Show Me What You Don't Know: Efficient Sampling from Invariant Sets for Model Validation

Armand Rousselot
Computer Vision and Learning Lab
Heidelberg University
armand.rousselot@iwr.uni-heidelberg.de

Joran Wendebourg
Computer Vision and Learning Lab
Heidelberg University

Ullrich Köthe
Computer Vision and Learning Lab
Heidelberg University

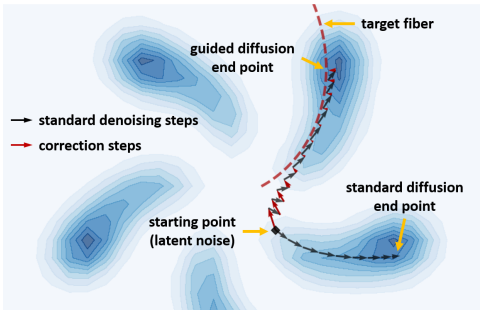
Abstract

The performance of machine learning models depends on discovering features in the data that are relevant to solving the given task while ignoring irrelevant variation. To check and visualize whether this is the case, we propose 'invariance auditing', a diagnostic tool to analyze feature extractors. Invariance auditing generates diverse inputs which all yield the same features as a given query input, and are therefore treated as identical by the model. Using external knowledge about the task at hand, this enables assessing if the learned invariances do or do not conform to the task's required semantics. Unlike existing work where a dedicated generative model is trained for each feature extractor, our algorithm is training-free and exploits a pretrained diffusion or flow-matching model to sample invariant inputs. Our fiber loss – which penalizes feature mismatch – guides the denoising process so the output shares the same representation as the query input. This replaces days of training with a single guided generation procedure at comparable quality. Experiments on popular datasets and model types demonstrate that our auditing method reveals invariances spanning very desirable and concerning behavior. For instance, it detects cases where Qwen-2B places patients with situs inversus (heart on the right side) in the same fiber as typical anatomy. Our code is available at https://github.com/vislearn/InvarianceAuditing.

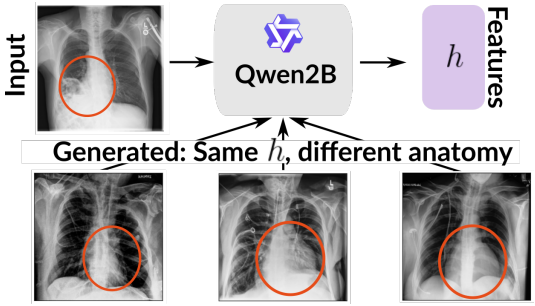
1 Introduction

Deep representation learning has seen massive success across applications [Bengio et al., 2013], yet understanding and explaining learned representations remains an open problem [Doshi-Velez and Kim, 2017, Samek et al., 2017]. The most prominent avenues strive to assign semantic meaning to the learned features [Ribeiro et al., 2016, Yang et al., 2022, Cunningham et al., 2023] or invert feature representations of given models, in order to inspect which information they encode [Har-Shalom and Weiss, 2025, Mahendran and Vedaldi, 2015]. However, to fully understand a learned representation, it is equally important to analyze what it does *not* encode. We refer to diagnostic tools for this purpose as *invariance auditing*, which reveals which features of the data a model ignores, surfacing systematic biases, shortcuts, and failure modes in its representations [Sammani et al., 2023, Bordes et al., 2021, Rombach et al., 2020]. Given a fixed model, the aim is to generate diverse natural inputs that are all assigned the same representation. Invariance auditing complements the analysis of adversarial robustness [Goodfellow et al., 2015]: while the latter asks if very similar inputs can have vastly different representations, we search for very different inputs that have the same representations.

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



(a) Black arrows represent the ordinary denoising steps in a flow-matching model. Alternating black (FM) and red (correction) arrows indicate the guided trajectory to the dashed target fiber.



(b) Given a case of "situs inversus", where the heart is located on the right side of the body, our invariance sampling – guided by the Qwen-2B vision model – finds several images that yield the same features but have the heart on the left side.

Figure 1: Illustration of invariance auditing using guided pretrained generative models.

The *fiber* (or invariant set) of the model ϕ to be analyzed (called the *subject model*) at representation value h is the equivalence class defined as the set of inputs x, that share the representation $\phi(x) = h$. The pioneering work of Rombach et al. [2020] formulated invariance auditing as training a conditional model that samples from the fiber distribution $p(x|h)$ (called the *fiber model*). While several works have since built on this approach [Bordes et al., 2021, Eulig et al., 2024] it comes with two major drawbacks: For every subject model a new fiber model has to be trained entirely from scratch, and there is no control over how closely samples lie to the target fiber.

We address these challenges in two complementary ways. First we propose to leverage the difference between sample and target representation, which we call *fiber loss*, as a regularizer for fiber model training. It was first introduced as a standalone optimization objective in Mahendran and Vedaldi [2015] and we show that fiber models trained with fiber loss are more reliable in terms of their achieved level of invariance. To this end we construct a benchmark based on color MNIST from Rombach et al. [2020] where the invariant sets and the distributions on them are known exactly. Second, we propose a training-free method for invariance auditing. Rather than retraining fiber models, we steer the denoising path of a pretrained diffusion/flow-matching model using the fiber loss to penalize feature mismatch (Figure 1). We show that we can create diverse, high-quality fiber samples on various subject models and datasets, making invariance auditing practical at scale. To our knowledge, this is the first time guidance of pretrained models has been exploited as a way to examine invariances. Together, as shown in Table 1, these methods are complementary: Guided invariance auditing enables rapid iteration during model development, while conditional fiber models support large-scale statistical analysis.

Table 1: Comparison of different methods for invariance auditing. We give a "✓" if a method fulfills a property and a "○" if a property is approximately fulfilled.

	Contract to fiber	Converges to fiber distribution	Fast setup	Fast sampling
Conditional models	○	✓	✗	✓
+ Fiber loss (ours)	✓	✓	✗	✓
Guidance (ours)	✓	○	✓	✗

In summary our contributions are:

- The fiber loss as a regularizer to control invariance levels in fiber model training.
- A training-free invariance auditing method via guidance of pretrained generative models.
- A controlled benchmark with precisely known fiber distributions, comparing a comprehensive range of fiber model designs.
- Large-scale invariance auditing of popular vision models on natural and medical images, confirming known and revealing previously unreported failure modes.

2 Related Work

Existing work can be broadly categorized into model inversion techniques, that construct an image from a given representation and generative models which sample from the fiber.

Model inversion techniques. The work of Erhan et al. [2009] introduces gradient ascent in the input space to maximize the activations of given specific neurons. Following this work, several attempts have been made to invert a given set of activations (i.e. intermediate representations) using various techniques, including gradient descent [Mahendran and Vedaldi, 2015], (de-)convolutional networks [Zeiler and Fergus, 2014, Dosovitskiy and Brox, 2016b] and GANs [Dosovitskiy and Brox, 2016a]. Output matching model inversion attacks apply these concepts to reconstruct input data information from model outputs [Yang et al., 2025]. Several works have attempted to curate sets of natural adversarial images for certain domains in a model-agnostic way [Hendrycks et al., 2021, Rahmanzadehgervi et al., 2024, Mayer et al., 2025], but the size and domain coverage of these sets remains limited. More recently, Sammani et al. [2023] fit a local linear model, similar to LIME [Ribeiro et al., 2016], to measure the level of sensitivity to changes in each pixel of the image. Model inversion seeks to invert a subject model deterministically, while the goal of invariance auditing is to efficiently explore a diverse, representative set of samples from the fiber.

Invariance auditing based on generative models. The pioneering work of Rombach et al. [2020] first formulated the task of sampling from fibers as a conditional generative problem, and utilized coupling-based normalizing flows to draw such samples. Eulig et al. [2024] utilize this approach to analyze the invariances of CT image denoisers. Bordes et al. [2021] replaced the generative model by a conditional diffusion model, which leads to better generative quality. They then compare the invariances of self-supervised models and thoroughly examine properties of the backbone and projector embeddings. The works of Appalaraju et al. [2020], Ericsson et al. [2021], Zhao et al. [2020] explore different approaches by utilizing deep image priors [Ulyanov et al., 2018]. As shown in Bordes et al. [2021] however, this leads to low quality samples. While these works lay the foundation for invariance auditing, it has not been widely adopted as a tool for trustworthy model design and diagnostics. We argue that this is largely due to the initial cost of training a conditional generative model for every subject model. We fill this gap by presenting an efficient method for invariance auditing using pretrained, unconditional diffusion models.

Learning and controlling invariances. A complementary line of work focuses on explicitly encouraging or discouraging models to rely on certain invariances during training (e.g. Arjovsky et al. [2019]). An extended discussion of this topic can be found in Section B.3. Invariance auditing serves a complementary visualization and diagnostic purpose: It identifies undesirable or lacking invariance structure in given subject models. Invariance control can then be leveraged to improve the models, and repeated invariance auditing will verify success. In Section 4.4 we illustrate this workflow on a dataset from Arjovsky et al. [2019]. Invariance auditing is used to make reliance of a baseline model on a known spurious correlation in the data apparent. Using invariance control to eliminate this reliance, invariance auditing confirms that the resulting classifier ignores the spurious correlation.

3 Invariance Auditing and Fibers

Given a subject model ϕ and a representation created by that model $\phi(x) = h$ the goal of invariance auditing is to sample from the fiber (or invariant set)

$$\phi^{-1}(h) := \{x : \phi(x) = h\}. \tag{1}$$

This task is often further narrowed down to conditional generative modeling with the target distribution

$$p(x|h) \propto p(x)\delta(\phi(x) - h) \tag{2}$$

The underlying distribution $p(x)$ can be freely chosen and does not have to be the same as the dataset distribution that the subject model was trained on. For example in a safety relevant context, $p(x)$ may be chosen to explicitly contain many high risk datapoints. Using a conditional generative model, one can inspect samples from the learned density $p_\theta(x|h)$, which intuitively answer the question:

Which inputs have the same representation and are thus indistinguishable for the subject model?

Due to limitations in numerical accuracy and convergence it is generally impossible to sample $\tilde{x}$ that achieves exactly $\phi(\tilde{x}) = h$ when h is continuous. Likewise, we generally cannot rely on having access to multiple samples from any single fiber in our dataset. Based on the above characterization of the fiber, we define two separate goals of an accurate fiber model: **Fidelity** and **Consistency**.

Fidelity describes the ability of the fiber model to produce samples close to the fiber. Following Bordes et al. [2021], we evaluate the fidelity of fiber models using the discrepancy between the target representation and the representation of fiber samples, which we call *fiber loss*

$$\mathcal{L}_{\text{fiber}} = \text{dist}(\phi(x), \phi(\tilde{x})). \tag{3}$$

One could choose any metric to measure the difference in representations, for the remainder of this paper, we use the Euclidean metric $||\phi(x) - \phi(\tilde{x})||_2^2$. In Section C.2 we show that this is a robust choice, which generally implies low fiber loss in other metrics as well. The fiber loss can also be used as an efficient objective for training and guidance of fiber models, which was not explored in prior work. In Section 4.1 we will show that it is necessary to include this additional loss term to achieve higher levels of fidelity. The meaning of the fiber loss depends on the scale of the representations. For evaluating fidelity we provide the fiber losses achieved with nearest neighbors from the dataset and by augmenting the original image as a point of comparison. Achieving fiber losses lower than the nearest neighbor means we can gain insights from fiber samples about the subject model that we could not have gained from the dataset alone.

Consistency describes the ability of the fiber model to capture the desired distribution on the fiber $p(x|h)$. As described earlier, one generally has neither access to samples, nor likelihood evaluation of this density. Instead, we estimate consistency by sampling from the learned marginal density

$$p_\theta(x) = \int p_\theta\big(x|h = \phi(x')\big)\, p(x')\, dx' \tag{4}$$

We can compare $p_\theta(x)$ and $p(x)$ via various sample-based metrics, e.g. FID. In Section 4.1 we construct an example, where the $p(x|h)$ is explicitly known, which allows us to benchmark architectures and training archetypes in terms of their KL-divergence on the fiber.

We emphasize that neither fidelity nor consistency can be falsely improved by a bias in the fiber model, which is important: A bias in the fiber model can result in fiber samples that have the wrong distribution on the fiber – or do not lie on the fiber at all. Fidelity and consistency verify that the obtained fiber samples are correct, or expose the bias in the fiber model if present.

3.1 Invariance Auditing using Conditional Generative Models

Starting with Rombach et al. [2020], invariance auditing has been formulated as a conditional generative modeling problem. Different generative paradigms have been used for invariance auditing, yet their relative performance in terms of fidelity and consistency has not been systematically evaluated. In Section 4.1 we comprehensively benchmark different models for invariance auditing on a modified version of color MNIST [Rombach et al., 2020], where the fiber distribution is explicitly known.

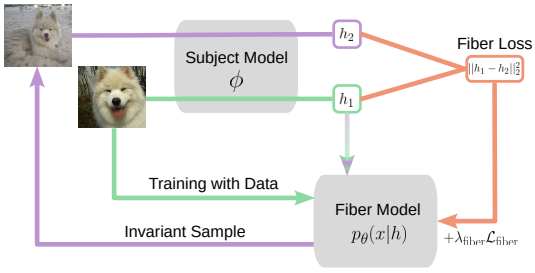


Figure 2: Conditional invariance auditing: The fiber model is trained to generate images conditioned on subject model representations h_1, producing samples with the same representation. The fiber loss verifies this and supplements training.

In this work we propose to use the fiber loss (Equation (3)) during training for improved fidelity; Figure 2 gives an overview of the workflow. Generative models with a fast sampling process can incorporate it at little additional cost, and as we show in Section 4.1, doing so produces samples significantly closer to the target fiber. We also compare full-dimensional models to injective models, which are specifically crafted for lower-dimensional manifolds. The intrinsic dimension of the fiber may differ from that of x. More precisely, assuming regularity of the subject model $\dim x - \text{rank}(D\phi) = \dim \phi^{-1}(h)$. We show that injective models offer improvements especially at high regularization strength (see Section 4.1).

3.2 Invariance Auditing using Unconditional Diffusion Models

One major drawback of training conditional generative models for invariance auditing is that for every subject model a new fiber model has to be trained from scratch. Depending on the task, training a conditional fiber model can even be more costly than training the subject model itself. Therefore, it is desirable to enable the use of large, pretrained generative models to create fiber samples, which are not conditioned on the subject model representation. We achieve this by utilizing guidance, specifically non-linear diffusion trajectory matching (NDTM) [Pandey et al., 2025]. The goal of NDTM is to guide the sampling process of a pretrained diffusion model to minimize some terminal cost function $\mathcal{L}_{\text{terminal}}(x_0)$ at the final sampling timestep $t = 0$. In our case, we set $\mathcal{L}_{\text{terminal}}$ as the fiber loss of Equation (3) in order to only obtain samples from the subject model fiber. Since this loss can only be computed at time $t = 0$, NDTM approximates

$$\mathbb{E}[\mathcal{L}_{\text{terminal}}(x_0, h)] \approx \mathcal{L}_{\text{terminal}}(\mathbb{E}[x_0|x_t], h) = ||\phi(\mathbb{E}[x_0|x_t]) - h||_2^2 \tag{5}$$

via Tweedie's formula in a simulation-free manner. To inject guidance of strength γ_t into the sampling process, a correction to the noisy sample at each timestep u_t is made, where the guided diffusion process alternates between a denoising step and a sample correction step

$$\hat{x}_t = x_t + \gamma_t u_t \tag{6}$$
$$x_{t-1} = \text{DenoisingStep}(\hat{x}_t). \tag{7}$$

In every sample correction step, u_t is obtained by performing N optimization steps via gradient descent to minimize the following loss function:

$$\mathcal{L} = \mathcal{L}_{\text{terminal}} + \kappa_t \mathcal{L}_{\text{control}} + \tau_t \mathcal{L}_{\text{score}} \tag{8}$$

$\mathcal{L}_{\text{control}}$ and $\mathcal{L}_{\text{score}}$ are both regularizers that penalize the magnitude of u_t and the deviation from the unguided score respectively. Using this technique we can create samples from the fibers of a subject model using a pretrained, unconditional diffusion model, without the need to perform any finetuning. This is, to our knowledge, the first time pretrained model guidance has been proposed for analyzing fibers. We provide additional details about regularizers and hyperparameter choices in Section C.

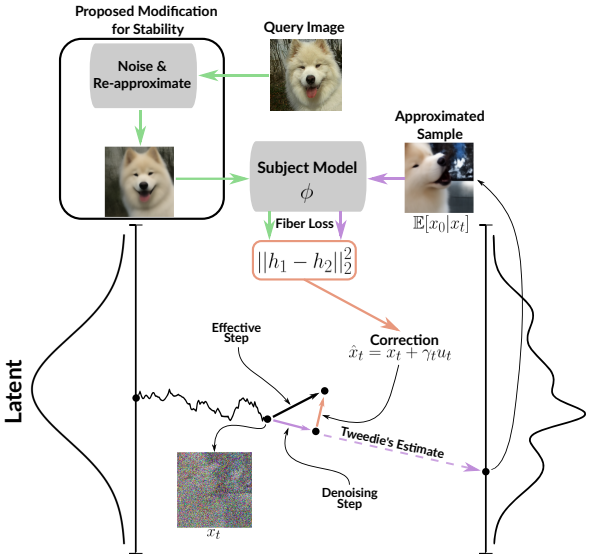


Figure 3: Guided invariance auditing: At each denoising step, Tweedie's estimate extrapolates the sample to the final step; the fiber loss against the target representation then corrects the sample. No finetuning or training is required.

For our experiments on ImageNet, we notice that the generation process can sometimes be unstable, leading to oversaturated and overtexturized images. The features h are potentially computed from detailed, local structures of x in the data space. However, Tweedie's estimate $\phi(\mathbb{E}[x_0|x_t])$ only reconstructs coarse structures at larger t. Assuming access to x, we instead propose the terminal loss

$$\mathcal{L}_{\text{terminal}}(x_t, x_t') = ||\phi(\mathbb{E}[x_0|x_t]) - \phi(\mathbb{E}[x_0|x_t'])||_2^2 . \tag{9}$$

Where, $x_t' \sim q(x_t'|x_0 = x)$ is the noising process of the query image to time t. Effectively, we noise and apply Tweedie's estimate to x to obtain an approximation with the same level of detail as the estimate based on x_t. We find this provides u_t with a training signal that is more aligned with the semantic content of the current generation stage. In Section C we compare samples obtained with our modified algorithm and standard NDTM. We present an algorithm of NDTM for fiber sampling with the definition of the regularizers and our modification in Algorithm 1 and show an overview in Figure 3.

Algorithm 1 Fiber Sampling with NDTM Guidance

Input: Pretrained diffusion model, Subject model ϕ, Query image x, Guidance strengths γ_t, Regularizer weights κ_t, τ_t, Learning rates ρ_t, Number of gradient steps N
Output: Fiber sample $\tilde{x}$ with $\phi(\tilde{x}) \approx \phi(x)$
$h \leftarrow \phi(x), \quad x_T \sim \mathcal{N}(0, \mathbf{I})$
for $t = T, \dots, 1$ **do**
  $h_t \leftarrow h$ {Modification: $x'_t \sim q(x'_t | x_0 = x), \quad h_t \leftarrow \phi(\mathbb{E}[x_0 | x'_t])$}
  $u_t \leftarrow 0$
  $\hat{x}_t \leftarrow x_t + \gamma_t u_t$
  for $i = 0$ to $N - 1$ **do**
    $\mathcal{L}_{\text{terminal}} \leftarrow \|\phi(\mathbb{E}[x_0 | \hat{x}_t]) - h_t\|_2^2$
    $\mathcal{L}_{\text{control}} \leftarrow \|u_t\|_2^2$
    $\mathcal{L}_{\text{score}} \leftarrow \|\nabla \log p(\hat{x}_t) - \nabla \log p(x_t)\|_2^2$
    $\mathcal{L} \leftarrow \mathcal{L}_{\text{terminal}} + \kappa_t \mathcal{L}_{\text{control}} + \tau_t \mathcal{L}_{\text{score}}$
    $u_t \leftarrow u_t - \rho_t \nabla_{u_t} \mathcal{L}$
    $\hat{x}_t \leftarrow x_t + \gamma_t u_t$
  end for
  $x_{t-1} \leftarrow \text{DenoisingStep}(\hat{x}_t)$
end for
$\tilde{x} \leftarrow x_0$
return $\tilde{x}$

4 Experiments

4.1 Color MNIST Benchmark

To benchmark different methods under known invariances, we construct a setup where the fiber and its distribution are known. Based on the Color MNIST dataset from Rombach et al. [2020] (not to be confused with the like-named dataset from Arjovsky et al. [2019], which we use in Section 4.4), we randomly colorize the background with a distribution $p(c_0)$ and derive a corresponding foreground color c_1 deterministically from it, see Appendix B.1 for details. The subject model turns the color images back into grayscale, and a proper fiber model must learn the distribution of background colors as well as their deterministic relationship to foreground colors. We can thus measure both quality criteria independently: High fidelity requires preservation of digit shapes and the correct foreground/background relationship, which can be measured by fiber loss. Consistency is high when the known $p(c_0)$ is reproduced, which can be measured by KL-divergence.

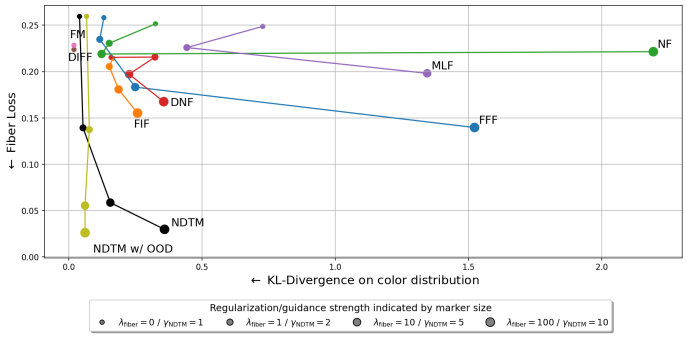


Figure 4: Benchmark of different invariance auditing methods and architectures on color MNIST. Fiber samples are evaluated by KL-divergence of background color distribution and the fiber loss (lower is better for both). Each colored line represents one method, with the marker size representing the strength of the training regularization λ_{fiber} or guidance γ_{NDTM}.

Within this setup, we perform a comparison between the design choices listed in Section 3. We train various classes of conditional fiber models and their injective variants where available. Architectures, as well as additional details and metrics are listed in Section B.1. For NDTM we additionally examine an OOD setting where the correlation between fore- and background color is removed from the diffusion model (i.e. fiber model) training data. Even though the background color distribution during training is $p(c_0)$ for both diffusion models, NDTM w/ OOD may also adjust c_0 during guidance to yield the invariant color combination between fore- and background.

In Figure 4 we show the results of these experiments in terms of fiber loss vs. KL-divergence of the color distribution. Regarding the different generative paradigms we see significant differences in KL-divergence, with diffusion/flow matching models clearly coming out on top. Without regularization conditional fiber models all achieve similar values of fiber loss. At their baseline, (most) injective models perform marginally better in terms of fiber loss, and are more robust to high regularization parameters λ_{fiber}. Trained both with and without the correct relationship between colors, NDTM can achieve much better fidelity. The consistency compared to conditional diffusion models is only slightly worse with the right settings for the in-distribution case, the consistency of NDTM w/ OOD is not affected by the guidance strength showing that even a diffusion model pretrained on a broader distribution can serve for guided fiber sampling. Overall, diffusion-based conditional models achieve the best consistency, while NDTM attains substantially higher fidelity with minimal loss in consistency in the right settings.



(a) Fibers of DINOv2

(b) Fibers of InceptionV3

Figure 5: Fiber samples drawn with NDTM on ImageNet. The first three columns show the original, nearest neighbor and the original with DINOv2 training augmentations. Best viewed zoomed in.

4.2 Invariances on ImageNet

We showcase the scalability of NDTM by analyzing the invariances of pretrained models on ImageNet. We follow a similar setup to Bordes et al. [2021] using a diffusion model trained on ImageNet from Dhariwal and Nichol [2021] to create samples that are invariant under the representations of DINOv2 ViT-B/14 backbone. Crucially, using guidance allows us to perform this sampling with a pretrained diffusion model, without any fine-tuning. In the fiber samples in Figure 5a we observe that image orientation and main object position are also preserved, confirming the finding of Bordes et al. [2021] that SSL-trained model backbones are not necessarily invariant to the augmentations used for training. In comparison to nearest neighbors from the dataset and originals with DINOv2 training augmentations our fiber samples achieve high fidelity. We provide fiber losses of all subject models and large image datasets in Table 5, which confirms that NDTM can contract to fibers even for high-dimensional, complex representations. The reported FIDs in Table 2 confirm that NDTM generated samples still follow the ImageNet distribution to similar quality as the unconditional base model. Note that our goal is *not to improve* the quality of generated images, but inspect the invariances of the subject model with the help of a good, pretrained generative model. These results confirm that NDTM enables large-scale, training-free inspection of invariances while preserving generative quality.

Table 2: FIDs and top-5 accuracy of a pretrained ConvNeXt-L on fibers of DINOv2 and InceptionV3. We measure FID both with a TensorFlow (TF) and PyTorch (PT) Inception model. Baselines show the unconditional diffusion FID and classifier performance on ImageNet.

Subject Model	Fiber Model	TF FID	PT FID	Classifier top-5 accuracy
Baseline on ImageNet		26.21	-	95.62%
DINOv2	NDTM	28.34	29.68	$81.24\% \pm 1.29$
InceptionV3	NDTM	33.52	3.81	$51.48\% \pm 1.89$

Next we inspect InceptionV3 [Szegedy et al., 2016] to ask the question: What changes in images are not captured by FID and could lead to low values despite distribution shift? To this end, we sample from the fiber of the PyTorch pretrained InceptionV3 model and compare FIDs computed both with this model in PyTorch and the TensorFlow

InceptionV3 model that was originally used in Heusel et al. [2017] in Table 2. When computing the FIDs for the DINOv2 fiber, both networks yield slightly different, but similar FID values. In contrast, sampling from fibers of the Inception network itself reveals that FID can be artificially low when errors of the generator align with the fibers of the metric network, exposing the blind spots of such representation-dependent evaluation. Inspecting these blind spots with the help of fiber samples in Figure 5b and the top-5 accuracies of a pretrained classifier applied to the fiber samples shows that members of the InceptionV3 fiber still have high visual quality, but potentially confuse different classes.

4.3 Invariances on Cue Conflict Dataset

We turn our attention towards a case where the target representation is created by an out-of-distribution input. We compare the representations of DinoV2 ViT-B/14 backbone and those after the final pooling layer of a ResNet50 model, pretrained on ImageNet, on the cue conflict dataset from Geirhos et al. [2018], which stylizes images with different textures to decouple them from object shape.

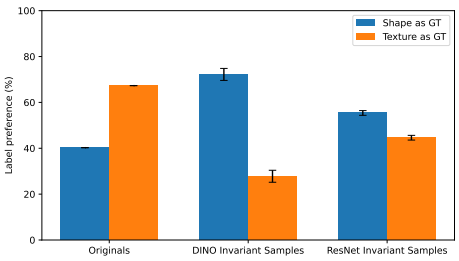


Figure 6: Preference for texture (orange) and shape (blue) labels of ConvNeXt-L on original data and subject model fibers. Cases where none of the two labels were predicted or both labels coincide are filtered out.

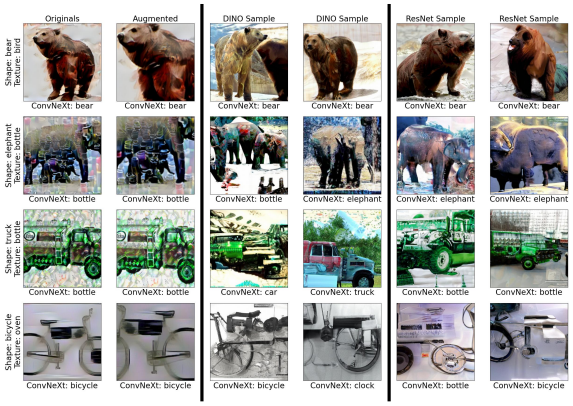


Figure 7: Fiber samples of both subject models on the cue conflict dataset. Below each image we show the label predicted by a ConvNeXt-L. All models are trained on ImageNet. Best viewed zoomed in.

We use the same ImageNet diffusion model from the previous section for NDTM to obtain the fiber samples shown in Figure 7. Under each image we show the label predicted by an ImageNet pretrained ConvNeXt-L [Liu et al., 2022]. Although in this OOD setting fiber losses are higher than on ImageNet, they remain well below nearest neighbors. We can see that samples from both invariant sets differ significantly from ImageNet, indicating that the subject models are not invariant to the distribution shift introduced by texture stylization. Inspecting the fibers more closely we find that DINOv2 representations generally preserve global shape information, while ResNet representations capture local shape information, which is in line with Burgert et al. [2025]. We quantify these biases in terms of the label preference used in Geirhos et al. [2018].

The ConvNeXt-L classifier is applied to originals and fiber samples to compare the accuracy when using the shape and texture labels of each image respectively. Specifically, in cases where the classifier agrees with exactly one of the two labels we report the percentage of times the shape or texture label was chosen respectively as *label preference* in Figure 6. As reported by Geirhos et al. [2018], ImageNet-trained convolutional models like ConvNeXt and ResNet are also biased towards texture, but we find that this bias is not yet expressed in the final pooling layer of ResNet-50. This indicates that the shape information is not lost in the fiber, rather it is not being chosen as the basis for classification by the last, fully-connected layer. In contrast, we find that DINOv2 representations exhibit a bias towards encoding image content by its shape. In fact, while the classifier recovers the shape label 25% of the time on its own, applied to DINOv2 fibers this increases to 35%.

4.4 Combining Invariance Auditing with Invariance Learning and Control

We demonstrate how invariance auditing and invariance control can work together to first expose and then eliminate a model's reliance on spurious features. We inspect two subject models on an MNIST variant from Arjovsky et al. [2019], which we call

"causal MNIST". The task is to predict whether a colored MNIST digit is high ($\geq$ 5) or low ($<$ 5). Color is strongly correlated with the label during training, but at test time this correlation is flipped; models that rely on this predictor therefore do not generalize. We train the fiber model for guidance with the training correlation between label and color, which determines $p(x)$. The query images are obtained using the reversed color correlation, which determines $p(h)$. If color affects the representation h the fiber samples $p(x|h)$ should have a different color correlation than $p(x)$. This is what invariance auditing detects in a naively trained subject model (ERM), which indicates that it takes the color shortcut. We then retrain the subject model using a method designed to control invariances (IRM), which removes this false reliance on a spurious correlation. Invariance auditing then confirms that the resulting model correctly uses only the digit shape to predict the label (see Figure 8). Full details are given in Section B.3.2.

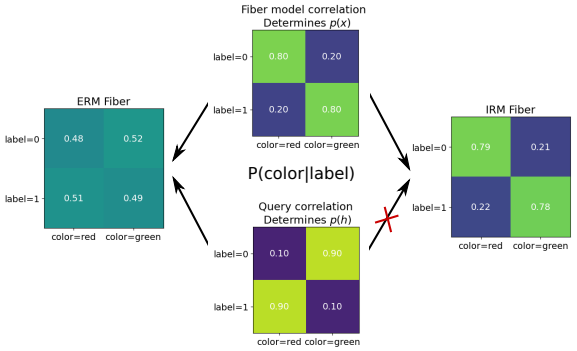


Figure 8: Color–label correlation in fiber samples on causal MNIST. The fiber model is trained with 80% color–label correlation, determining $p(x)$. Query representations are obtained from inputs with reversed correlation, determining $p(h)$. A shift in correlation on the fiber away from $p(x)$ indicates that the subject model prediction is not invariant to color.

4.5 Invariances on CheXpert

To inspect fibers in medical image analysis, we train two classifiers of different quality on five classes of the CheXpert dataset [Irvin et al., 2019]. For the first classifier we freeze the BiomedClip foundation model [Zhang et al., 2023] and train a classification head. The second classifier is a ConvNeXt trained end-to-end. Averaged across the five classes the classifiers achieve an AUROC of 0.73 and 0.85 respectively; further details in Section B.4. The fiber loss is computed at the classifier logits, which yields images with the same class probabilities. We measure classifier agreement by drawing fiber samples for originals where both classifiers are correct, then labeling them with the other model.

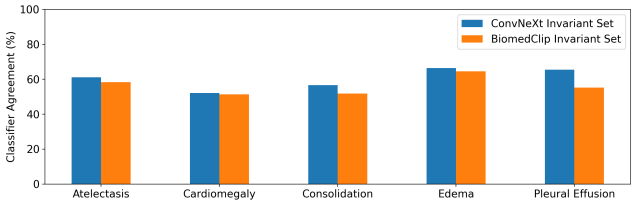


Figure 9: Cross-model agreement on CheXpert invariances. For images both models classify correctly, we measure how often they assign the same label to each other's fiber samples.

We plot the rate of agreement as the percentage of fiber samples where both classifiers stay with the original label in Figure 9. As we can see, models disagree on their fibers indicating that at least one of the models has not learned desirable fibers. In Figure 10 we show samples from the fiber, where the ground truth label for cardiomegaly on the original was negative, but the other classifier now predicts a positive label. We find that the ConvNeXt classifier detects critical changes in the BiomedClip fiber, while the BiomedClip classifier mostly flags images with unclear changes.

In Section B.4 we repeat these experiments for two identically trained end-to-end classifiers. This shows how fiber sampling enables targeted stress-testing of representations in safety-critical settings. In particular, the observed disagreement between classifiers highlights that invariances can be model-dependent and that undesirable blind spots may only become apparent when viewed from an external reference. This suggests a practical workflow for invariance auditing: training a complementary reference model, or leveraging a pretrained one provides an additional signal to detect semantic changes in fiber samples and to identify invariances that are undesirable for the task at hand.

In the recently introduced natural medical adversarial images [Mayer et al., 2025] the authors show that, when prompting vision-language models (VLMs) like Qwen-2B with images that show rare medical cases (e.g. anatomy), they will give answers that align with typical anatomy, disregarding the information in the images. Our goal is to find out whether this decision is made by the language model or the information is already lost at the vision model embedding.

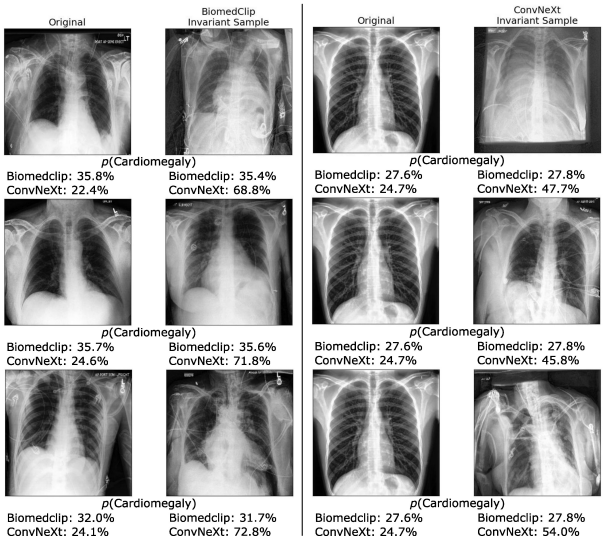


Figure 10: Originals with negative cardiomegaly diagnosis and samples of the fiber from BiomedClip (left) and ConvNeXt (right), obtained by our search for contradicting samples.

To this end, we create fiber samples of "situs inversus" cases from Mayer et al. [2025] where the heart is located on the right side of the body (from the patient's perspective). Our subject model is the Qwen-2B pretrained vision encoder. Fiber samples in Figure 11 suggest that positional information of the heart is weakly encoded by the vision encoder. This explains the finding of Mayer et al. [2025], that even tuning the prompt to take into account rare medical occurrences, the VLM was unable to give the correct response. Beyond this, the experiment illustrates the flexibility of invariance auditing: it enables the targeted analysis of rare cases that would otherwise be overlooked by population-level evaluation. Without requiring large-scale datasets to detect systematic failures, we verify for all four situs inversus images that typical anatomy occurs in the fiber – a failure mode that aggregate metrics would be unlikely to surface.

5 Conclusion

In this work we presented two complementary improvements for invariance auditing. On the one hand, the contraction of conditional generative models to the fiber is improved using our fiber loss, increasing their reliability. On the other hand NDTM-based guidance produces invariant samples without any model training. Using this framework, we show that we can rediscover and visualize several findings from literature, including texture bias of CNNs and anatomical blind spots of VLMs. There remains a trade-off between spending compute on training a conditional fiber model and optimizing the cost function during inference when using guidance, making the two methods complementary. This naturally suggests a usage pattern – using NDTM for rapid iteration during model development and conditional fiber models for large-scale statistical analysis. Combining both is a promising future direction.

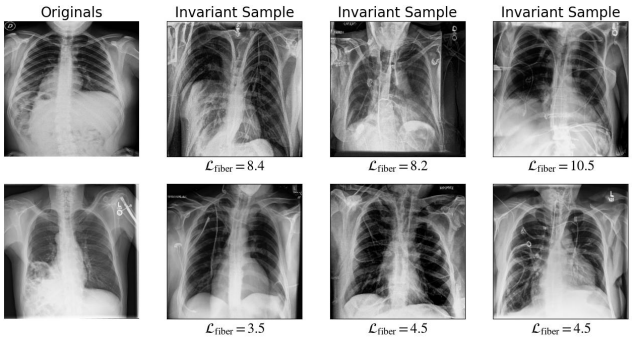


Figure 11: Invariances of the Qwen-2B vision model on a "situs inversus" image (shown on the left). The atypical anatomy does not appear in the selected samples.

With this expedited process we demonstrate how invariance auditing can be integrated into the design and evaluation of trustworthy models. Our experiments illustrate its merits by revealing a wide range of different invariances. Invariance auditing can give more detailed insights into the sources of known biases, at the same time it can also expose unknown failure-cases in modern vision and self-supervised models, such as omission of clinically relevant features in classifier representations. As such, invariance auditing presents a unified solution for exposing model blind spots. While it does not by itself prescribe how to remove them, it provides the critical first step by making them visible and measurable – a practical foundation for iterative development of more robust and trustworthy models.

Acknowledgments and Funding Transparency Statement

We would like to thank Elias Eulig for support and fruitful discussions, as well as Leon Mayer, Piotr Kalinowski and Lena Maier-Hein for providing swift, early access to the medical adversarial examples. This work is supported by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy EXC-2181/1 - 390900948 (the Heidelberg STRUCTURES Cluster of Excellence). It is also supported by the German Federal Ministry of Education and Research (BMBF) (project EMUNE/031L0293A). AR acknowledges support by the Carl-Zeiss-Stiftung (Projekt P2021-02-001 "Model-based AI"). The authors acknowledge support by the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) through grant INST 35/1597-1 FUGG. We ran experiments on NVIDIA A100 GPUs of the bwForCluster Helix. UK thanks the Klaus Tschira Stiftung for their support via the SIMPLAIX project. None of the authors received any other additional revenues in relation to this work in the last 36 months.

References

Srikar Appalaraju, Yi Zhu, Yusheng Xie, and István Fehérvári. Towards good practices in self-supervised representation learning. *arXiv preprint arXiv:2012.00868*, 2020.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

Florian Bordes, Randall Balestriero, and Pascal Vincent. High fidelity visualization of what your self-supervised representation knows about. *arXiv preprint arXiv:2112.09164*, 2021.

Tom Burgert, Oliver Stoll, Paolo Rota, and Begüm Demir. Imagenet-trained cnns are not biased towards texture: Revisiting feature reliance through controlled suppression. *arXiv preprint arXiv:2509.20234*, 2025.

Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae. *arXiv preprint arXiv:1804.03599*, 2018.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR, 2020.

Elliot Creager, Jörn-Henrik Jacobsen, and Richard Zemel. Environment inference for invariant learning. In *International conference on machine learning*, pages 2189–2200. PMLR, 2021.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. *Advances in neural information processing systems*, 29, 2016a.

Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4829–4837, 2016b.

Felix Draxler, Peter Sorrenson, Lea Zimmermann, Armand Rousselot, and Ullrich Köthe. Free-form flows: Make any architecture a normalizing flow. In *International Conference on Artificial Intelligence and Statistics*, pages 2197–2205. PMLR, 2024.

Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1, 2009.

Linus Ericsson, Henry Gouk, and Timothy M Hospedales. How well do self-supervised models transfer? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5414–5423, 2021.

Elias Eulig, Fabian Jäger, Joscha Maier, Björn Ommer, and Marc Kachelrieß. Reconstructing and analyzing the invariances of low-dose ct image denoising networks. *Medical Physics*, 2024.

Shreyas Daniel Gaddam. Chexpert — multi-label classifier. `https://www.kaggle.com/code/shreydan/chexpert-multi-label-classifier`, 2024. Kaggle Notebook.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *International conference on learning representations*, 2018.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *stat*, 1050:20, 2015.

Daniella Har-Shalom and Yair Weiss. What do visual neural networks learn? *Annual Review of Vision Science*, 11, 2025.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15262–15271, 2021.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Christian Horvat and Jean-Pascal Pfister. Denoising normalizing flow. *Advances in neural information processing systems*, 34:9099–9111, 2021.

Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 590–597, 2019.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

Ivan Kobyzev, Simon JD Prince, and Marcus A Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE transactions on pattern analysis and machine intelligence*, 43 (11):3964–3979, 2020.

David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). In *International conference on machine learning*, pages 5815–5826. PMLR, 2021.

Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. *Advances in Neural Information Processing Systems*, 37:122458–122483, 2024.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022.

Francesco Locatello, Michael Tschannen, Stefan Bauer, Gunnar Rätsch, Bernhard Schölkopf, and Olivier Bachem. Disentangling factors of variation using few labels. In *8th International Conference on Learning Representations*, 2019.

R. J. Lyon, B. W. Stappers, S. Cooper, J. M. Brooke, and J. D. Knowles. Fifty years of pulsar candidate selection: from simple filters to a new principled real-time classification approach. *Monthly Notices of the Royal Astronomical Society*, 459(1):1104–1123, 2016. doi: 10.1093/mnras/stw656.

Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5188–5196, 2015.

Leon Mayer, Piotr Kalinowski, Caroline Ebersbach, Marcel Knopp, Tim Rädsch, Evangelia Christodoulou, Annika Reinke, Fiona R Kolbinger, and Lena Maier-Hein. 6 fingers, 1 kidney: Natural adversarial medical images reveal critical weaknesses of vision-language models. *arXiv preprint arXiv:2512.04238*, 2025.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research Journal*, 2024.

Yonatan Oren, Shiori Sagawa, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust language modeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4227–4237, 2019.

Kushagra Pandey, Farrin Marouf Sofian, Felix Draxler, Theofanis Karaletsos, and Stephan Mandt. Variational control for guidance in diffusion models. In *International Conference on Machine Learning*, pages 47755–47780. PMLR, 2025.

Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 78(5):947–1012, 2016.

Hamed Rahimian and Sanjay Mehrotra. Distributionally robust optimization: A review. *arXiv preprint arXiv:1908.05659*, 2019.

Pooyan Rahmanzadehgervi, Logan Bolton, Mohammad Reza Taesiri, and Anh Totti Nguyen. Vision language models are blind. In *Proceedings of the Asian Conference on Computer Vision*, pages 18–34, 2024.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

Tal Ridnik, Emanuel Ben-Baruch, Nadav Zamir, Asaf Noy, Itamar Friedman, Matan Protter, and Lihi Zelnik-Manor. Asymmetric loss for multi-label classification. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 82–91. IEEE, 2021.

Robin Rombach, Patrick Esser, and Björn Ommer. Making sense of cnns: Interpreting deep representations and their invariances with inns. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*, pages 647–664. Springer, 2020.

Elan Rosenfeld, Pradeep Ravikumar, and Andrej Risteski. The risks of invariant risk minimization. In *International Conference on Learning Representations*, volume 9, 2021.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296*, 2017.

Fawaz Sammani, Boris Joukovsky, and Nikos Deligiannis. Visualizing invariant features in vision models. In *2023 24th International Conference on Digital Signal Processing (DSP)*, pages 1–5. IEEE, 2023.

Peter Sorrenson, Felix Draxler, Armand Rousselot, Sander Hummerich, Lea Zimmermann, and Ullrich Köthe. Lifting architectural constraints of injective flows. *arXiv preprint arXiv:2306.01843*, 2023.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9446–9454, 2018.

Xin Wang, Hong Chen, Si'ao Tang, Zihao Wu, and Wenwu Zhu. Disentangled representation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9677–9696, 2024.

Wencheng Yang, Song Wang, Di Wu, Taotao Cai, Yanming Zhu, Shicheng Wei, Yiying Zhang, Xu Yang, Zhaohui Tang, and Yan Li. Deep learning model inversion attacks and defenses: a comprehensive survey. *Artificial Intelligence Review*, 58(8):242, 2025.

Yu Yang, Seungbae Kim, and Jungseock Joo. Explaining deep convolutional neural networks via latent visual-semantic filter attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8333–8343, 2022.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

Sheng Zhang, Yanbo Xu, Naoto Usuyama, Hanwen Xu, Jaspreet Bagga, Robert Tinn, Sam Preston, Rajesh Rao, Mu Wei, Naveen Valluri, et al. Biomedclip: a multimodal biomedical foundation model pretrained from fifteen million scientific image-text pairs. *arXiv preprint arXiv:2303.00915*, 2023.

Nanxuan Zhao, Zhirong Wu, Rynson WH Lau, and Stephen Lin. What makes instance discrimination good for transfer learning? *arXiv preprint arXiv:2006.06606*, 2020.

A Sample Refinement using Latent Gradient Descent

Depending on the desired level of fidelity, the invariance sampling process may not contract sufficiently to the fiber, achieving a fiber loss that is low enough. We can further refine the sample to reduce the fiber loss by applying a few (~ 100) gradient descent optimization steps directly on the sample, minimizing the fiber loss. Doing so directly in data space however, is likely to lead to pixel-level adjustments, which are not semantically meaningful, similar to adversarial attacks. We show this effect in Figure 12. For this reason, one needs to regularize the refinement process towards a natural image prior, for example Mahendran and Vedaldi [2015] introduce several regularizing terms to this end. We find that a good natural image prior is enforced by performing sample refinement in some semantic latent space, e.g. of an autoencoder, as shown in Algorithm 2. In Figure 12 we show the improvements in fiber loss and resulting samples gained for by refining samples in a VAE latent space using gradient descent to minimize the fiber loss. The latter leads to a much smoother result, fixing the color correlation and keeping the digit shape. This is owed to a more natural landscape for optimization. Crucially, gradient descent in data space leads to manipulating single pixel values to arrive at an invariance under the VAE of the subject model (i.e. the VAE applied to the decolorized images). While this may be interesting in some applications, in this setting we are interested in semantic variation, which is not encoded in these changes.

Algorithm 2 Latent Sample Refinement using an Autoencoder

Input: Encoder E, Decoder D, Fiber sample $\tilde{x}$, Target x, Subject model ϕ, Learning rate η, Number of gradient steps N
Output: Refined fiber sample $\hat{x}$
$\tilde{z}_0 \leftarrow E(\tilde{x})$
$h \leftarrow \phi(x)$
for $i = 0$ to $N - 1$ **do**
  $\tilde{x}_i \leftarrow D(\tilde{z}_i)$
  $\tilde{h}_i \leftarrow \phi(\tilde{x}_i)$
  $\mathcal{L}_i \leftarrow \| h - \tilde{h}_i \|_2^2$
  $\tilde{z}_{i+1} \leftarrow \tilde{z}_i - \eta \nabla_{\tilde{z}_i} \mathcal{L}_i$
end for
$\hat{x} \leftarrow D(\tilde{z}_N)$
return $\hat{x}$

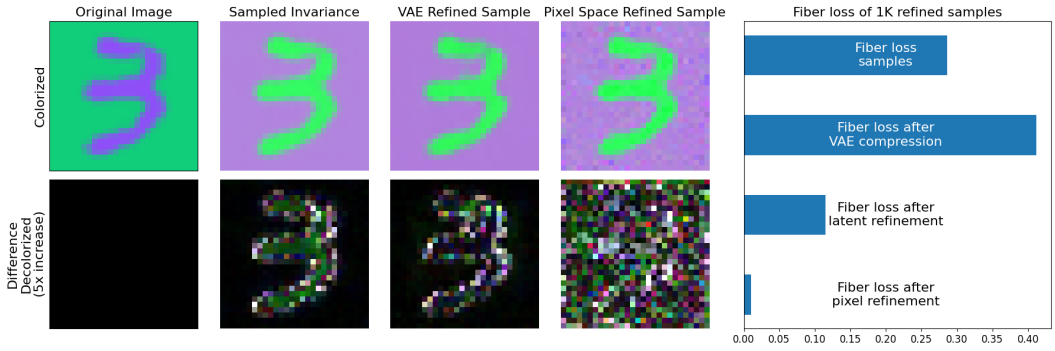


Figure 12: Sample refinement on a colorMNIST example. From left to right: Original image, invariant sample, sample refined with gradient descent in a VAE latent space and sample refined with gradient descent in pixel space. Below each image we show the difference of the decolorized image to the decolorized original. Note that even for non-zero difference the fiber loss may be small if they fall exactly in the invariances of the almost lossless autoencoder subject model, which is achieved by pixel level gradient descent. On the right we show the fiber loss improvement when applying sample refinement to fiber samples of a conditional normalizing flow fiber model on color MNIST.

B Experimental Details

Code to reproduce all of our experiments is available at `https://github.com/vislearn/InvarianceAuditing`.

B.1 Colored MNIST

B.1.1 Dataset

The dataset is based on MNIST images, which we color according to a fixed relation between foreground and background color. For each image, we obtain a random background color $c_0 \in \mathbb{R}^3$ and the corresponding foreground color $c_1 \in \mathbb{R}^3$. Each pixel is then colored according to its brightness. For this we sample for every color channel a value c^i i.i.d. from a Gaussian mixture distribution and compute the colored image by

$$\mathbf{x}_{\text{colored},i} = (1 - \mathbf{x}) \cdot c^i + \mathbf{x} \cdot ((c^i + 0.5) \mod 1). \tag{10}$$

That is, the background of the image gets colored in $c_0 = (c^1, c^2, c^3)$ and the digit in $(c_0 + 0.5) \mod 1$. This ensures that the digit always has a certain contrast to the background. The Gaussian mixture distribution we use to sample each c^i is given by

$$p(c^i) = 0.6 \cdot \mathcal{N}(c^i, 0.7, 0.08^2) + 0.35 \cdot \mathcal{N}(c^i, 0.5, 0.015^2) + 0.05 \cdot \mathcal{N}(c^i, 0.1, 0.02^2) \tag{11}$$

and shown in Figure 13 on the left. The resulting digit color distribution is shown on the right. The decolorize function is now the analytical inverse of equation 10 which is

$$\mathbf{x}_i = \frac{\mathbf{x}_{\text{colored},i} - c^i}{((c^i + 0.5) \mod 1) - c^i}. \tag{12}$$

We see now that if the digit color is not exactly $(c^i + 0.5) \mod 1$, the intensity of corresponding channel will be different than the original one. In order to examine the more common case where the features $\phi(x) = h$ are in vector shape, we encode the decolorized images by an effectively lossless autoencoder.

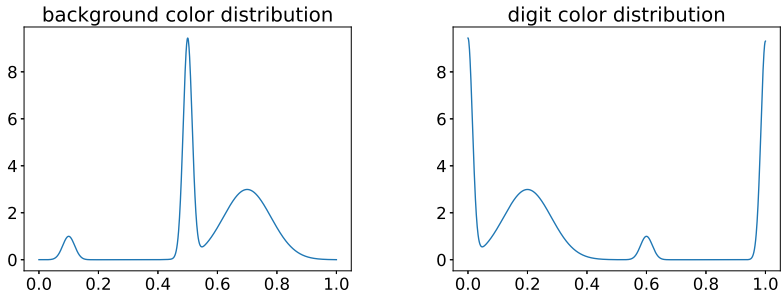


Figure 13: Gaussian mixture distributions for sampling the color values for each channel to colorize MNIST.

B.1.2 Model training

We train three different model classes as conditional fiber models: Normalizing flows (NF) [Kobyzev et al., 2020], free-form flows (FFF) [Draxler et al., 2024] and diffusion (DIFF) / flow-matching models (FM) [Ho et al., 2020, Lipman et al., 2023]. To answer the question whether injective models are more suited to learn the low-dimensional fiber manifold, we train injective equivalents where available. We choose denoising normalizing flows (DNF) [Horvat and Pfister, 2021] and multilevel flows (MLF) [Kingma and Dhariwal, 2018] for models based on invertible neural networks and free-form injective flows (FIF) [Sorrenson et al., 2023] as the injective equivalent of free-form flows. To stay accurate to the setting of Rombach et al. [2020] all conditional fiber models and the unconditional generative model for NDTM on color MNIST are latent generative models, learning the distribution of a VAE latent space. For clarity, we stress that this is not the lossless VAE serving as part of the subject model.

blocks per residual net	3
layers per block	3
layer size	512
latent dimension	54
activation function	SiLU
epochs	700
lr scheduler	one-cycle
max lr	0.0002
reconstr. loss weight	2000

Table 3: Hyperparameters of the fiber model VAE used for the color MNIST experiments.

The VAE is made up of 7.9 million parameters, distributed on encoder and decoder, each containing a convolutional part and a fully connected residual net. We list the hyperparameters in Table 3.

We trained all fiber models for up to 1000 epochs and checked for convergence in epochs 100 and 250, in which case we would apply early stopping. In general, we chose the architectures such that each fiber model contains around 10-15M parameters in total (including the VAE). Despite a fiber dimension of $\dim \phi^{-1}(h) = 3$, for all injective models except free-form injective flows we found a latent dimension of six to yield the best results. We use the onecycle learning rate scheduler and tune the learning rate from a range of $[2 \times 10^{-4}, 2 \times 10^{-3}]$.

B.1.3 Additional Results

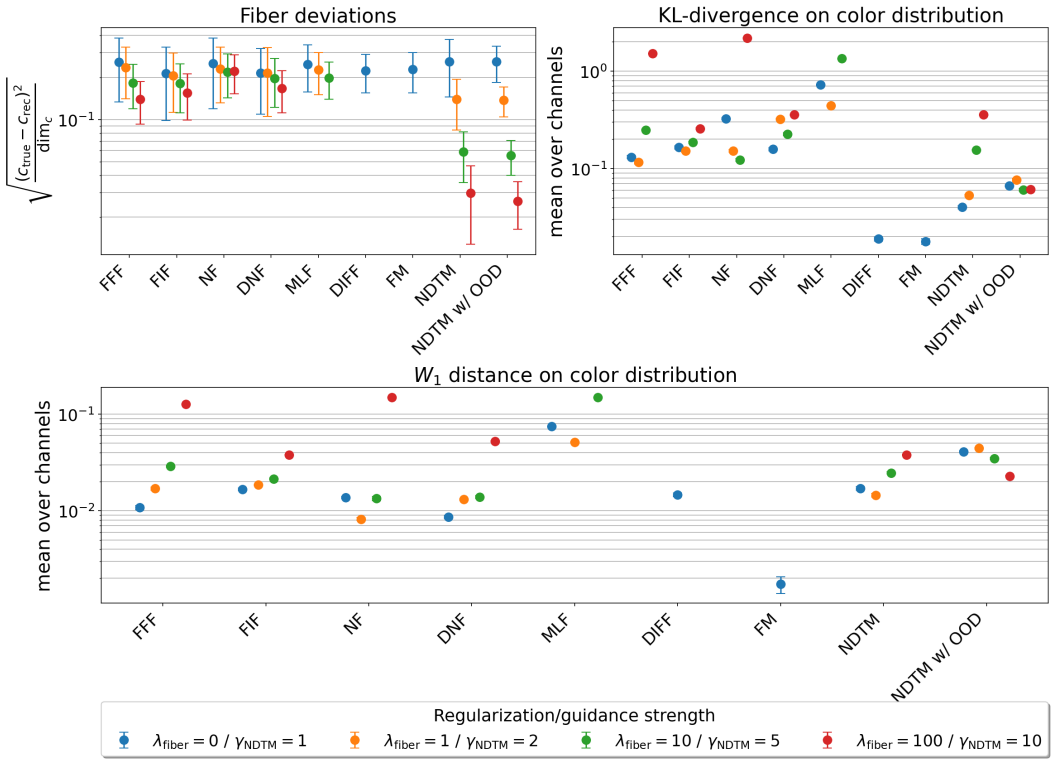


Figure 14: Comparative plot of fiber loss (top left), KL-divergence (top right) and W_1-distance (bottom) on the color distribution of the color MNIST benchmark. λ gives the weight of the fiber loss during training of the conditional models, while γ shows the guidance strength of NDTM. Standard deviations are taken across all sample sets and target embeddings (Note that for KL-divergence and W_1 distance the standard deviation is imperceptibly small for almost all models).

In Figure 14 we provide comparisons of fiber losses, KL-divergences and Wasserstein distances on the color distribution for all models. These results show that there generally exists a trade-off between

models that achieve good consistency and high fidelity. We also provide standard deviations for all metrics, obtained by creating ten (three for NDTM) independent samples for each fiber. Note that for the fiber loss, we also compute the standard deviation across different target embeddings, which is why it shows higher uncertainty. This encapsulates the difference in how well the methods can recover high fidelity samples across all possible target embeddings.

B.2 ImageNet & Cue Conflict

In Table 4 we report estimates for the time required to train and sample with a conditional model like in Bordes et al. [2021] vs. sampling time of NDTM on a NVIDIA A40 GPU. We find that it only becomes faster to train a model when creating more than 9k samples, disregarding the time for hyperparameter optimization.

Table 4: Estimates of the time required to train and sample from a conditional fiber model vs. sampling via NDTM guidance on ImageNet. Numbers are based on sampling with 100 timesteps and $N = 5$ optimization steps for NDTM on a NVIDIA A40 GPU.

Training conditional model	Time per sample conditional model	Time per sample NDTM
$\sim$ 14000 min	0.3 min	1.8 min

In Table 5 we list the fiber losses of experiments on ImageNet and the cue conflict dataset. This confirms that NDTM manages to contract closely to fibers in relation to e.g. nearest neighbors in a dataset, even when the target representations are created by OOD inputs.

Table 5: Fiber losses of all subject models on large image datasets. As a comparison we provide the fiber losses of the nearest neighbors from the dataset as well as the original image with typical DINOv2 training augmentations. Standard deviations are measured across 10 independently drawn samples per fiber.

Subject Model	Dataset	$\mathcal{L}_{\text{fiber}}$ invariant samples	$\mathcal{L}_{\text{fiber}}$ nearest neighbor	$\mathcal{L}_{\text{fiber}}$ augmented
DINOv2	ImageNet	$\mathbf{273 \pm 4}$	1225	440
Inception	ImageNet	$\mathbf{34.8 \pm 0.7}$	123.9	94.9
DINOv2	Cue Conflict	$\mathbf{487 \pm 3}$	1069	946
ResNet	Cue Conflict	$\mathbf{38.1 \pm 0.7}$	74.9	101.3
BiomedClip	CheXpert	$\mathbf{1.6 \pm 0.0}$	23.3	-
ConvNext	CheXpert	$\mathbf{1.2 \pm 0.0}$	10.9	-
Qwen	CheXpert	$\mathbf{5.5 \pm 3.0}$	17.8	-

For evaluating ImageNet FID, fiber losses and classifier accuracy on the fiber in tables 2 & 5, we draw a set of 10k fiber samples.

B.3 Causal MNIST

In this section we illustrate how invariance auditing can be combined with methods that control which invariances a model learns. The resulting workflow can be used to develop models whose decisions are more robust and trustworthy.

B.3.1 Related work on controlling invariances

Several works develop methods that encourage or discourage models to rely on certain invariances during training. Invariant learning methods [Peters et al., 2016, Arjovsky et al., 2019, Rosenfeld et al., 2021, Creager et al., 2021, Krueger et al., 2021] aim to learn representations whose optimal predictors remain stable across multiple environments, thereby discouraging reliance on spurious correlations. Related approaches such as group distributionally robust optimization [Rahimian and Mehrotra, 2019, Oren et al., 2019, Sagawa et al., 2019] promote invariances useful for generalization by penalizing worst-case risks. In the context of representation learning, methods based on data augmentations [Chen et al., 2020, Oquab et al., 2024] implicitly enforce invariances to predefined transformations, while approaches like information bottleneck methods [Tishby et al., 2000] discard nuisance factors, keeping only task relevant features. Conversely, several works study how to control and quantify

sensitivity to specific attributes, for example through disentanglement objectives [Burgess et al., 2018, Locatello et al., 2019, Wang et al., 2024].

B.3.2 Experimental Details

This section provides additional detail on the experiment performed in Section 4.4. In this experiment we want to show how a combination of invariance auditing and invariance control can first reveal and then eliminate unwanted invariances. To this end we compare a model that is trained with classical empirical risk minimization (ERM) to one that is trained with invariant risk minimization (IRM) [Arjovsky et al., 2019] on a version of color MNIST from the causal invariance literature. Note that this dataset is distinct from the one in Section 4.1, which is based on the color MNIST dataset introduced by Rombach et al. [2020], to distinguish them we will refer to the dataset from Arjovsky et al. [2019] as "causal MNIST". The binary classification task labels in causal MNIST are given by whether a digit is < 5 or ≥ 5. Noise is introduced to obtain the final labels by flipping them with a probability of 25%. Afterwards, the images are colored red or green with different probabilities, depending on three environments: In the two environments used for training the color ID coincides with the final label 80% and 90% of the time, while in the testing environment this correlation is flipped and it coincides with the final label only 10% of the time. This way color serves as a stronger predictor than the digit itself, but the correlation is spurious and relying on color for prediction does not generalize to the test environment. In Arjovsky et al. [2019] it is shown that ERM relies on this spurious correlation and is not robust under domain shift, while IRM does not. All subject models are trained according to the publicly available implementation of Arjovsky et al. [2019][1].

Our aim is to exemplify a general workflow for trustworthy model design: We first show that invariance auditing reveals that ERM relies on the known spurious correlation. Adopting a method for invariance control like IRM then removes the undesired reliance on this predictor, making the model invariant to color. Invariance auditing then verifies that invariance control was successful.

To perform invariance auditing we train a diffusion model on the environment with 80% color – label correlation for NDTM. This determines the color correlation of the underlying distribution $p(x)$ for invariance auditing. As target representations for invariance auditing we sample classifier logits $h \sim p(h)$ using input images from the environment with 10% color – label correlation. If no color information is present in h, i.e it is invariant to color, then the color – label correlation in $p(x|h)$ should be the same as in $p(x)$. Otherwise we expect to see a bias towards the color – label correlation of the inputs used to create $p(h)$ depending on how strongly the representation encodes color.

This is what we observe in Figure 15. The robust IRM model exhibits no shift in color – label correlation, while the ERM model does. This shows that when reliance on a spurious correlation is suspected, invariance auditing can be used to expose it. Pairing this diagnostic method with invariant learning methods like IRM can serve as a tool to first identify and then remove unwanted reliance on spurious features.

B.4 Chexpert

Our implementation of the subject model training is based on the Kaggle notebook available at Gaddam [2024]. As our subject models we pick a ConvNeXt-tiny (end-to-end trained classifier) Liu et al. [2022] and a frozen BiomedClip feature extractor [Zhang et al., 2023] with a 5-layer ResNet classification head. We train both classifiers for 36k steps on the CheXpert dataset [Irvin et al., 2019], with an asymmetric loss function [Ridnik et al., 2021]. As no test set is provided, we reserve a part of the original training set for validation, and use the original validation set for testing. ROC curves on test data are given in Figure 17, upper panel. We calibrate the probabilities of each classifier via temperature scaling and set the classification threshold for each classifier, s.t. it is optimal under F1-score. The unconditional diffusion model architecture is a transformer UNet with 4 blocks and 2 layers each, totaling 18M parameters, which we trained for 120k steps. All models take a resolution of 384x384.

In Figure 16 we show random invariant samples of both subject models, compared to their nearest neighbors on the dataset. While the fiber loss for NDTM is the difference in logits, for a more

[1] `https://github.com/facebookresearch/invariantriskminimization`

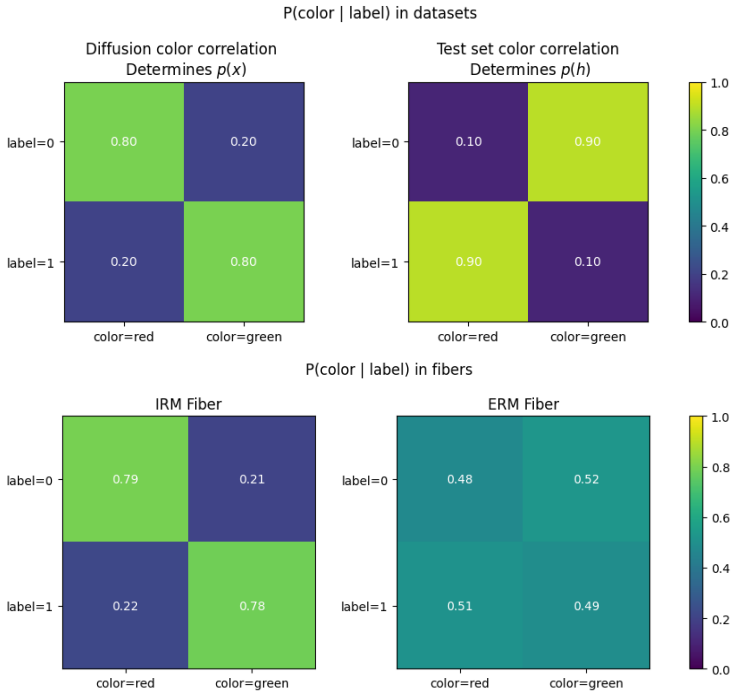


Figure 15: Top: Color – label correlation in the diffusion model used for NDTM and the test data used to create the target representations for invariance auditing. Bottom: Color – label correlation in the fiber samples of a model that does not encode color information in h (IRM) and one that does (ERM).

interpretable distance measure we show fiber loss in terms of summed absolute difference in class probability $\sum_{c=1}^{C} |p_c(x) - p_c(\tilde{x})|$.

We can see that NDTM achieves excellent fiber losses, with the assigned probabilities differing by less than 1% in sum across all classes. In Figure 17 we highlight the difference in agreement between dataset originals and fiber samples for both models by showing ROC curves on various combinations of images and labels. When relabeling original images with either the ConvNeXt or BiomedClip model, we find their agreement rate on the original data (middle panel). We then label each fiber with the other subject model, then compute the ROC curves using the subject model with respect to which the fiber samples were originally drawn. This shows that their agreement mainly holds on original data, but not their fibers.

Now we repeat the experiment for two ConvNeXt classifiers, which were trained identically. We show the corresponding ROC curves and classifier agreement measured analogous to Section 4.5 in Figure 18. This shows that identically trained models with the same architecture tend to also share the same fibers.

In order to verify that the invariances we find for the Qwen-2B vision encoder in rare anatomical cases are valid we can compare the fiber losses of Figure 11 with fiber losses on typical anatomy we obtained from a small sample of CheXpert representations. In the latter case the fiber loss is around 5.5 ± 3.0. The fiber losses of situs inversus images are comparable, meaning the fiber samples achieve the same level of invariance without reflecting the rare anatomy.

B.5 Tabular Experiment: HTRU2

To illustrate that invariance auditing can also be used for non-image modalities we audit the invariances of a tabular classifier on the HTRU2 dataset [Lyon et al., 2016] for pulsar classification. The subject model classifier is a small MLP (2 layers, 128 hidden dimension) trained to detect pulsars from measured radio emissions. To perform fiber sampling we train an unconditional ResNet diffusion

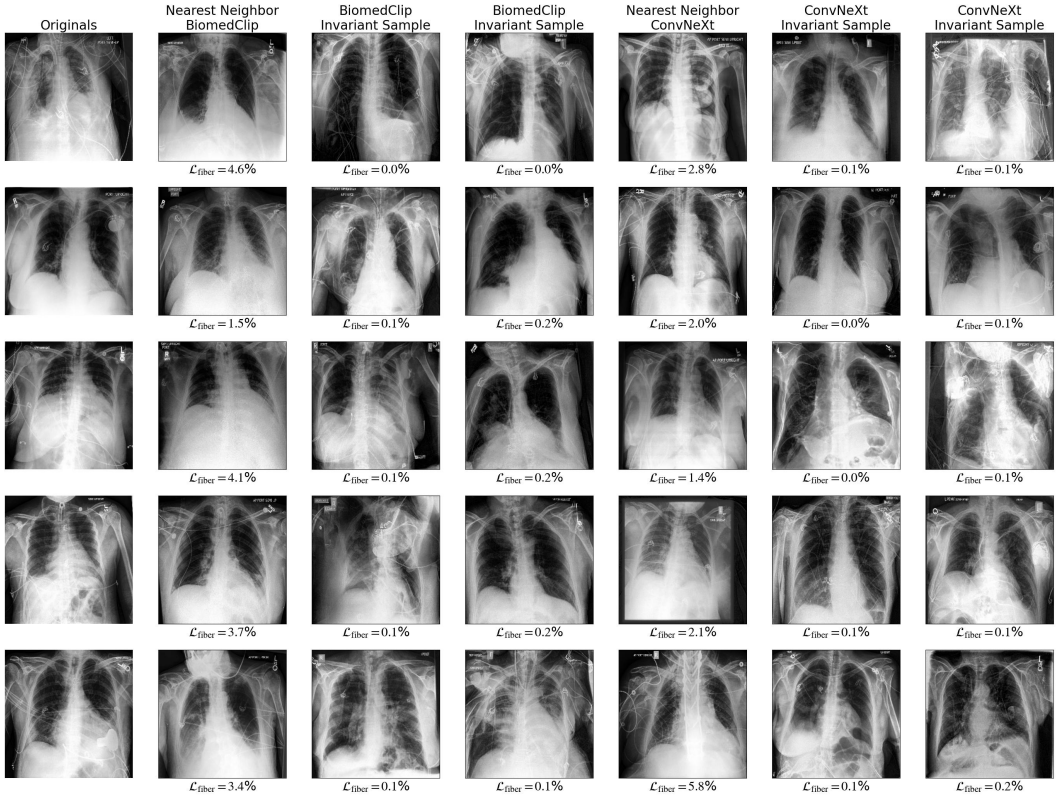


Figure 16: From left to right: Original image, nearest neighbor and two random invariant samples of BiomedClip, nearest neighbor and two random invariant samples from ConvNeXt. Below each image we give the summed absolute difference in predicted probabilities across all classes.

model (8 layers, 256 hidden dimension) and use the difference in classifier logits as fiber loss for NDTM.

First, we measure fidelity and consistency in Figure 19 left, to verify that fiber sampling was successful. Then we use the fiber samples to determine the importance of each parameter in the following way: We normalize each input feature j from the tabular dataset to obtain values x_j, then measure the standard deviation of the difference between the fiber sample value to the original value $\sigma(\Delta_j)$. The parameter importance is then the inverse of this:

$$\sigma(\Delta_j)^{-1} = \left(\sqrt{\mathbb{E}[|x_{j,\text{orig}} - x_{j,\text{fiber}}|^2] - \mathbb{E}[|x_{j,\text{orig}} - x_{j,\text{fiber}}|]^2} \right)^{-1} \tag{13}$$

The intuition behind this is that features that can vary a lot have low relevance for the classifier prediction, while parameters with small variation are important. In Figure 19 right we compare the importance values obtained this way for each parameter to standard classifier gradient importance $\left\| \nabla_{x_j} \phi(x) \right\|$. As a baseline we order features j from most to least important based on the mutual information they have with the classification label (see Table 7 in Lyon et al. [2016]).

For all methods, profile kurtosis is the strongest predictor of pulsars, which is in line with it having the highest mutual information with the classification label. On the other features there is considerable disagreement. Gradient importance, which only considers variation in the local neighborhood, suggests that the classifier does not give higher importance to features with higher mutual information. Without profile kurtosis Spearman's rank correlation is $\rho = -0.04$ between the importance ranking of the gradient and mutual information. Invariance auditing, which considers the global variation of features, arrives at a different conclusion: The classifier's feature importance measured by fiber sample variance is strongly correlated with the importance ranking through mutual information, $\rho = 0.61$. It should be noted that, since the features in HTRU2 are themselves correlated, marginal feature importance measures have limited significance. Yet, this experiment highlights the ability of

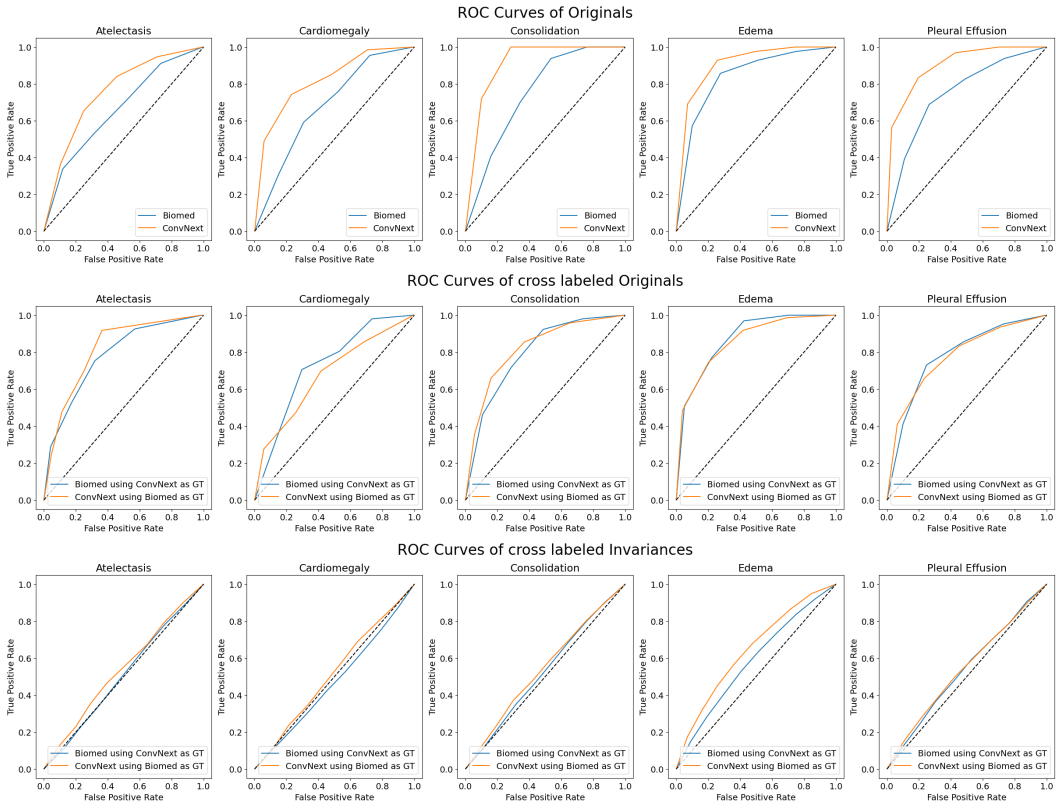


Figure 17: ROC curves for each of the five classes, that the CheXpert classifiers were trained on. We show results for comparing the BiomedClip classifier to the ConvNeXt classifier. Top: ROC curves for originals, using the ground truth labels. Center: ROC curves for originals, using the opposite model to create target labels. Bottom: ROC curves for invariances sampled for each model, with target labels created by the opposite model.

invariance auditing to perform global exploration of the fibers. Relying only on the local importance measures we may wrongly conclude that the classifier ignores high-information features, while invariance auditing reveals the correlation with mutual information on a global scale.

C Details on NDTM

In this section we provide additional details on NDTM and our modification of it. NDTM guides the generation process of diffusion/flow-matching models, by adding a learnable correction u_t between each denoising step

$$\hat{x}_t = x_t + \gamma_t u_t \tag{14}$$

$$x_{t-1} = \text{DenoisingStep}(\hat{x}_t). \tag{15}$$

Here γ_t is the guidance strength and u_t is a parameter of same dimensionality as x_t, which is optimized to minimize the following loss function:

$$x'_t \sim q(x'_t|x_0 = x) \tag{16}$$

$$\mathcal{L}_{\text{terminal}}(x_t, x'_t) = ||\phi(\mathbb{E}[x_0|x_t]) - \phi(\mathbb{E}[x_0|x'_t])||_2^2 . \tag{17}$$

$$\mathcal{L}_{\text{control}} = ||u_t||_2^2 \tag{18}$$

$$\mathcal{L}_{\text{score}} = ||\nabla_{x_t+\gamma_t u_t} \log p(x_t + \gamma_t u_t) - \nabla_{x_t} \log p(x_t)||_2^2 \tag{19}$$

$$\mathcal{L} = \mathcal{L}_{\text{terminal}} + \kappa_t \mathcal{L}_{\text{control}} + \tau_t \mathcal{L}_{\text{score}} \tag{20}$$

$\mathcal{L}_{\text{terminal}}$ minimizes the terminal cost, in our case the fiber loss, $\mathcal{L}_{\text{control}}$ regularizes the size of the correction u_t and $\mathcal{L}_{\text{score}}$ regularizes the change in score after the correction step, which is approximated

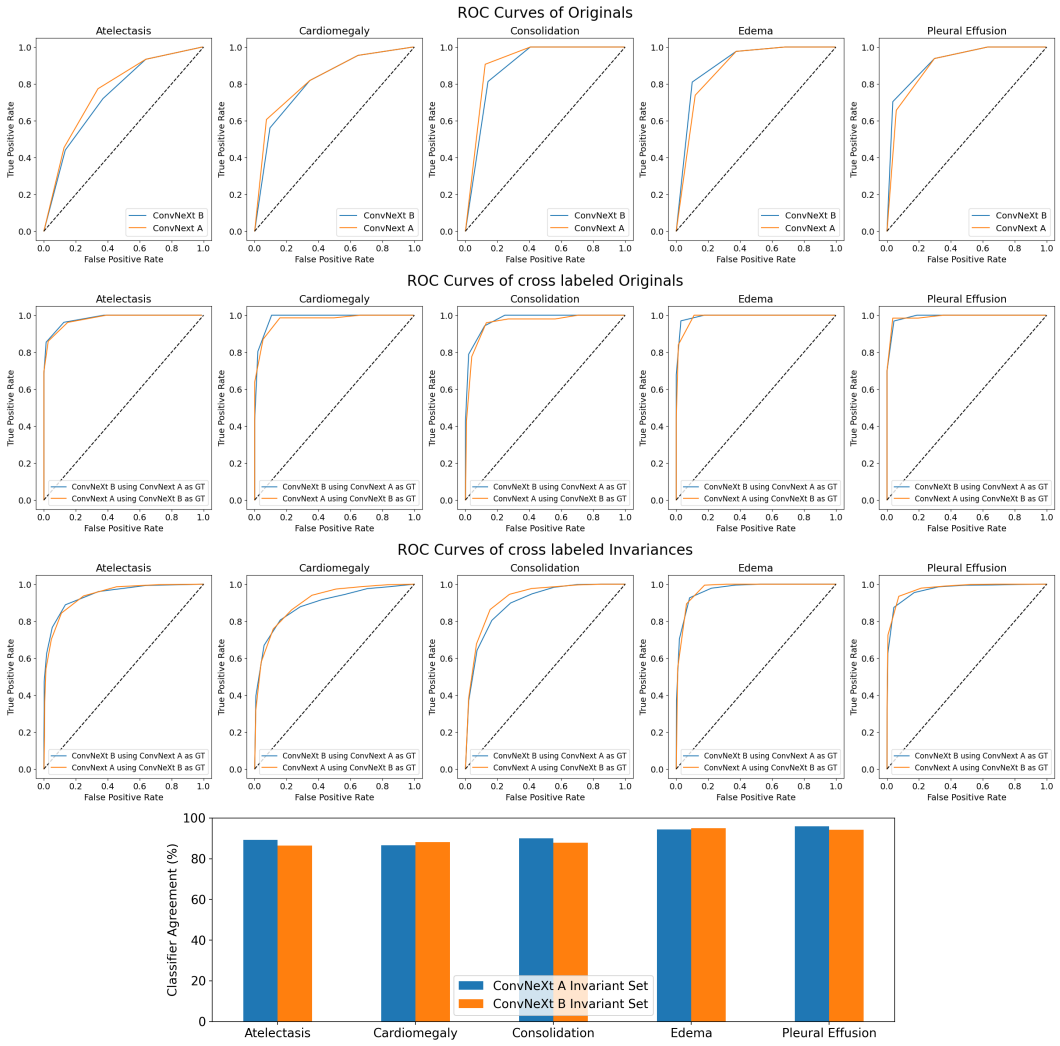


Figure 18: ROC curves for each of the five classes, that the CheXpert classifiers were trained on. We show results for two identically trained ConvNeXt classifiers. Top: ROC curves for originals, using the ground truth labels. Center: ROC curves for originals, using the opposite model to create target labels. Bottom: ROC curves for invariances sampled for each model, with target labels created by the opposite model. Below the ROC curves we show the classifier agreement on invariant sets sampled for originals, where both models predict the correct label.

by the diffusion network score estimate. We find the hyperparameter choices $\kappa_t = 10^{-4}$ and $\tau_t = 0$ to be consistently stable choices for most settings. For each correction step, u_t is initialized to zero and optimized by N (usually 4 - 8) steps of gradient descent to minimize $\mathcal{L}$. The learning rate linearly decreases from 2×10^{-3} at the start of the denoising process to 0 at the end. Similar to classifier free guidance, we find it helpful to adjust the guidance strength γ_t based on t. Specifically, we find that, depending on the diffusion model, high guidance strength is most helpful at a mid-range interval (starting around $t \in [800, 500]$), similar to prior works [Kynkäänniemi et al., 2024]. Increasing γ_t towards the end of the generation process can also help to reduce the fiber loss further, which we attribute to a better match in relevant fine textures.

C.1 Proposed Modification

In Section 3.2 we discussed our proposed modification to the terminal cost, that replaced the features of the input image $\phi(x)$ with that of a noised and re-approximated version $\phi(\mathbb{E}[x_0|x'_t])$. In Figure 20 we show invariant samples of ImageNet taken with this new formulation compared to a static target.

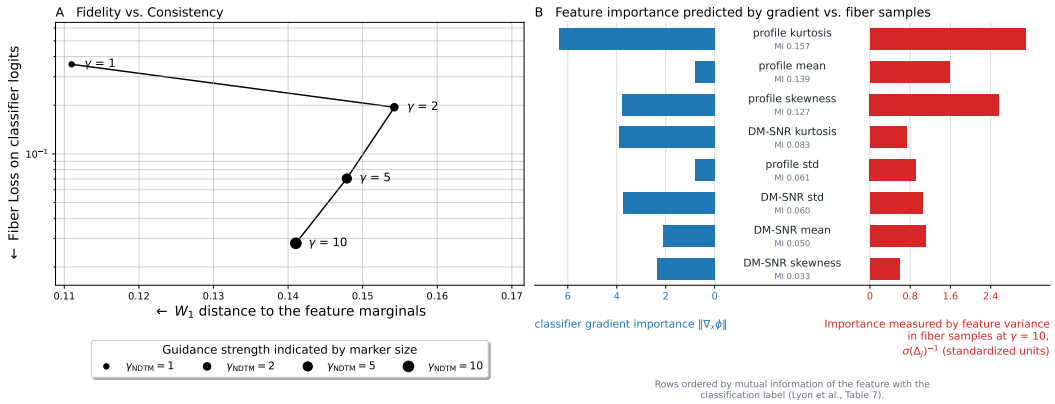


Figure 19: Invariance auditing of a tabular classifier on the HTRU2 dataset. Left: fidelity vs. consistency for fiber sampling with NDTM. The fiber losses correspond to differences in predicted probabilities of 6.1% at $\gamma = 1$ to 0.3% at $\gamma = 10$. Right: Feature importance as measured by gradients (blue) and feature variance in fiber samples (red). The features are ordered from the ones that have the most to least mutual information (MI) with the classification label.

Table 6: Numerical results of performing invariance auditing for DINOv2 on ImageNet with unmodified and modified NDTM. Standard deviations are measured across the entire dataset.

Metric	Unmodified NDTM	Modified NDTM
FID	28.87	28.34
Fiber loss	242 ± 328	272 ± 142

We can see that NDTM with re-approximated target features focuses a lot more on semantic content, while keeping colors and textures of the image similar to the underlying distribution $p(x)$. In contrast, static target feature guidance tries to create the same features that the detailed texture and colors of x yield with the coarse approximation $\mathbb{E}[x_0|x_t]$, which results in out-of-distribution colors and textures in the final sample. In Table 6 we compare the resulting fiber loss and FID of the two sample sets. Our modification results in slightly higher fiber losses but increased stability (as indicated by the lower standard deviations of fiber losses) and slightly better consistency as measured by FID, which confirms its benefits for stability.

Empirically, we find that NDTM exhibits higher fiber losses in OOD settings (see Table 5) and on images-subject model combinations where representations depend on fine details that emerge late in the sampling process. In both cases we observe sampling instabilities for unmodified NDTM, which we suspect are due to a mismatch in granularity of the features present in the clean input image and the current noisy sample approximation. We find that our modification improves stability in these cases, but fiber loss is still elevated.

As we use changing inputs to the subject model for modified NDTM, a possible concern is that additional information from x may leak into the optimization process. To investigate this we include an additional experiment on color MNIST in Figure 21, where we fix the background color (and by implication the digit color) to a constant value for 5k different images. Then we draw invariant samples using NDTM with and without our modification. If information could leak in the latter case, we should see a bias towards the fixed color, however, the sampled distribution $p_\theta(x|h)$ remains unchanged. Additionally, we can compare the diversity of our fiber samples for DINOv2 on ImageNet to those obtained by Bordes et al. [2021]. On a visual comparison, both methods manage to roughly reconstruct the same amount of information from the embeddings, verifying that our modification does not lead to a collapse onto the input image x.



Figure 20: Random DINOv2 invariant samples on ImageNet without (middle panel), then with (lower panel) our modification to NDTM.

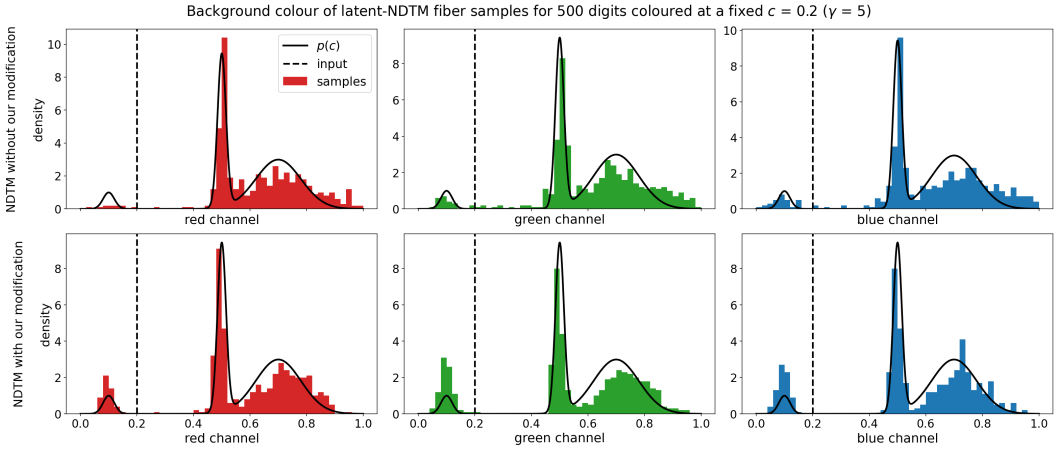


Figure 21: Color distribution resulting from coloring 5k MNIST digits at a fixed color (dashed line) and creating fiber samples with NDTM. The upper panel shows sampling without our modification, the lower panel with our modification. The solid black line shows the ground truth color distribution.

C.2 Choice of Fiber Loss Metric

In Section 3 we introduced the fiber loss as the difference between target and fiber sample representations measured by any metric

$$\mathcal{L}_{\text{fiber}} = \text{dist}(\phi(x), \phi(\tilde{x})). \tag{21}$$

The fiber is fully determined by the subject model, and regardless of which metric we choose, if the fiber loss is zero, the sample will be exactly on the fiber. In practice, due to numerical limitations in the optimization process we only achieve fiber losses close to zero and the choice of fiber loss metric determines which shape of numerical deviations are preferred/tolerated. For this paper we chose to use ℓ_2 distance as fiber loss metric, as it is a quite general choice in the absence of further information about the geometry of the subject model representations. As shown in Table 7, using ℓ_2 fiber loss as terminal loss for NDTM also yields samples that have low distance as measured in other metrics – as good as using these other metrics for NDTM themselves. We conclude that ℓ_2 distance is a robust, geometry-agnostic choice. A more informed choice could be made on a case-by-case basis by taking into account the geometry of the subject model representations. Specifically, differences in representations could be measured by relevance to the task-outcome, e.g. using differences in

Table 7: Comparison of different fiber loss metrics for invariance auditing of DINOv2 on ImageNet with NDTM. We use different metrics as terminal loss for NDTM (top to bottom) and report the fiber loss as measured by each metric (left to right). At the bottom we show the nearest neighbor distances for each metric in the dataset. Standard deviations are measured across the entire dataset.

Terminal Loss Metric ↓ / Evaluation Metric →	ℓ_2	ℓ_1	Cross-Entropy
ℓ_2	272 ± 176	349 ± 103	5.34 ± 0.32
ℓ_1	277 ± 215	335 ± 110	5.36 ± 0.37
Cross-Entropy	660 ± 461	533 ± 171	5.37 ± 0.29
Nearest Neighbor	1224 ± 589	748 ± 186	5.17 ± 0.29

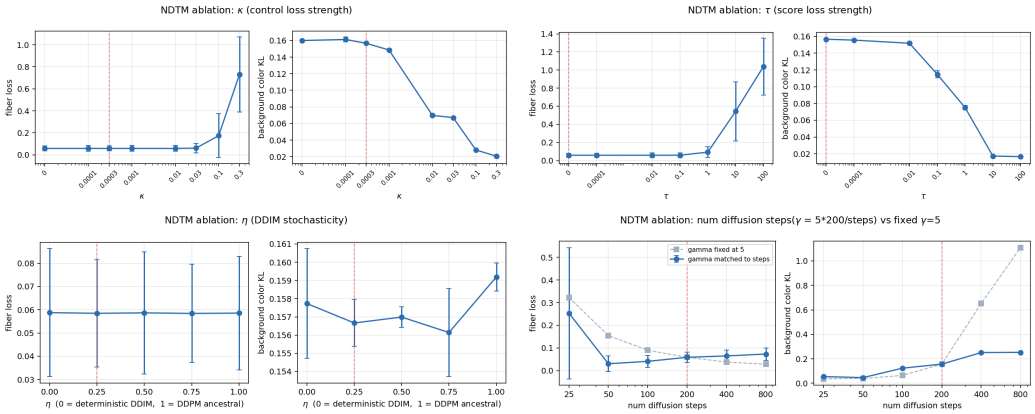


Figure 22: Ablations of different NDTM hyperparameters on color MNIST. Unless subject to ablation, all other parameters are kept at the values $\gamma = 5$, $\kappa = 3 \cdot 10^{-4}$, $\tau = 0$, $\eta = 0.25$, 200 diffusion steps, which were used for the experiments in Section 4.1 (marked by the red line).

classification probabilities for classifier subject models as was showcased for evaluation in Section B.4. Since our aim in this paper is to be subject model agnostic wherever possible, we leave an exploration of different fiber loss metrics for future work.

C.3 Ablation of NDTM hyperparameters

We perform additional ablations on NDTM hyperparameters on the color MNIST benchmark to determine the robustness of our results against hyperparameter choices. In particular, we show ablation results over the control and score regularizer strengths κ and τ, the sampling scheme used parameterized by η (interpolating between DDIM at $\eta = 0$ and DDPM at $\eta = 1$), and finally the number of sampling steps. For the number of sampling steps in particular there are two variants: NDTM guidance is applied in each sampling step, so doubling the sampling steps effectively doubles the guidance. We therefore present two versions of this ablation, one where the guidance strength γ is adjusted to compensate for this effect and one where it is kept constant.

All results are shown in Figure 22. We can see that NDTM for fiber sampling is robust against most hyperparameters: Sampler stochasticity η has little to no effect, the regularizers both exhibit a long, stable plateau of no effect, then provide the expected trade-off between fidelity and consistency. We note Figure 22 suggests that for both regularizers there is a small interval where consistency can be significantly improved without significantly affecting fidelity until the trade-off begins. Finally, the number of sampling steps shows the expected fidelity-consistency trade-off through larger effective guidance when not compensating through γ. With the compensation, we find that a low number of sampling steps can lead to even better results in both than the 200 we used for Figure 4.

Table 8: Licenses of used assets

Asset	**Type**	**License**
MNIST (LeCun et al., 2010)	Dataset	CC BY-SA 3.0
ImageNet	Dataset	Custom (non-commercial / restricted)
CheXpert (Irvin et al., 2019)	Dataset	Custom Stanford license (non-commercial)
Cue Conflict Dataset (Geirhos et al., 2018)	Dataset	CC BY 4.0
Color MNIST (Rombach et al., 2020)	Dataset	CC BY-SA 3.0 (derived)
DINOv2 ViT-B/14 (Oquab et al., 2023)	Model	Apache 2.0
InceptionV3 (Szegedy et al., 2016)	Model	Apache 2.0
ResNet-50 (ImageNet pretrained)	Model	BSD 3-Clause
ConvNeXt (Liu et al., 2022)	Model	MIT
BiomedCLIP (Zhang et al., 2023)	Model	MIT
Qwen-2B vision encoder	Model	Custom Apache 2.0
ImageNet diffusion model (Dhariwal and Nichol, 2021)	Model	MIT
Kaggle notebook (Gaddam, 2024)	Code	Apache 2.0

Table 9: Total estimated GPU compute time of each experiment, including training of fiber and subject models where no pretrained models are used. The main source of compute cost was sampling 10k fiber samples to compute FID for ImageNet models. All experiments were performed on a compute cluster with NVIDIA A100 40GB GPUs.

Experiment	Total GPU time
ColorMNIST	~ 100h
ImageNet	~ 800h
Cue Conflict	~ 400h
CheXpert	~ 140h
Causal MNIST	< 5h
HTRU2	< 5h

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: For each of our main claims we provide a matching experiment, specifically the performance and scalability of guided invariance auditing is shown on colorMNIST and ImageNet.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss limitations in Section 5, including the training-guiding compute trade-off and the need for methods to remove blind spots once they have been identified.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We provided details on all experimental setups in Section B and the pretrained models used are all publicly available. We will release our code upon acceptance.

   Guidelines:

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

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

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

Answer: [No]

Justification: We will release our code upon acceptance.

Guidelines:

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

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: Experimental details are given in Section B. Each experiment section in the main contains the high-level experimental setting which is necessary to understand the results.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: We provide standard deviations for all numerical results in our tables. Qualitative results in Section 4.5 are explicitly labeled as such.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We provide approximate runtimes of all experiments in Table 9.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: None of the ethical concerns listed in the code of ethics apply in our case.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: With this work our goal is to advance the interpretability and trustworthiness of machine learning models. We do not anticipate any negative societal impacts from this.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: Our paper does not present any new models/data that has high risk of misuse.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All existing assets are properly credited via citation. Licenses are provided in Table 8.

    Guidelines:

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

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

13. **New assets**

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

    Answer: [Yes]

    Justification: We introduce a new benchmark in Section 4.1, which is documented in Section B.1.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: No human subjects were involved in this paper.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: No human subjects were involved in this paper.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs do not constitute any core, non-standard components of our method development.

    Guidelines:

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