{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{x}, \sigma \boldsymbol{I})$.

Definition 3. *A set of predictors* $\{f_{\boldsymbol{\theta}_1}, .., f_{\boldsymbol{\theta}_M}\}$ *are* ***globally independent*** *on a dataset* $\mathcal{D}$ *iff every pair of them are locally independent at every* $\boldsymbol{x} \in \mathcal{D}$.

This formalizes the notion that models can rely on different features. In our case, we seek a set of models globally independent from one another. We obtain a tractable objective using the relation between statistical independence and geometric orthogonality developed in [61].

Proposition 1. *A pair of predictors* $f_{\boldsymbol{\theta}_1}$, $f_{\boldsymbol{\theta}_2}$ *are locally independent at* $\boldsymbol{x}$ *iff the mutual information* $MI(f_{\boldsymbol{\theta}_1}(\widetilde{\boldsymbol{x}}), f_{\boldsymbol{\theta}_2}(\widetilde{\boldsymbol{x}})) = 0$ *with* $\widetilde{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{x}, \sigma \boldsymbol{I})$.

For infinitesimally small perturbations ($\sigma \to 0$), samples $\widetilde{\boldsymbol{x}}$ can be approximated through linearization by the input gradients $\nabla_{\boldsymbol{x}} f$. These are 1D Gaussian random variables whose correlation is given by $\cos\left(\nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}_1}(x), \nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}_2} x\right)$. Their mutual information [23] is $-\frac{1}{2}\ln\left(1 - \cos^2\left(\nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}_1}(x), \nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}_2}(x)\right)\right)$. Therefore, the statistical independence between the models' outputs as their inputs are perturbed by small Gaussian variations can be enforced by making their input gradients orthogonal. Our **local independence loss** for a pair of models is:

$$\mathcal{L}_{\text{indep}}\left(\nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}_{m_1}}(\boldsymbol{x}),\ \nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}_{m_2}}(\boldsymbol{x})\right) \;=\; \cos^2\left(\nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}_{m_1}}(\boldsymbol{x}),\ \nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}_{m_2}}(\boldsymbol{x})\right) \qquad (1)$$

with $\cos^2(\boldsymbol{v}, \boldsymbol{w}) = (\boldsymbol{v}^\intercal \boldsymbol{w})^2 \,/\, (\boldsymbol{v}^\intercal \boldsymbol{v})(\boldsymbol{w}^\intercal \boldsymbol{w})$. To enforce *global* independence, this loss will be applied to all training points and pairs of models in Eq. (3).

3.2 On-manifold constraint

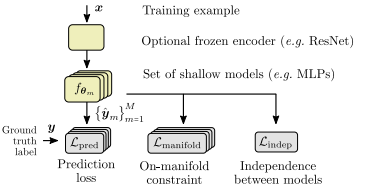


Fig. 2: Method overview during training.

The independence constraint (1) makes models' input gradients orthogonal to one another. The number of models satisfying it grows exponentially with the input dimension (d_{in}) but many are practically irrelevant because the natural data manifold usually occupies much fewer dimensions. Intuitively, when the constraint affects a model's gradients in dimensions pointing outward the manifold, it does not affect its predictions on natural data. Consequently, the independence constraint could be satisfied by models that produce identical predictions on every natural input (thus defeating its purpose) because their decision boundaries are identical when projected on the manifold. The issue stems from the *isotropic* perturbations in Eq. (1). Only perturbations *on* the manifold are meaningful.

One straightforward solution would be to enforce independence after projecting the data on a learned approximation of the manifold. This approach, proposed in [60,61] failed in our early experiments because of the difficulty of optimizing the independence objective under such a strict on-manifold constraint. Instead, we implement a soft constraint as a regularizer that proved easy to train and resilient to imperfect models of the manifold.

To learn the data manifold $\mathcal{M}$, we need unlabeled examples, ideally containing the type of OOD data expected at test time *e.g.* a broad collection of natural images: $\mathcal{D}_{\text{OOD}} = \{\boldsymbol{x}_i\} \sim \text{P}_{\text{OOD}}$. We use this data off-line to prepare a function $\text{proj}_{\mathcal{M}}(\boldsymbol{x}, \boldsymbol{v})$ that projects an arbitrary vector $\boldsymbol{v}$ at $\boldsymbol{x}$ in the input space onto the manifold (Figure 3). During training, we penalize each model with the distance between its input gradients and their projection on the manifold. The **on-manifold loss** is defined as

$$\mathcal{L}_{\text{manifold}}\big(\nabla f(\boldsymbol{x})\big) = \Big|\Big|\text{proj}_{\mathcal{M}}\big(\boldsymbol{x}, \nabla_{\boldsymbol{x}} f(\boldsymbol{x})\big) - \nabla_{\boldsymbol{x}} f(\boldsymbol{x})\Big|\Big|_2^2. \tag{2}$$

We describe possible implementations of $\text{proj}_{\mathcal{M}}(\cdot)$ in Appendix D with a variational auto-encoder (VAE) or a simple PCA. In summary, the on-manifold loss encourages a model to be sensitive to variations in the input that are likely to be encountered in natural test data. It typically has no effect on in-domain performance (Figure 6c) since it only removes a model's sensitivity to unnatural inputs such as variations of isolated pixels unlikely to appear in natural images.

The overall learning objective combines the predictive, independence, and on-manifold losses:

$$\begin{aligned}\mathcal{L}(\mathcal{D}_{\text{tr}}, \boldsymbol{\theta}_1 \dots \boldsymbol{\theta}_M) = \Sigma_{\boldsymbol{x}\in\mathcal{D}_{\text{tr}}}\Big[& \quad (1/M) \quad \Sigma_{m=1}^M \mathcal{L}_{\text{pred}}\big(y,\, \sigma(f_{\boldsymbol{\theta}_m}(\boldsymbol{x}))\big) \\ & + (1/M^2)\ \Sigma_{m_1=1}^M \Sigma_{m_2=1}^M \lambda_{\text{indep}}\ \ \mathcal{L}_{\text{indep}}\big(\nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}_{m_1}}(\boldsymbol{x}), \nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}_{m_2}}(\boldsymbol{x})\big) \\ & + (1/M) \quad \Sigma_{m=1}^M \lambda_{\text{manifold}}\ \ \mathcal{L}_{\text{manifold}}\big(\nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}_m}(\boldsymbol{x})\big) \ \Big].\end{aligned} \tag{3}$$

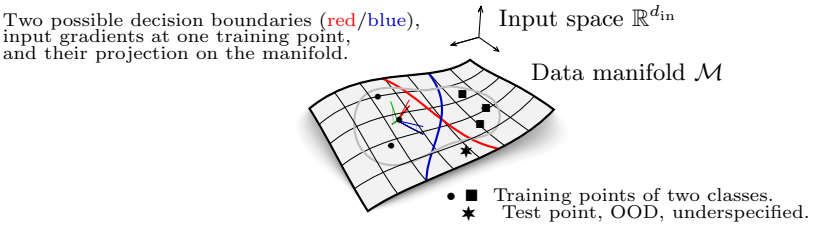


Fig. 3: Effect of the proposed method in input space. Data such as natural images is assumed to lie on a low-dimensional manifold. The training set covers a subset of this manifold (gray ellipse). OOD test data (★) lies outside this subset. In this example, our method discovers two models (red and blue decision boundaries) whose input gradients are orthogonal (shown at one training point, in colors matching the boundary). Even though a third model (green vector) could satisfy the orthogonality constraint, its input gradient would point outside the manifold. This would violate the *on-manifold* constraint, which requires gradients to closely match their projection on the manifold.

3.3 Fine-tuning

After training a set of models with (3), we propose to relax the independence and on-manifold constraints ($\lambda_{\text{indep}} \leftarrow 0$, $\lambda_{\text{manifold}} \leftarrow 0$) then fine-tune the models. This eases the optimization and typically allows the models to reach a higher predictive accuracy. Concretely, we apply binary masks on the data such that each model is fine-tuned only on the elements most relevant to itself:[5]

$$\mathcal{D}_{\text{tr}}^m = \{(\boldsymbol{x}_i \odot \mathbf{mask}_i^m, y_i) : (\boldsymbol{x}_i, y_i) \in \mathcal{D}_{\text{tr}}\} \tag{4}$$

with $\mathbf{mask}_i^m \in \{0,1\}^{d_{\text{in}}}$. They are computed before starting the fine-tuning to highlight the data most relevant to each model. Each element (pixel, channel) is unmasked only for the model with the largest corresponding gradient magnitude:

$$\mathbf{mask}_i^m = \mathbb{1}\big(m = \underset{1 \leqslant m \leqslant M}{\operatorname{argmax}} \nabla f_{\boldsymbol{\theta}_m}(\boldsymbol{x}_i)\big) \quad \forall\, (\boldsymbol{x}_i, \cdot) \in \mathcal{D}_{\text{tr}}. \tag{5}$$

We fine-tune each model on its own masked version of the data.[6] This ensures that the models remain distinct despite disabling the regularizers ($\lambda_{\text{indep}} \leftarrow 0$, $\lambda_{\text{manifold}} \leftarrow 0$). See Algorithm 1 for a summary.

3.4 Distilling multiple models into one

Finally, after training/fine-tuning a set of models, we propose to combine the best of them into a global one that uses all of the most relevant features.We train this global model from scratch, without regularizers, on masked data as described above, using masks from *multiple* selected models combined with a

[5] In our implementation, masked elements are not replaced with zeros, but rater with random values from other instances in the current mini-batch.

[6] We obtain very similar results between fine-tuning and retraining models from scratch on the masked data.

Algorithm 1: Training and fine-tuning models.

Inputs: Labeled examples $\mathcal{D}_{\text{tr}}$. Unlabeled examples $\mathcal{D}_{\text{OOD}}$ (typically $\mathcal{D}_{\text{tr}} \subset \mathcal{D}_{\text{OOD}}$). Architecture f.
Result: Set of independent models $\{f_{\boldsymbol{\theta}_1} \dots f_{\boldsymbol{\theta}_M}\}$.
Method:
With $\mathcal{D}_{\text{OOD}}$, estimate dimensionality d_{manifold} [57] and set the number of models $M \leftarrow d_{\text{manifold}}$.
With $\mathcal{D}_{\text{OOD}}$, prepare function $\text{proj}(\cdot)$ by PCA decomp. or by training a VAE.
With $\mathcal{D}_{\text{tr}}$, train M instances of f in parallel (Eq. 3):
$\{\boldsymbol{\theta}_1 \dots \boldsymbol{\theta}_M\} \leftarrow \text{argmin}\, \mathcal{L}(\mathcal{D}_{\text{tr}}, \boldsymbol{\theta}_1 \dots \boldsymbol{\theta}_M)$.
Determine masks on input data (Eq. 5): $\{\mathbf{mask}_i^m\}_{i,m}$
foreach m **do** // Optional fine-tuning on masked data
  $\mathcal{D}_{\text{tr}}^m \leftarrow \{(\boldsymbol{x}_i \odot \mathbf{mask}_i^m, y_i)\}_i$ // Prepare masked data
  $\lambda_{\text{indep}} \leftarrow 0,\ \lambda_{\text{manifold}} \leftarrow 0$ // Use only predictive loss
  $\boldsymbol{\theta}_m \leftarrow \text{argmin}\, \mathcal{L}(\mathcal{D}_{\text{tr}}^m, \boldsymbol{\theta}_m)$ // Fine-tune

logical OR. In our experiments, we combine the two models with the highest accuracies on an OOD validation set. We repeat this pairwise combination as long the accuracy of the global model increases, usually for 2–3 iterations (as formalized in Algorithm 2 in the Appendix).

4 Experiments

We first present experiments that validate the method on controlled data with multiple known features (collages, Section 4.1). We then demonstrate applications to existing datasets: WILDS-Camelyon17 and GQA (Sections 4.2 and J).

4.1 Experiments on controlled data: collages

This diagnostic dataset contains images with binary labels that are constructed to contain multiple predictive features [66,72]. Each image contains four tiles representing one of two classes respectively from MNIST (0/1), CIFAR-10 (automobile/truck), Fashion-MNIST (pullover/coat), and SVHN (0/1).

- At **training time**, the labels are perfectly correlated with the four tiles (0/1 respectively for the first/second possible class in each tile). There are (at least) four equally-valid ways of understanding the task (*i.e.* relying on any of the four tiles).
- At **test time**, we evaluate a model on four test sets that represent different OOD conditions. In each, only one tile is correlated with the correct label while others tiles are randomized. By examining the performance on the four test sets, we can identify which tile(s) the model relies on

Task difficulty. This dataset is surprisingly challenging because the tiles vary greatly in learning difficulty (*e.g.* MNIST 0s/1s are very distinct while

Fig. 4: Examples of collages [72]. Tr. labels are correlated with all four tiles.

Fashion-MNIST pullovers/coats look extremely similar). It would be reasonable to learn a model that relies on all four tiles. However, an ERM-trained baseline surprisingly uses only a few MNIST pixels (achieving ∼99% accuracy on the MNIST test set and ∼50% on the others), as shown in previous work on the simplicity bias of neural networks [72].

We follow [72] and use our method to learn multiple models compatible with the data. We then report the accuracy of the best model on each test set, *i.e.* the best accuracy assuming perfect model selection. This avoids confounding the performance of the learning algorithm and with that of the selection strategy.

Applying the proposed method. We follow Algorithm 1. We prepare unlabeled data to defines the data manifold as the union of the training and test sets, thus covering all combinations of contents of the four tiles. With this data, we estimate the dimensionality of the manifold with [57] as about 23.8 ($\sigma = 0.16$ over 10 runs). We prepare two generative models of the manifold: a PCA with 24 components (capturing ∼85% of the variance) and a VAE with 24 latent dimensions (details in Appendix F). We define a simple architecture (2-layer MLPs) and train multiple instances in parallel with the proposed objective. The only hyperparameters are the number of models and weights of independence/on-manifold constraints. We plot a range of values in the appendix (Figure 9).

Results. Our method learns models that focus on different parts of the images. Remarkably, learning as few as 4 models is sufficient to obtain models with high accuracy on all of four test sets. Let us examine several ablations.

- The baseline ($\lambda_{\text{indep}} = \lambda_{\text{manifold}} = 0$) only learns about MNIST.
- The independence constraint ($\lambda_{\text{indep}} >$, $\lambda_{\text{manifold}} = 0$) is crucial for learning distinct models. On its own, it requires training a very large number of models ($\gg$32) before picking up features outside the MNIST tiles. Visualizations of input gradients (Figure 6) reveal that these models each rely only on a single or a few pixels. These trivial solutions to the independence constraint, akin to adversarial examples, are avoided with the on-manifold constraint.
- In the full method ($\lambda_{\text{indep}} > 0$, $\lambda_{\text{manifold}} > 0$) the models discover distinct features that align with the semantic contents of images. The effect of the on-manifold constraint on input gradients is striking (Figure 6). It forces models to be sensitive to natural variations of the data – rather than unlikely single-pixel patterns. Remarkably, **image regions emerge as meaningful features without inductive bias for spatial locality** (*e.g.* no convolutions).

Hyperparameters. A number of models between 4 and 24 give excellent results. As expected, the larger this number, the more granular the features these models learn (Figure 6). The effect breaks down for >24 models, matching theoretical expectations since the dimensionality of the manifold was estimated at ∼24. The method is stable over a range of regularizer weights. Additional

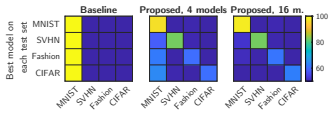
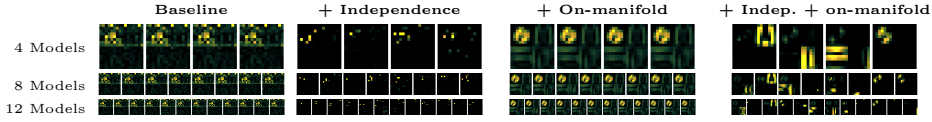


Fig. 5: Collages dataset: accuracy on the four test sets (columns) of models with best accuracy on each set (rows). Diagonal patterns indicate that **models specialize and learn different, non-overlapping features**. The baseline only learns features relevant to MNIST.



(a) With standard training, all models rely on a small, identical region of the image, despite the fact that predictive features are present all over.

(b) Independence produces distinct gradients, but many models are needed to discover new features and they are sensitive to isolated pixels.

(c) The on-manifold constraint forces gradients to align with natural variations of the data. Accuracy is virtually identical to the baseline.

(d) With both constraints, **we learn semantically relevant features in all image regions** with as few as 4 models.

Fig. 6: Input gradients for a random test image from the collages dataset. It is remarkable that, with the proposed method (d) **image regions emerge as meaningful features without any inductive bias for spatial locality** such as convolutions (models in these experiments are fully-connected MLPs).

comparisons in Table 1 show that a VAE is better than PCA to represent the manifold. This agrees with the general expectation that natural images form a non-linear manifold in pixel space. We also found overall results to be robust to variations in architecture and hyperparameters of the VAE.

Fine-tuning. We report the accuracy of models fine-tuned on masked inputs as proposed in Section 3.3. This optional step relaxes the independence constraint to maximize each model's predictive performance. The accuracy jumps significantly and almost reaches the upper-bound on each test set (Table 1). We experimented with relaxing both the independence and on-manifold constraints. Disabling the former has a significant effect. But the latter has no significant effect on accuracy on its own as expected and discussed in Section 3.2.

Distilling multiple models into one. We report the performance of combinations of features described in Section 3.4. This procedure is most effective after training a large number of models (24 here). This is unsurprising since models then discover finer-grained features. Each combination selects features relevant to only one specific tile to achieve near-maximal accuracy on the test set of that tile. Simple traditional ensembling of models completely failed in our experiments.

Comparison with existing methods. No other method reported in Table 1 performed well on this dataset. The method of Teney *et al.* [72] is technically the most similar to ours, but it requires training a much larger number of models and still achieves much lower accuracy. While all experiments of this section used a model taking raw pixels as input, we repeated the whole evaluation using a shared, frozen ResNet to extract features in Appendix H. This implementation

is computationally appealing for larger-scale applications, and gave essentially similar findings with higher overall accuracy thanks to the deeper architecture.

Collages (accuracy in %)	Best model on				
	MNIST	SVHN	Fashion	CIFAR-10	Average
Upper bound (training on test-domain data)	99.9	92.4	80.8	68.6	85.5
ERM Baseline	99.8	50.0	50.0	50.0	62.5
Spectral decoupling [55]	99.9	49.8	50.6	49.9	62.5
Penalty: gradients' L1 norm	98.5	49.6	50.5	50.0	62.1
Penalty: g. L2 norm [30]	96.6	52.1	52.3	54.3	63.8
Input dropout (ratio 0.9)	97.4	50.7	56.1	52.1	64.1
Indep. loss (cos. sim.) [60]	99.7	50.4	51.5	50.2	63.0
Indep. loss (dot prod.) [72]	99.5	53.5	53.3	50.5	64.2
With many more models					
Indep. (cos. sim.), 1024 models	99.5	58.1	66.8	63.0	71.9
Indep. (dot prod.), 128 models	98.7	84.9	71.6	61.5	79.2
Proposed method (8 models)					
Indep. + on-manifold PCA	97.3	69.8	62.2	60.0	72.3
Indep. + on-manifold VAE*	96.5	85.1	61.1	62.1	76.2
(*) + FT (fine-tuning)	99.7	90.9	81.4	67.4	84.8
(*) + FT + combi. (1×)	99.9	92.2	79.3	66.3	84.4
(*) + FT + combi. (2×)	99.9	92.5	80.2	67.5	85.0
(*) + **FT + combi. (3×)**	**99.9**	**92.3**	**80.8**	**68.5**	**85.4**

Table 1: Accuracy on *collages* of existing and proposed methods (8 models per method unless specified). The 4 test sets simulate different OOD conditions: only one tile in each set is correlated with the labels. **Standard training only learns a fraction of predictive features**. Existing methods cannot do better than chance except on MNIST, or they require training a large number of models. Ours learns a variety of features and give near-optimal predictions on every test set (last row).

4.2 Experiments on real data: WILDS-Camelyon17

Dataset. The WILDS-Camelyon17 benchmark [40] provides histopathology images to classify as "*tumor*" or "*normal*". The images come from different sets of hospitals in the training, validation (val-OOD), and test splits (test-OOD). The challenge is to learn a model that generalizes from the training hospitals to those of the test set. The original authors [40] trained a Densenet-121 model from scratch on this data with 10 random seeds. They showed that the performance on val-OOD and test-OOD varies wildly across seeds, demonstrating that the task is severely underspecified with only the standard training images (the dataset provides additional hospital labels that could enable generalization; neither ERM nor our method uses them).

Implementation of our method. We use frozen features (last-layer activations) from one of the pretrained models from [40] as input. We will show that we can recover even more variability in performance than the complete models trained on different random seeds, even while keeping the model frozen (*i.e.* retraining only a classifier). We first determined that the best ERM-trained classifier on frozen features is a simple linear one, rather than an MLP. Our method simplifies in two ways with a linear classifier. First, input gradients are equal to the classifier weights, and the proposed regularizers do not require second-order derivatives anymore during back-propagation. Second, we found empirically that the soft on-manifold regularizer can be replaced with a hard constraint: we explicit project the input gradients onto the manifold and apply the independence

WILDS-Camelyon17	Best accuracy (%) on val-OOD	test-OOD
Pseudo-Label [43]	–	67.7 ±8.2
DANN [17]	–	68.4 ±9.2
FixMatch [68]	–	71.0 ±4.9
CORAL [70]	–	77.9 ±6.6
NoisyStudent [84]	–	**86.7** ±1.7
ERM Baseline	84.9 ±0.1	68.4 ±0.1
+ Independence constraint	85.3 ±0.5	74.6 ±0.9
+ On-manifold soft regularizer, VAE	85.4 ±0.4	80.3 ±1.7
+ On-manifold hard projection, VAE	88.2 ±2.1	76.3 ±2.8
+ On-manifold soft regularizer, PCA	87.8 ±0.3	79.0 ±2.9
+ On-manifold hard projection, PCA*	**88.4** ±0.7	81.6 ±1.4
(*) + Fine-tuning & distillation	**88.4** ±0.7	**82.5** ±2.4

Table 2: Accuracy on WILDS-Camelyon17 while training 12 models. Each proposed component improves the accuracy of the best model from each run. The data appears simple enough that a PCA approximates the manifold well enough. This allows implementing the on-manifold constraint as a hard projection instead of a soft regularizer.

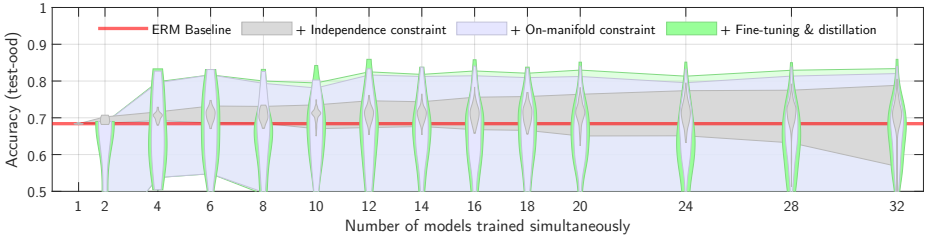


Fig. 7: Spread of accuracies on WILDS-Camelyon17 of models trained with different ablations of our method. The upper/lower bounds of the **shaded areas** show the highest/lowest accuracy of any model from one run, averaged over 6 seeds. The **violins** show distributions of accuracies over *all* seeds (hence some values outside the shaded areas). The independence constraint (**gray**) produces a wide variety of models, as opposed to the baseline (**red** line). However, the highest accuracy in each run grows slowly with the number of models. With the on-manifold constraint (**blue**), the improvement is larger and requires fewer models. Fine-tuning/distillation (**green**) bring additional marginal improvements.

regularizer on these projections, as proposed in [60]. As noted in Section 3.2, this option completely failed in our early experiments with MLPs, but it seems viable with linear classifiers. This further simplifies the implementation.

Results. We plot in Figure 12 the spread of accuracies of models trained with different methods (using features from the first pretrained model from [40], see Appendix I for similar results with the others). The **ERM baseline** simply recovers the accuracy of the original complete Densenet, with essentially no variation across random seeds. With our **independence constraint**, the spread of accuracies significantly widens, both below and above the baseline. In Figure 13, we see that the models span various trade-offs in accuracy on val-OOD and on test-OOD, neither of which is correlated with the accuracy on in-domain data (val-ID), thus showing evidence of underspecification. Back to Figure 12, with our additional **on-manifold constraint**, the best models reach higher accura-

cies. This also tops out when training a handful of models (about 10–14, near the intrinsic dimensionality of the data estimated at 12 with [57]). Keeping in mind that we use frozen features, these results show that the ERM-pretrained model extracts features useful for OOD performance but that are ignored by the pretrained classifier. Similar findings were recently reported in [39,59]. Our method recovers these features and produces alternative classifiers with a variety of trade-offs in performance across various OOD conditions.

Ablations. In Table 2, we compare additional ablations of our method, using a fixed number of 10 models. The **essential components are the independence and on-manifold constraints**. The fine-tuning and distillation steps contribute to a marginal improvement. We report similar *relative* improvements in Appendix I with other pretrained models, but the absolute performance is very much **dependent on a "good" pretrained model**.

Model selection. Our method brings similar relative improvements on either val-OOD or test-OOD but typically with different models for each (see Figure 12) despite both being OOD relative to the training data. **Model selection absent labelled target-domain data therefore remains an issue** on this dataset. Fortunately, only little such data may be sufficient. We repeated a few experiments while holding out 1% of test-OOD (less than 1,000 instances) and we observed a 99.87% correlation coefficient between the accuracy on test-OOD and this held-out data. While all our results assume perfect "oracle" model selection, it seems reasonable that real applications could provide a small amount of labelled test data to achieve similar results.

5 Discussion

We presented a method that highlights cases of underspecification by training multiple models with similar in-domain performance yet different OOD behaviour. This method offers a partial solution to building robust models since it discovers features that are otherwise missed by standard ERM due to shortcut learning or other implicit inductive biases [50,66].

What do we gain from identifying cases of underspecification? The level of underspecification (indicated by the number of models that can be trained with the proposed constraints) shows how far from unique a solution to a learning problem is. Diagnosing underspecification is not a pass-or-fail test: all but the simplest tasks and models are underspecified to some extent. Measuring underspecification should help determining the level of trust attributed to an ML model. Our constructive approach has the added advantage of exposing the range of predictive features present in the data.

Importance to both engineering and science. There is a continuing source of research questions in the apparent mismatch between empirical practices in ML and some hard limitations of learning methods. The concept of underspecification has the potential to unify phenomena including shortcut learning, distribution shifts, and adversarial robustness. These are important for ML as an engineering discipline (improving reliability and applicability of ML methods)

as well as a scientific endeavour (understanding the structure of real-world data and how/why existing methods work).

A first implication of underspecification is that ERM is insufficient to guarantee OOD generalization. Identified cases of underspecification point at the need for additional task-specific information in the design of reliable learning methods. If such information cannot be integrated, learned models are at risk of unexpected behaviour when deployed on OOD data, because the depend on stochastic or arbitrary factors (*e.g.* texture *vs.* shape in image classification [20]).

A second implication is that ID and OOD performance are not necessarily coupled. Without further assumptions, in-domain validation is not a reliable model selection strategy for OOD performance despite some suggestions *e.g.* in [24,48]. It might be useful as a heuristic owing to some inherent structure in real-world data, but its limits of applicability are yet to be understood.

A third implication is that high OOD performance of a model is no guarantee for its reliability. High apparent performance might happen by accident in an underspecified setting. In such cases, the model behaviour depends on hidden assumptions and it could still fail unexpectedly. Identifying underspecification remains important to identify these hidden assumptions, which is particularly important for high-stakes applications such as medical imaging [5,21].

The proposed analysis also corroborates existing explanations for techniques that successfully improve generalization, such as data augmentation and contrastive learning. Both were indeed shown to depend on the injection of additional knowledge, respectively in the design of the augmentations [10,34,42] and pair selection strategy [86]. And this extra knowledge is often task-specific [83]. For example, augmenting images with rotations may help in identifying flowers but not traffic signs. Injecting task-specific knowledge is sometimes vilified in a "data-driven" culture. This study suggests that we would rather benefit from highlighting this practice and making assumptions more explicit, thus helping one to identify the limits of applicability of various methods.

Conclusion. This paper made theoretical and methodological steps on the study of underspecification. It complements an observational study [12] with a method to diagnose and address the problem.

Limitations. The proposed method for building models with better generalization is only a partial solution since it requires an external model selection procedure. New methods for model selection [15,19,35,79], robust evaluation [18,36], and explainability [22,75] are all suitable to implement this selection. Interactive approaches [14] are another option that injects expert knowledge. Another possible extension is to apply the method to the end-to-end training of larger models. Finally, this work focused on i.i.d. training data. We hope to extend the analysis to forms of data known to be valuable for OOD generalization such as multiple environments [4,54,11], counterfactual examples [36,71], and non-stationary data [1,26,56,78]. The analysis of multi-environment training as used for domain generalization may elucidate why these methods often fail in practice [24].

References

1. Alesiani, F., Yu, S., Yu, X.: Gated information bottleneck for generalization in sequential environments. arXiv preprint arXiv:2110.06057 (2021) 14
2. Allen-Zhu, Z., Li, Y.: Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. arXiv preprint arXiv:2012.09816 (2020) 20
3. Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L.: Bottom-up and top-down attention for image captioning and VQA. CVPR (2018) 25, 33
4. Arjovsky, M., Bottou, L., Gulrajani, I., Lopez-Paz, D.: Invariant risk minimization. arXiv preprint arXiv:1907.02893 (2019) 14, 20
5. Banerjee, I., Bhimireddy, A.R., Burns, J.L., Celi, L.A., Chen, L.C., Correa, R., Dullerud, N., Ghassemi, M., Huang, S.C., Kuo, P.C., et al.: Reading race: Ai recognises patient's racial identity in medical images. arXiv preprint arXiv:2107.10356 (2021) 14
6. Bareinboim, E., Correa, J., Ibeling, D., Icard, T.: On Pearl's hierarchy and the foundations of causal inference. ACM Special Volume in Honor of Judea Pearl (provisional title) (2020) 20, 21
7. Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence **35**(8), 1798–1828 (2013) 21
8. Bhatt, U., Zafar, M.B., Gummadi, K., Weller, A.: Counterfactual accuracies for alternative models. In: ICLR Workshop on Machine Learning in Real Life Workshop (2020) 20
9. Breiman, L.: Statistical modeling: The two cultures (with comments and a rejoinder by the author). Statistical science **16**(3), 199–231 (2001) 20
10. Cubuk, E.D., Dyer, E.S., Lopes, R.G., Smullin, S.: Tradeoffs in data augmentation: An empirical study (2021) 14
11. Damien Teney, Ehsan Abbasnejad, A.v.d.H.: Unshuffling data for improved generalization. arXiv preprint arXiv:2002.11894 (2020) 14, 20
12. D'Amour, A., Heller, K., Moldovan, D., Adlam, B., Alipanahi, B., Beutel, A., Chen, C., Deaton, J., Eisenstein, J., Hoffman, M.D., et al.: Underspecification presents challenges for credibility in modern machine learning. arXiv preprint arXiv:2011.03395 (2020) 1, 2, 14, 20
13. Dancette, C., Cadene, R., Teney, D., Cord, M.: Beyond question-based biases: Assessing multimodal shortcut learning in visual question answering. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2021) 33
14. Das, S., Cashman, D., Chang, R., Endert, A.: Beames: Interactive multimodel steering, selection, and inspection for regression tasks. IEEE computer graphics and applications **39**(5), 20–32 (2019) 14
15. Deng, W., Gould, S., Zheng, L.: What does rotation prediction tell us about classifier accuracy under varying testing environments? arXiv preprint arXiv:2106.05961 (2021) 14, 22
16. Fisher, A., Rudin, C., Dominici, F.: All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. J. Mach. Learn. Res. **20**(177), 1–81 (2019) 20
17. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. J. Mach. Learn. Res. (2016) 12

18. Gardner, M., Artzi, Y., Basmova, V., Berant, J., Bogin, B., Chen, S., Dasigi, P., Dua, D., Elazar, Y., Gottumukkala, A., et al.: Evaluating NLP models via contrast sets. arXiv preprint arXiv:2004.02709 (2020) 14
19. Garg, S., Balakrishnan, S., Kolter, J.Z., Lipton, Z.C.: Ratt: Leveraging unlabeled data to guarantee generalization. arXiv preprint arXiv:2105.00303 (2021) 14, 22
20. Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W.: Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. arXiv preprint arXiv:1811.12231 (2018) 2, 14
21. Ghimire, S., Kashyap, S., Wu, J.T., Karargyris, A., Moradi, M.: Learning invariant feature representation to improve generalization across chest x-ray datasets. In: International Workshop on Machine Learning in Medical Imaging (2020) 14
22. Goyal, Y., Wu, Z., Ernst, J., Batra, D., Parikh, D., Lee, S.: Counterfactual visual explanations. In: International Conference on Machine Learning. pp. 2376–2384. PMLR (2019) 14
23. Gretton, A., Herbrich, R., Smola, A.J.: The kernel mutual information. In: 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). vol. 4, pp. IV–880. IEEE (2003) 5
24. Gulrajani, I., Lopez-Paz, D.: In search of lost domain generalization (2021) 2, 14
25. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. **3**, 1157–1182 (2003) 20
26. Hälvä, H., Hyvarinen, A.: Hidden markov nonlinear ica: Unsupervised learning from nonstationary time series. In: Conference on Uncertainty in Artificial Intelligence. pp. 939–948. PMLR (2020) 14, 20
27. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2016) 24
28. Heinze-Deml, C., Meinshausen, N.: Conditional variance penalties and domain shift robustness. arXiv preprint arXiv:1710.11469 (2017) 20
29. Heljakka, A., Trapp, M., Kannala, J., Solin, A.: Representational multiplicity should be exposed, not eliminated. arXiv preprint arXiv:2206.08890 (2022) 20
30. Hoffman, J., Roberts, D.A., Yaida, S.: Robust learning with jacobian regularization. arXiv preprint arXiv:1908.02729 (2019) 11
31. Hsu, H., Calmon, F.d.P.: Rashomon capacity: A metric for predictive multiplicity in probabilistic classification. arXiv preprint arXiv:2206.01295 (2022) 4, 20
32. Hudson, D.A., Manning, C.D.: GQA: A new dataset for real-world visual reasoning and compositional question answering. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2019) 2, 3, 33, 35
33. Hyvarinen, A., Morioka, H.: Nonlinear ICA of temporally dependent stationary sources. In: Artificial Intelligence and Statistics. pp. 460–469. PMLR (2017) 20
34. Ilse, M., Tomczak, J.M., Forré, P.: Designing data augmentation for simulating interventions (2021) 14
35. Immer, A., Bauer, M., Fortuin, V., Rätsch, G., Khan, M.E.: Scalable marginal likelihood estimation for model selection in deep learning. arXiv preprint arXiv:2104.04975 (2021) 14, 22
36. Kaushik, D., Hovy, E., Lipton, Z.C.: Learning the difference that makes a difference with counterfactually-augmented data. arXiv preprint arXiv:1909.12434 (2019) 14
37. Kervadec, C., Antipov, G., Baccouche, M., Wolf, C.: Roses are red, violets are blue... but should VQA expect them to? In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2021) 2, 3, 33
38. Khemakhem, I., Kingma, D., Monti, R., Hyvarinen, A.: Variational autoencoders and nonlinear ica: A unifying framework. In: International Conference on Artificial Intelligence and Statistics. pp. 2207–2217. PMLR (2020) 20

39. Kirichenko, P., Izmailov, P., Wilson, A.G.: Last layer re-training is sufficient for robustness to spurious correlations. arXiv preprint arXiv:2204.02937 (2022) 13
40. Koh, P.W., Sagawa, S., Marklund, H., Xie, S.M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R.L., Beery, S., et al.: Wilds: A benchmark of in-the-wild distribution shifts. arXiv preprint arXiv:2012.07421 (2020) 11, 12, 32
41. Koh, P.W., Sagawa, S., Marklund, H., Xie, S.M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R.L., Gao, I., et al.: Wilds: A benchmark of in-the-wild distribution shifts. In: Proc. Int. Conf. Mach. Learn. (2021) 2, 3
42. von Kügelgen, J., Sharma, Y., Gresele, L., Brendel, W., Schölkopf, B., Besserve, M., Locatello, F.: Self-supervised learning with data augmentations provably isolates content from style. arXiv preprint arXiv:2106.04619 (2021) 14
43. Lee, D.H., et al.: Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on challenges in representation learning, ICML (2013) 12
44. Lee, Y., Yao, H., Finn, C.: Diversify and disambiguate: Learning from underspecified data. arXiv preprint arXiv:2202.03418 (2022) 21
45. Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., Bachem, O.: Challenging common assumptions in the unsupervised learning of disentangled representations. In: international conference on machine learning. pp. 4114–4124. PMLR (2019) 20, 21
46. Marx, C., Calmon, F., Ustun, B.: Predictive multiplicity in classification. In: International Conference on Machine Learning. pp. 6765–6774. PMLR (2020) 20
47. Mehrer, J., Spoerer, C.J., Kriegeskorte, N., Kietzmann, T.C.: Individual differences among deep neural network models. Nature communications **11**(1), 1–12 (2020) 2, 20
48. Miller, J.P., Taori, R., Raghunathan, A., Sagawa, S., Koh, P.W., Shankar, V., Liang, P., Carmon, Y., Schmidt, L.: Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization. In: Proc. Int. Conf. Mach. Learn. (2021) 2, 14
49. Mitchell, T.M.: The need for biases in learning generalizations. Rutgers University (1980) 1, 4
50. Ortiz-Jimenez, G., Salazar-Reque, I.F., Modas, A., Moosavi-Dezfooli, S.M., Frossard, P.: A neural anisotropic view of underspecification in deep learning. In: Proc. Int. Conf. Learn. Representations (2021) 13
51. Pagliardini, M., Jaggi, M., Fleuret, F., Karimireddy, S.P.: Agree to disagree: Diversity through disagreement for better transferability. arXiv preprint arXiv:2202.04414 (2022) 21
52. Parker-Holder, J., Metz, L., Resnick, C., Hu, H., Lerer, A., Letcher, A., Peysakhovich, A., Pacchiano, A., Foerster, J.: Ridge rider: Finding diverse solutions by following eigenvectors of the hessian. arXiv preprint arXiv:2011.06505 (2020) 20
53. Pennington, J., Socher, R., Manning, C.: Glove: Global Vectors for Word Representation. In: Conference on Empirical Methods in Natural Language Processing (2014) 25
54. Peters, J., Bühlmann, P., Meinshausen, N.: Causal inference by using invariant prediction: identification and confidence intervals. Journal of the Royal Statistical Society: Series B (Statistical Methodology) (2016) 14, 20
55. Pezeshki, M., Kaba, S.O., Bengio, Y., Courville, A., Precup, D., Lajoie, G.: Gradient starvation: A learning proclivity in neural networks. arXiv preprint arXiv:2011.09468 (2020) 11

56. Pfister, N., Bühlmann, P., Peters, J.: Invariant causal prediction for sequential data. Journal of the American Statistical Association **114**(527), 1264–1276 (2019) 14, 20
57. Pope, P., Zhu, C., Abdelkader, A., Goldblum, M., Goldstein, T.: The intrinsic dimension of images and its impact on learning. arXiv preprint arXiv:2104.08894 (2021) 8, 9, 13, 21
58. Renard, X., Laugel, T., Detyniecki, M.: Understanding prediction discrepancies in machine learning classifiers. arXiv preprint arXiv:2104.05467 (2021) 20
59. Rosenfeld, E., Ravikumar, P., Risteski, A.: Domain-adjusted regression or: Erm may already learn features sufficient for out-of-distribution generalization. arXiv preprint arXiv:2202.06856 (2022) 13
60. Ross, A., Pan, W., Celi, L., Doshi-Velez, F.: Ensembles of locally independent prediction models. In: Proc. Conf. AAAI (2020) 5, 6, 11, 12, 20
61. Ross, A.S., Pan, W., Doshi-Velez, F.: Learning qualitatively diverse and interpretable rules for classification. arXiv preprint arXiv:1806.08716 (2018) 5, 6, 20
62. Rudin, C., Chen, C., Chen, Z., Huang, H., Semenova, L., Zhong, C.: Interpretable machine learning: Fundamental principles and 10 grand challenges. arXiv preprint arXiv:2103.11251 (2021) 20
63. Schölkopf, B., Locatello, F., Bauer, S., Ke, N.R., Kalchbrenner, N., Goyal, A., Bengio, Y.: Toward causal representation learning. Proceedings of the IEEE (2021) 20, 21
64. Selvaraju, R.R., Lee, S., Shen, Y., Jin, H., Ghosh, S., Heck, L., Batra, D., Parikh, D.: Taking a hint: Leveraging explanations to make vision and language models more grounded. In: Proc. IEEE Int. Conf. Comp. Vis. (2019) 3, 22, 33
65. Semenova, L., Rudin, C., Parr, R.: A study in rashomon curves and volumes: A new perspective on generalization and model simplicity in machine learning. arXiv preprint arXiv:1908.01755 (2019) 4, 20
66. Shah, H., Tamuly, K., Raghunathan, A., Jain, P., Netrapalli, P.: The pitfalls of simplicity bias in neural networks. arXiv preprint arXiv:2006.07710 (2020) 2, 3, 8, 13
67. Shimodaira, H.: Improving predictive inference under covariate shift by weighting the log-likelihood function. Journal of statistical planning and inference **90**(2), 227–244 (2000) 1
68. Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C.A., Cubuk, E.D., Kurakin, A., Li, C.L.: Fixmatch: Simplifying semi-supervised learning with consistency and confidence. Proc. Advances in Neural Inf. Process. Syst. (2020) 12
69. Stacey, J., Belinkov, Y., Rei, M.: Natural language inference with a human touch: Using human explanations to guide model attention. arXiv preprint arXiv:2104.08142 (2021) 22
70. Sun, B., Feng, J., Saenko, K.: Correlation alignment for unsupervised domain adaptation. In: Domain Adaptation in Computer Vision Applications. Springer (2017) 12
71. Teney, D., Abbasnedjad, E., van den Hengel, A.: Learning what makes a difference from counterfactual examples and gradient supervision. arXiv preprint arXiv:2004.09034 (2020) 14
72. Teney, D., Abbasnejad, E., Lucey, S., van den Hengel, A.: Evading the simplicity bias: Training a diverse set of models discovers solutions with superior OOD generalization. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2022) 2, 3, 8, 9, 10, 11, 20, 22, 23, 24
73. Teney, D., Anderson, P., He, X., van den Hengel, A.: Tips and tricks for visual question answering: Learnings from the 2017 challenge. CVPR (2018) 25

74. Teney, D., Kafle, K., Shrestha, R., Abbasnejad, E., Kanan, C., van den Hengel, A.: On the value of out-of-distribution testing: An example of Goodhart's law. In: Proc. Advances in Neural Inf. Process. Syst. (2020) 33
75. Thiagarajan, J., Narayanaswamy, V.S., Rajan, D., Liang, J., Chaudhari, A., Spanias, A.: Designing counterfactual generators using deep model inversion (2021) 14
76. Träuble, F., Creager, E., Kilbertus, N., Locatello, F., Dittadi, A., Goyal, A., Schölkopf, B., Bauer, S.: On disentangled representations learned from correlated data. In: Proc. Int. Conf. Mach. Learn. (2021) 21
77. Vapnik, V.: Statistical learning theory. john wiley&sons. Inc., New York (1998) 2
78. Venkateswaran, P., Muthusamy, V., Isahagian, V., Venkatasubramanian, N.: Environment agnostic invariant risk minimization for classification of sequential datasets. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. pp. 1615–1624 (2021) 14
79. Wald, Y., Feder, A., Greenfeld, D., Shalit, U.: On calibration and out-of-domain generalization. arXiv preprint arXiv:2102.10395 (2021) 14, 22
80. Wang, H., Yeung, D.Y.: A survey on bayesian deep learning. ACM Computing Surveys (CSUR) **53**(5), 1–37 (2020) 20
81. Weinberger, K.Q., Saul, L.K.: Unsupervised learning of image manifolds by semidefinite programming. Int. J. Comput. Vision pp. 77–90 (2006) 3
82. Wolpert, D.H.: The lack of a priori distinctions between learning algorithms. Neural computation **8**(7), 1341–1390 (1996) 1
83. Xiao, T., Wang, X., Efros, A.A., Darrell, T.: What should not be contrastive in contrastive learning. arXiv preprint arXiv:2008.05659 (2020) 14
84. Xie, Q., Luong, M.T., Hovy, E., Le, Q.V.: Self-training with noisy student improves ImageNet classification. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2020) 12
85. Yashima, S., Suzuki, T., Ishikawa, K., Sato, I., Kawakami, R.: Feature space particle inference for neural network ensembles. arXiv preprint arXiv:2206.00944 (2022) 20
86. Zimmermann, R.S., Sharma, Y., Schneider, S., Bethge, M., Brendel, W.: Contrastive learning inverts the data generating process. arXiv preprint arXiv:2102.08850 (2021) 14

Appendices

A Related work

Underspecification in ML. An empirical study by d'Amour *et al.* in [12] showed that models trained with different random seeds have wide variations in OOD performance despite similar in-domain performance. The effect was surprising in its prevalence but it echoed earlier observations [47]. It results from the well-known impossibility of achieving OOD generalization solely through ERM and i.i.d. data [4,63]. Indeed, since this learning objective does not constrain the model's behaviour outside the training distribution (*i.e.* multiple hypotheses are plausible), additional assumptions/expertise is necessary *e.g.* to tweak the network architecture to the task. Data-driven OOD generalization is also possible but it requires heterogeneous (non-i.i.d.) data such as multiple environments [4,54,11], counterfactual examples [28,72], or non-stationary data [26,33,56]. The impossibility of OOD generalization from i.i.d. data is intimately related to identifiability in causal discovery [6] and non-linear ICA [38], and the impossibility of unsupervised disentanglement in representation learning [45].

Multiple hypotheses compatible with the data. The **Rashomon effect** [9] is almost synonymous with underspecfication but it is agnostic to OOD data. The Rashomon set [16] is the set of models of a given class whose training loss or in-domain (ID) risk lies below a threshold [31,62,65]. **Predictive multiplicity** [8,46,58] refers to the existence of multiple models compatible with the data that make conflicting predictions (on ID data again, rather than OOD). This paper is about models with low ID risk, yet different OOD behaviour. We propose a method to **learn multiple models** compatible with the data. Related approaches were described for ensembling [60], interpretability [61], and control of inductive biases [72]. The number of models required in [72] was very large. We solve this issue by constraining models to align with the data manifold. Other works concerned with the identification of multiple solutions to a learning problem include [52], Bayesian deep learning [80], and classical feature selection [25], none of which are suitable to large-scale models and datasets.

Ensembles. Our method for combining robust features resembles traditional ensembling, which lowers variance by combining predictors with uncorrelated errors. In comparison however, we lower prediction bias by selecting features causally related to the target. The key in our approach is to discover predictive features that are otherwise missed by ERM.

Diversity in feature space. Allen-Zhu and Li [2] explained the success of deep ensembles with the use of different features by different networks simply because of different initializations. Their argument is also based on the ubiquity of underspecification which they call "multi-view structure" of data. Concurrently to our work, [85] used this explanation to build better ensembles by encouraging diversity in *feature space* with a Bayesian framework. Even more recently, Heljakka *et al.* [29] showed how to quantify representational (*i.e.* feature-wise) multiplicity.

Diversity in prediction space. Concurrently to our work, [51] and [44] proposed to learn models that maximally disagree in OOD predictions. Both learn only two models. Obviously, neither would do well on our *collage* dataset (Section 4.1) which requires discovering 4 predictive signals. On real vision datasets, our experiments also suggest that many predictive features ($\gg$2) can be discovered.

B Analogy with disentanglement in representation learning

There are parallels between our discovery of simple predictive features and the topic of disentanglement in representation learning. Disentanglement aims to identify all independent factors of variation in the data-generating process [7] *i.e.* in a **task-independent** manner. In comparison, we are interested only in features that are correlated with **task-specific** labels in a given dataset. Many disentanglement methods are based on the independence of the factors (even though it is usually insufficient [76]) while we rely on the independence of *predictions* with each feature. Our method can therefore be seen as a form of disentanglement in the space of predictors. This parallel also has implications for identifiability. Disentanglement of the correct generative factors was shown to be impossible without additional assumptions or supervision [45]. In our setting, the identification of features that are causally-related to the label (as opposed to spuriously-correlated) also requires additional information [6,63]. Our formulation relegates this need to an external "model selection strategy" that allows flexibility in its actual realization.

C Practical considerations

Number of models to train. We initially estimate the dimensionality of the manifold (d_{manifold}) from $\mathcal{D}_{\text{OOD}}$. This is the upper bound on the number of predictors aligned with the manifold that can be mutually independent. The dimensionality is estimated with a simple method [57] based on distances between nearest neighbours. Only the worst case (highest degree of underspecification per Definition 1) should allow training a number of models $M = d_{\text{manifold}}$ while satisfying the independence and on-manifold constraints. In our experiments, underspecification is usually less severe. We observed that the value of the predictive loss ($\mathcal{L}_{\text{pred}}$) on training and validation data would remain high for some models, while it would converge normally for others. Therefore, monitoring the loss per model could potentially serve for evaluating the degree of underspecification of a given setting (dataset and architecture) by noting the number of models that converge. A thorough investigation and validation of this technique is an important topic left for future work.

Is the data manifold necessary in the definition of underspecification ? Our definition without the reference to the data manifold would be

pointless because virtually all conceivable models would be severely underspecified. The manifold (*e.g.* natural photographs) only fills a fraction of the ambient input space (*e.g.* all possible RGB pixel grids). Our definition captures underspecification with respect to this smaller range of inputs that are expected at test time.

How to select models to combine into a global predictor? Our approach is agnostic to the model selection strategy. Many options are possible to identify models that rely on "robust" features. The simplest is an evaluation on an OOD validation set. Alternatively, interpretability methods allow domain experts to inspect models and the features they rely on. The selection may also be semi-automated with task-specific heuristics [15,19,35,79] or additional annotations *e.g.* human attention and rationales [64,69].

Computational cost. With parallelization, our method scales sublinearly in computing time and linearly in memory w.r.t. the number of models. Models share mini-batches. And since they use the same architecture, most computations can be parallelized with **grouped convolutions** (one group per model). The method is also applicable on a frozen feature extractor (Section 4.2 and Appendix H). However, fine-tuning a shared extractor with the independence loss is not possible. The model could simply dispatch redundant features on multiple channels and satisfy the independence constraint without the desired increase in diversity. Compared to [72], we significantly reduce the number of models needed to discover useful features.

D Projections on the manifold

The on-manifold loss defined in Eq. (2) requires projecting a model's input gradients on the data manifold. The manifold is characterized by a provided set of unlabeled data $\mathcal{D}_{\text{OOD}} = \{\boldsymbol{x}_i\} \sim \text{P}_{\text{OOD}}$. We propose two implementations of the projection function $\text{proj}(\cdot)$, using either a simple principal component analysis (PCA) model of the manifold or a variational auto-encoder (VAE). A PCA is fast to evaluate but can only model a linear manifold, which is likely overly simplistic for most real datasets. A VAE allows learning a non-linear manifold. Better implementations are probably possible by taking advantage of recent developments in OOD detection and generative models such as GANs, EBMs, and diffusion models.

In the case of a PCA model, the projection is a straightforward linear projection on the top components of a PCA basis of $\mathcal{D}_{\text{OOD}}$. In the case of a VAE, we train an auto-encoder on the data $\mathcal{D}_{\text{OOD}}$. We interpret it as a generative model of the manifold, assuming that inputs to the auto-encoder will be mapped to a nearby projection on the manifold. We denote the auto-encoder as a function $\text{proj}_{\boldsymbol{\psi}} : \mathbb{R}^{d_{\text{in}}} \rightarrow \mathcal{M}$ of parameters $\boldsymbol{\theta}$ such that

$$\text{proj}_{\boldsymbol{\psi}}(\boldsymbol{x}) \approx \boldsymbol{x} \quad \forall\ \boldsymbol{x} \sim \text{P}_{\text{OOD}}. \tag{6}$$

Therefore, the reconstruction error $||\boldsymbol{x} - \text{proj}_{\boldsymbol{\psi}}(\boldsymbol{x})||$ is minimal for points $\boldsymbol{x}$ on to the manifold.

The function proj is trained to reconstruct points, but it is also readily capable of projecting a vector $\boldsymbol{v} \in \mathbb{R}^{d_{\text{in}}}$ originating at $\boldsymbol{x}$. In our case, we will use it to project the input gradient ($\boldsymbol{v} = \nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}}(\boldsymbol{x})$). We overload the function as $\text{proj}_{\boldsymbol{\psi}} : \mathbb{R}^{d_{\text{in}}} \times \mathbb{R}^{d_{\text{in}}} \rightarrow \mathcal{M}$ with

$$\text{proj}_{\boldsymbol{\psi}}(\boldsymbol{x}, \boldsymbol{v}) \approx \boldsymbol{v} \quad \forall\ \boldsymbol{x} \sim \text{P}_{\text{OOD}}, \quad (\boldsymbol{x} + \boldsymbol{v}) \sim \text{P}_{\text{OOD}}. \tag{7}$$

Here again, the reconstruction error $||\boldsymbol{v} - \text{proj}_{\boldsymbol{\psi}}(\boldsymbol{x}, \boldsymbol{v})||$ is minimal for a point $\boldsymbol{x}$ on the manifold and vector $\boldsymbol{v}$ **aligned with the manifold**. We can therefore apply proj$(\cdot)$ to our input gradients, measure the distance with their projection on the manifold, and use this distance as our on-manifold loss in Eq. (2).

Reconstructing vectors with an auto-encoder. This section describes how to take a standard auto-encoder, typically trained to reconstruct a point $\boldsymbol{x}$ of its input space, and use it to reconstruct a vector $\boldsymbol{v}$ in this space (input gradients in our case). The auto-encoders used this work are compositions of linear layers, ReLU activations, and a sigmoid output activation. Table 3 provides the equivalent operations performed during forward propagation in any such layer for reconstructing an input or gradient.

Table 3: Operations performed at each layer of a VAE during forward propagation for reconstructing points (left column) and vectors/gradients (right column).

Operation on point $\boldsymbol{x}$	**Operation on vector $\boldsymbol{v}$**
Linear layer $\boldsymbol{x} \leftarrow \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}$	$\boldsymbol{v} \leftarrow \boldsymbol{W}\boldsymbol{v}$
ReLU activation $\boldsymbol{x} \leftarrow \boldsymbol{x} \odot \mathbb{1}(\boldsymbol{x} > 0)$	$\boldsymbol{v} \leftarrow \boldsymbol{v} \odot \mathbb{1}(\boldsymbol{x} > 0)$
Sigmoid activation $x \leftarrow \sigma(x) = 1/(1 + e^{-x})$	$\boldsymbol{v} \leftarrow \boldsymbol{v}\ \sigma(x)\ \big(1 - \sigma(\boldsymbol{x})\big)$

E Distilling multiple models into one

After training a set of models, we propose to combine the best of them into one a global predictor with superior OOD performance. We train this predictor with the same fine-tuning as described in Section 3.3, using masks of selected models aggregated with a logical OR. We outline in Algorithm 2 a simple procedure to iteratively combine models pairwise and greedily.

F Experimental details: collages

Collages dataset, raw pixels as input. The *collages* dataset is described in [72]. We used the 4-block ordered version (each of the four source datasets ap-

Algorithm 2: Distilling multiple models into one with greedy pairwise combinations.

Inputs:
$\boldsymbol{S} = \{f_{\boldsymbol{\theta}_1} \dots f_{\boldsymbol{\theta}_M}\}$: Set of independent models.
getNBestModels$(\cdot)$: Model selection strategy, *e.g.* evaluation on OOD validation set.

Result:
$f_{\boldsymbol{\theta}_\star}$: Best combined model according to given strategy.

Method:
$\lambda_{\text{indep}} \leftarrow 0,\ \lambda_{\text{manifold}} \leftarrow 0$ // Use only predictive loss
do
  $\{\boldsymbol{\theta}_k, \boldsymbol{\theta}_l\} \leftarrow$ get2BestModels$(\boldsymbol{S})$
  $\mathbf{mask}_i^\star \leftarrow \mathbf{mask}_i^k \vee \mathbf{mask}_i^l \quad \forall i$ // Combine masks
  $\mathcal{D}_{\text{tr}}^\star \leftarrow \{(\boldsymbol{x}_i \odot \mathbf{mask}_i^\star, y_i)\}_i$ // Prepare masked data
  $\boldsymbol{\theta}_\star \leftarrow \text{argmin}\, \mathcal{L}(\mathcal{D}_{\text{tr}}^m, \boldsymbol{\theta}_k)$ // Fine-tune
  $\boldsymbol{S} \leftarrow \boldsymbol{S} \bigcup \boldsymbol{\theta}_\star$ // Append to set of models
while get1BestModel$(\boldsymbol{S}) = \boldsymbol{\theta}_\star$ // New combination is best
$\boldsymbol{\theta}_\star \leftarrow$ get1BestModel$(\boldsymbol{S})$ // Return best model

pear in the same quadrant in every instance). We re-generated the collages with bilinear downsampling by 1/4th using the code from the authors.[7] The reason we do not use full-size images is purely computational. The original data [72] used nearest-neighbour downsampling because it preserves the contrast and dynamic range, but bilinear downsampling produces smoother images that better match the manifold assumption of natural data that is important in this work. The bilinear downsampling causes a compression of the dynamic range that is uneven across the tiles and mostly affects SVHN and CIFAR. To compensate for it, we apply a standard local contrast normalization (`adapthisteq` in Matlab). We also replace the pixels with a constant value near 0 with small noise. These pixels exist in the MNIST data but are again unrepresentative of natural data.

In Matlab code:

```
% Local contrast normalization
img = adapthisteq(img, 'NumTiles', [2 2]);
% Replace dark pixels with small random values
img(img < .05) = rand(nnz(img < .05), 1) * .05;
```

Collages dataset, ResNet features as input. We obtain feature maps by passing full-resolution images of collages into a standard frozen ImageNet-pretrained ResNet-18 [27]. We chose to use feature maps from an intermediate layer `res3b_relu` because it retains a reasonable spatial resolution (8×8 with 128 channels). The independence loss (1) for these experiments uses a dot product rather than a cosine similarity. It works better but we don't know why.

[7] https://github.com/dteney/collages-dataset

Hyperparameters. See Table 4. The classifiers' hyperparameters are chosen to optimize the baseline reported as "Upper bounds" in the results. None of these hyperparameters are particularly tuned to the proposed methods.

Baselines. The "dropout" baseline in Table 1 is implemented with as a standard dropout on the input pixels, with a different dropout mask for each model. The idea is that each model sees a different (dropped-out) version of the input, and therefore might pick up different features.

G Experimental details: GQA

VQA Model. For our experiments on visual question answering, we use a simplified version of the classical BUTD model [73]. For the text input (question), we use standard 300-dimensional GloVE embeddings [53] (frozen) averaged over the sequence. For the image input, we use the object features provided with the GQA dataset. These are 2048-dimensional from a "bottom-up" object detector [3]. We average and L2-normalize these features to obtain a single 2048D vector representing each image. The text and question vectors are linearly projected to a common dimension (256), combined with an element-wise product, then passed through a 1-layer MLP to obtain scores over candidate answers.

Grad-CAM Rank correlation. The rank correlation reported in Table 8 is the Spearman rank correlation of grad-CAM scores on the validation set, average over questions and pairs of models. The grad-CAM scores correspond to the 2048D input gradients multiplied by the unpooled (36×2048D) object features. The grad-CAM scores are used because they provide a spatial importance map over the image despite the model using globally-pooled features (hence an input gradient that is spatially uniform over the image).

The proposed method is applied in the same way as with the collages. Almost all hyperparameters are remarkably identical to those for the collages (see Table 5).

Table 4: Hyperparameters used on the collages dataset.

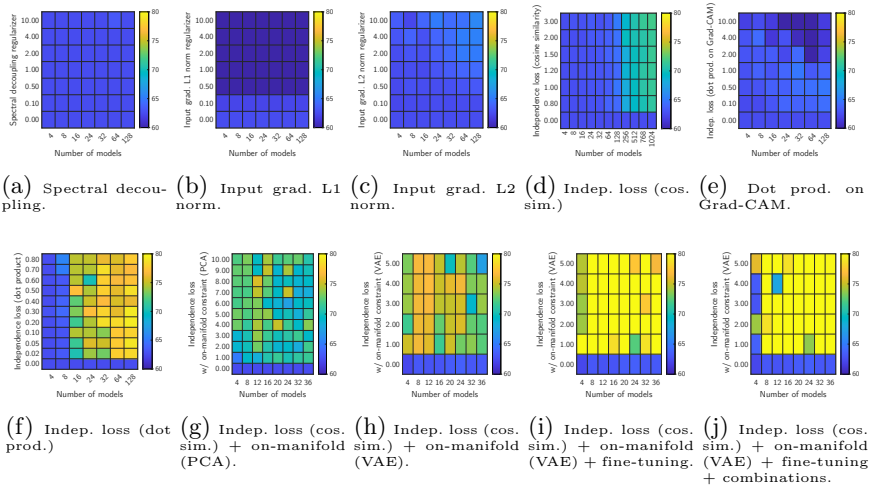
Hyperparameter	**Collages, pixels as input**	**ResNet features as input**
Classifier		
Input dimensions	16×16×1 (grayscale)	8×8×128 (res3b_relu layer)
Architecture	1 Hidden layer (fully-connected) 8 neurons Output layer (fully-connected).	1 Hidden layer (channel-wise) 16 neurons Output layer (fully-connected)
Hidden Activations	Leaky ReLU, leak scale 0.01	Leaky ReLU, leak scale 0.01
Output activation	Sigmoid	Sigmoid
Mini-batch size	256	256
Optimizer	Adam	Adam
Learning rate	0.002	0.001
Optimization length	10,000 Updates No early stopping	30,000 Updates No early stopping
PCA Manifold model		
Number of components	24	–
Retained variance	85%	–
VAE Manifold model		
Architecture	2-Hidden layer MLP 128 neurons per layer	2-Hidden layer MLP 128 neurons per layer
Latent dimensions	24	24
Hidden activations	ReLU	ReLU
Output activations	Sigmoid	Sigmoid
Mini-batch size	256	128
Optimizer	Adam	Adam
Learning rate	0.001	0.001
Optimization length	100 Epochs	300 Epochs
Weight of KL loss	0.01 (small to prioritize accurate reconstructions)	0.01

Table 5: Hyperparameters used on GQA and WILDS-Camelyon17.

Hyperparameter	**GQA**	**WILDS-Camelyon17**
Classifier		
Input dimensions	1×300 (text) 1×2048 (image)	1×1024 (DenseNet features)
Hidden layers dimension	256	–
Hidden activations	Leaky ReLU	–
Output activations	Sigmoid	Sigmoid
Mini-batch size	256	256
Optimizer	Adam	Adam
Learning rate	0.002	0.001
Optimization length	50,000 Updates (40 epochs) No early stopping	12,000 Updates (10 epochs)
PCA Manifold model		
Number of components	168 (performs best)	Same as number of models
Retained variance	85%	Varies: 75% with 2 components, 85% with 3, 90% with 6, 95% with 13, 97% with 25
VAE Manifold model		
Architecture	2-Hidden layer MLP 512 neurons per layer	1-Hidden layer MLP 128 neurons per layer
Latent dimensions	32	14
Hidden Activations	ReLU	ReLU
Output activation	None	None
Mini-batch size	256	256
Optimizer	Adam	Adam
Learning rate	0.001	0.001
Optimization length	100 Epochs	20 Epochs
Weight of KL loss	0.01	0.01

H Additional results: collages

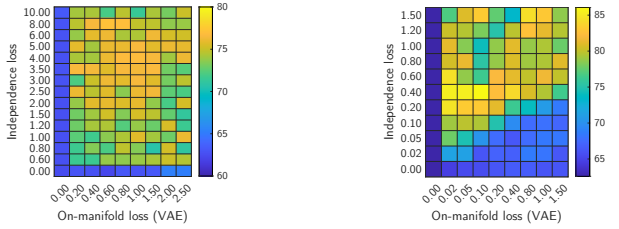
We provide below additional results on the *collages* dataset. We also include experiments using **features from a ResNet-18 as inputs** rather than raw pixels. We use a standard ResNet-18 pretrained on ImageNet then kept frozen. Our method is applied identically as in other experiments, except that the independence and on-manifold constraints are now applied in the space of ResNet feature maps rather than in pixel space. The manifold model is a VAE trained on such feature maps. **The behaviour of our method with ResNet features is qualitatively the same as in other experiments.** This demonstrates the applicability of the method on features from deep architectures.



(a) Spectral decoupling. (b) Input grad. L1 norm. (c) Input grad. L2 norm. (d) Indep. loss (cos. sim.) (e) Dot prod. on Grad-CAM.

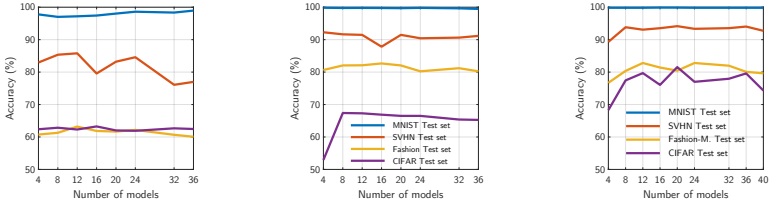
(f) Indep. loss (dot prod.) (g) Indep. loss (cos. sim.) + on-manifold (PCA). (h) Indep. loss (cos. sim.) + on-manifold (VAE). (i) Indep. loss (cos. sim.) + on-manifold (VAE) + fine-tuning. (j) Indep. loss (cos. sim.) + on-manifold (VAE) + fine-tuning + combinations.

Fig. 8: **(Collages, pixels as input)** Accuracy (average over the 4 test sets) of **existing methods (first row)** and **ablations (second row)** for various hyperparameters and numbers of models. The only existing methods with non-trivial performance (*e.g.* (f)) require training at least an order of magnitude more models than ours (j).



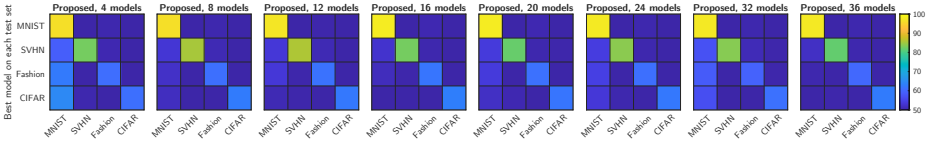
(a) Pixels as input. (b) ResNet features as input.

Fig. 9: **(Collages)** Accuracy of the proposed method (average over the 4 test sets) for various loss weights of the independence and on-manifold losses (with 12 models). **Performance is stable over a range of values** and both losses are important, as seen from the leftmost & lowermost cells. Performance is also repeatable: each cell reports a single run *i.e.*not averaged over multiple random seeds.

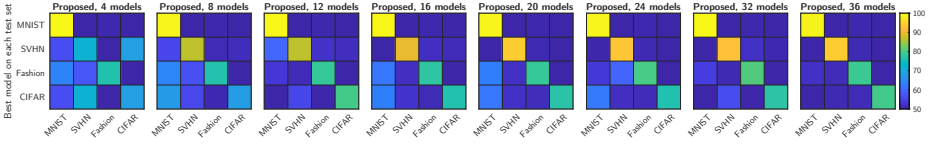


(a) Pixels as input. (b) Pixels as input + fine-tuning. (c) ResNet features as input.

Fig. 10: **(Collages)** Accuracy of the proposed method as a function of the number of models trained. Non-trivial accuracy is obtained on all 4 test sets with as few as 4 models. In line with our theoretical predictions, the accuracy decreases after >24 models, which is the recommended value based on the intrinsic dimensionality of the dataset (Section C).



(a) **Pixels** as input.



(b) **ResNet** features as input.

Fig. 11: **(Collages)** Accuracy on the 4 test sets (columns) of models with the best accuracy on each set (rows). The diagonal pattern indicates that the models specialize and learn different, non-overlapping features.

Table 6: Accuracy of the proposed method and ablations using ResNet features as input. The independence and on-manifold constraints are both important for high performance. There remains a small gap with the upper-bound accuracy, but we achieve drastically better results than the baseline.

Collages dataset, **ResNet** features as input (accuracy in %)	Best model on				
	MNIST	SVHN	Fashion	CIFAR-10	Average
Upper bounds (training on OOD data)	100.0	98.0	92.3	89.9	95.1
Baseline	99.8	49.9	50.7	49.9	62.6
Independence loss					
4 models	86.8	60.2	62.1	51.6	65.2
8 models	88.4	61.5	63.3	51.9	66.3
16 models	91.6	59.3	59.9	52.1	65.7
24 models	93.9	58.2	57.7	52.1	65.5
Independence loss + on-manifold constraint (VAE)					
4 models	99.6	71.7	76.1	68.3	78.9
8 models	99.2	86.8	76.3	67.8	82.5
16 models	99.5	90.0	77.5	76.1	85.8
24 models	**99.2**	**93.0**	**80.1**	**76.0**	**87.1**

I Additional results: WILDS-Camelyon17

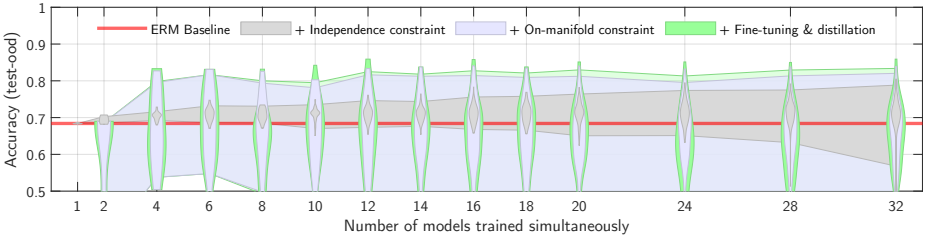


Fig. 12: Spread of accuracies on WILDS-Camelyon17 of models trained with different ablations of our method. The upper/lower bounds of the **shaded areas** show the highest/lowest accuracy of any model from one run, averaged over 6 seeds. The **violins** show the distribution of accuracies over all seeds (hence some values outside the shaded areas). **Take-aways.** The independence constraint (**gray**) produces a wide variety of models compared to the baseline (**red**). However, the highest accuracy in each run grows slowly with the number of models. With the on-manifold constraint (**blue**), the improvement is clearly larger and only requires a handful of models. The fine-tuning and distillation (**green**) bring an additional marginal improvement.

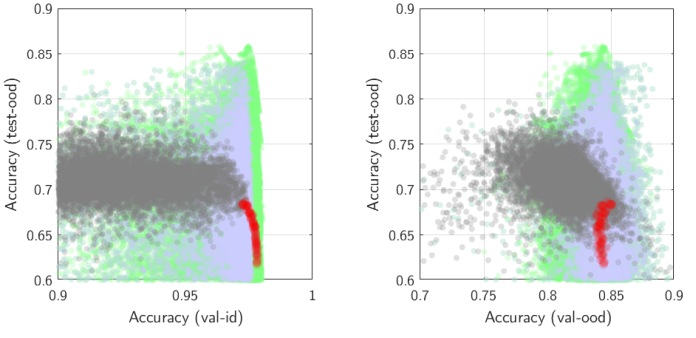


Fig. 13: On WILDS-Camelyon17, each model (same colors as Figure 12) achieves a different trade-off in performance on different splits, as seen with test-OOD *vs.* val-ID/val-OOD (left/right, respectively). Each dot represents one model at one training epoch. We plot every epoch since each model moves on these charts as training progresses. The proposed method (green) produces models with much better performance on all splits (toward the upper right) than the baseline (red).

Table 7: Best accuracy (%) on the test-OOD split of WILDS-Camelyon17 (average over 6 random seeds). We show the difference in performance between the baseline and various ablations of our method while training 12 models (as in Table 2), starting with each of the 10 pretrained models provided by the dataset authors [40]. **While the ranking of methods is roughly similar, the absolute accuracy (first and last rows) is highly variable across pretrained models.** A possible solution for eliminating this variability would be to apply our method during pretraining – at far greater computational expense than when training linear classifiers.

WILDS-Camelyon17	Pretrained model									
	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
ERM Baseline	68.4 ±0.1	78.0 ±0.1	73.3 ±0.0	60.4 ±0.1	78.3 ±0.0	78.5 ±0.1	64.4 ±0.1	74.2 ±0.0	71.9 ±0.2	60.2 ±0.1
+ Independence constraint	+6.2	+4.5	+0.8	+4.4	-1.5	+2.9	+1.3	+0.8	+4.1	+2.3
+ On-manifold soft regularizer, VAE	+11.9	**+7.1**	+0.6	+11.5	-1.0	+5.1	+3.6	+2.2	+6.2	+2.3
+ On-manifold hard projection, VAE	+7.9	+4.2	+2.2	+12.9	-0.3	**+7.4**	**+9.9**	+2.6	+7.1	**+4.5**
+ On-manifold soft regularizer, PCA	+10.6	+4.7	+2.6	+8.6	-0.4	+6.4	+7.4	+2.9	**+8.4**	+4.2
+ On-manifold hard projection, PCA*	+13.2	+5.8	+2.6	+12.2	+0.1	+5.3	+7.2	+3.4	+6.7	+3.4
(*) + Fine-tuning & distillation	**+14.1**	+6.3	**+2.8**	**+15.1**	**+0.9**	+5.7	+9.3	**+3.8**	+6.7	+3.5
(*) + Fine-tuning & distillation	82.5 ±2.4	84.3 ±2.1	76.2 ±0.4	75.5 ±2.1	79.3 ±0.4	84.2 ±0.6	73.8 ±2.5	78.0 ±1.8	78.6 ±1.0	63.7 ±0.4

J Experiments on visual question answering

In this section, we demonstrate the applicability of our method on a more complex task. We choose visual question answering (VQA) because it is notorious for dataset biases [74] that cause shortcut learning [13]. We train a simple architecture (details in Appendix G) on the GQA dataset [32] and report accuracy on the GQA and GQA-OOD [37] validation sets. We focus on binary questions to lighten the computational expense. As input the model, we use frozen 2048-dimensional image features (globally-pooled bottom-up features [3]). Our method is applied in the space of these features. In Table 8, we observe that our method produces models with slightly better accuracy on all test sets. The independence regularizer alone is not sufficient, most likely because it is unable to discover meaningful features in the large 2048-dimensional space. The on-manifold constraint does solve this difficulty. Interestingly, the individual models are better as well as a simple ensemble of all of these models. This suggests a benefit from the greater variety of features learned collectively by the models. We show in Figure 1 and in Appendix J that the models typically focus each on different regions of images (with grad-CAM-weighted visualizations as in [64]). It is important to note that our method is applied across channels of image features, and that the spatial diversity emerges naturally. We quantitatively verify this increase in diversity in Table 8 with the average Spearman rank correlation of grad-CAM scores across models.

Table 8: Application to visual question answering. Models trained with the proposed method achieve higher accuracy. They also show higher diversity in the image regions they rely on (last column, lower correlation of grad-CAM scores).

GQA yes/no (accuracy in %)	N. of models	GQA Val. (best)	GQA Val. (ensemble)	GQA-OOD Val-head (best)	GQA-OOD Val-tail (best)	Grad-CAM rank corr.
Baseline	3	67.9	69.3	70.4	66.5	0.68
Baseline	16	68.6	68.9	71.2	67.3	–
+ Independence	3	67.6	69.2	70.5	67.1	0.57
+ Independence	4	67.6	69.4	70.1	66.2	–
+ Independence	8	67.8	70.0	70.7	66.7	–
+ Independence	12	68.1	70.3	70.6	69.8	–
+ Independence	16	68.1	**70.5**	71.1	69.6	–
+ Ind. + on-manifold	3	68.7	69.7	71.9	**72.5**	0.59
+ Ind. + on-manifold	4	69.2	70.4	**72.9**	70.5	–
+ Ind. + on-manifold	8	68.8	70.0	71.5	69.1	–
+ Ind. + on-manifold	12	69.0	70.2	72.6	69.4	–
+ Ind. + on-manifold	16	**69.3**	70.3	72.4	71.5	–

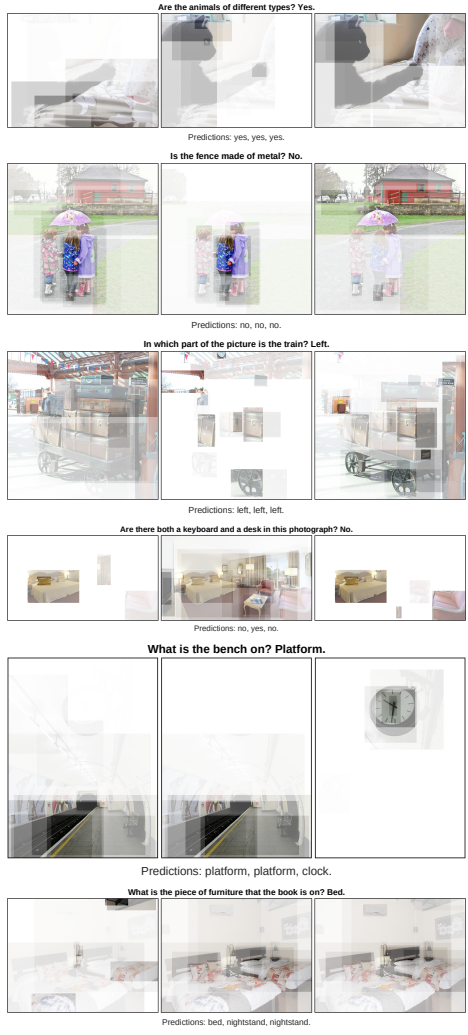


Fig. 14: Examples from the GQA validation set. We show the input question, ground truth answer, and predicted answer from 3 models trained with our method. The input images are weighted with grad-CAM scores over object detections. The models often predict the same answer while focusing on different regions. This diversity in spatial locations emerges naturally: our independence constraint is applied across the *channels* of globally-pooled visual features.

The experiments above use only the binary (yes/no) question of the GQA dataset [32]. We repeated them with the full dataset. The setup, model, and hyperparameters are unchanged. The models' accuracy remains close to that of the baseline (Table 9) even when the independence constraint induces the models to focus on different input features. In Figure 14, we include examples from the validation set that illustrate this effect. Contrary to Section J, we did not observe improvements in accuracy when training three models with our method. Our ongoing work is exploring the range of hyperparameters (larger number of models, different regularizer weights) to further investigate these observations.

Table 9: Experiments on the full GQA dataset. Models maintain very similar accuracy across the board while focusing on different image features.

GQA full (accuracy in %)	N. of models	GQA Val. (best)	GQA Val. (ensemble)	GQA-OOD Val-head (best)	GQA-OOD Val-tail (best)	Grad-CAM rank corr.
Baseline	3	49.7	52.3	53.3	**35.6**	0.1404
With PCA manifold model						
+ Independence	3	50.9	**52.7**	54.4	35.0	0.2230
+ Ind. + on-manifold	3	**51.0**	52.5	54.7	35.2	0.1868
With PCA manifold model						
+ Independence	3	50.7	51.7	55.1	34.0	0.1626
+ Ind. + on-manifold	3	50.7	51.6	**55.8**	33.8	**0.1012**