Evaluating the Interpretability of Generative Models by Interactive Reconstruction

Andrew Slavin Ross
Harvard University
Cambridge, MA, USA
andrew_ross@g.harvard.edu

Nina Chen
Harvard University
Cambridge, MA, USA
nchen1@college.harvard.edu

Elisa Zhao Hang
Harvard University
Cambridge, MA, USA
ezhaohang@college.harvard.edu

Elena L. Glassman
Harvard University
Cambridge, MA, USA
glassman@seas.harvard.edu

Finale Doshi-Velez
Harvard University
Cambridge, MA, USA
finale@seas.harvard.edu

ABSTRACT

For machine learning models to be most useful in numerous sociotechnical systems, many have argued that they must be human-interpretable. However, despite increasing interest in interpretability, there remains no firm consensus on how to measure it. This is especially true in representation learning, where interpretability research has focused on "disentanglement" measures only applicable to synthetic datasets and not grounded in human factors. We introduce a task to quantify the human-interpretability of generative model representations, where users interactively modify representations to reconstruct target instances. On synthetic datasets, we find performance on this task much more reliably differentiates entangled and disentangled models than baseline approaches. On a real dataset, we find it differentiates between representation learning methods widely believed but never shown to produce more or less interpretable models. In both cases, we ran small-scale think-aloud studies and large-scale experiments on Amazon Mechanical Turk to confirm that our qualitative and quantitative results agreed.

1 INTRODUCTION

Many have speculated that machine learning (ML) could unlock significant new insights in science and medicine thanks to the increasing volume of digital data [28]. However, much of this data is unlabeled, making it difficult to apply many traditional ML techniques. Generative modeling, a subfield of ML that does not require labels, promises to help by distilling high dimensional input data into lower dimensional meaningful axes of variation, which we call

representations. To be most useful in "unlocking insights," though, these representations must be understood by human researchers.

Motivated by the need for human understanding, a burgeoning research area of interpretable ML has emerged [21, 25]. While some of this work has used user studies to quantify interpretability [33, 39, 42, 56], there have been concerns within the HCI community that these studies do not generalize to more realistic use cases [11]. These studies are also largely in the context of discriminative rather than generative modeling—even in the few that consider representations [4]. Within generative modeling, ML researchers have tried to quantify interpretability via measures of *disentanglement*, which measure how well individual representation dimensions match ground-truth factors that generated the data [57]. However, this work is not tested on actual human users, nor are disentanglement measures computable without knowing ground-truth factors.

In this work, we develop a method for quantifying the interpretability of generative models by measuring how well users can interactively manipulate representations to reconstruct target instances. To validate it, we use both MTurk and lab studies to determine whether models known to be understandable a priori can be distinguished from those known to be complex, and also whether our quantitative metrics match qualitative feedback from users. We also investigate the relationship between our human-grounded interpretability measures and synthetic disentanglement measures.

Our main contributions are as follows:

- A task for evaluating the interpretability of generative models, where users interactively manipulate representation dimensions to reconstruct target instances.
- Large-scale experiments on Amazon Mechanical Turk and smaller-scale think-aloud studies showing our task distinguishes entangled from disentangled models and that performance is meaningfully related to human model understanding, as demonstrated and reported by study participants.
- Novel results suggesting that ML methods which improve disentanglement on synthetic datasets also improve interpretability on real-world datasets.

2 RELATED WORK

2.1 Human-Centered Interpretability Measures

While there have been criticisms that "interpretability" is ill-defined [48], several works have focused on quantifying it, particularly for

discriminative models [1, 2, 21, 23, 33, 40, 42, 47, 56, 61, 63]. Per Doshi-Velez and Kim [21] and Miller [52], these works typically ground interpretability as the capacity of the model to be sufficiently understood in an appropriate context, and operationalize it as a user's ability to perform various tasks given visualizations of a model according to some performance measures—where the tasks and measures are presumed to be relevant to the desired context. They then study the effect of varying visualizations or models on task performance measures.

As a concrete example, Kulesza et al. [41] present different visualizations of a random forest [10] model, and as their task, ask users a series of questions in a thinkaloud-style protocol [46]. Their performance measure is the difference between the number of accurate and inaccurate statements about the model made by the user, which they compute by transcribing interviews and individually categorizing each statement. As their theoretical grounding, they draw on notions of understanding from Norman [53], who defines understanding in terms of user mental model accuracy. They find that visualizations produce more accurate mental models when complete (the whole truth) and sound (nothing but the truth), even if satisfying those conditions dramatically increases complexity.

More commonly, interpretability is operationalized as simulability: whether humans can use visualizations to predict the behavior of the model in new circumstances [21, 33, 42, 43, 47, 56, 63]. Though the theoretical grounding of this method is perhaps less clear than the mental model accuracy paradigm of Kulesza et al. [41] or the cognitive load paradigm of Abdul et al. [1], simulation tasks have the advantage of being model-agnostic and easy to analyze programmatically, which allows them to be used in semi-automated human-in-the-loop optimization procedures such as Lage et al. [43]. However, such simulability tasks generally do not present "the whole truth" of the model at once. Our method extends the simulability paradigm from discriminative to generative modeling, but in a way that presents the whole truth of the model.

2.2 Interpretable Representation Learning

Background. Representation learning is generally concerned with finding ways of associating high-dimensional *instances*, which we denote x, with low-dimensional *representations*, which we denote z. Representation learning methods roughly fall in two categories: *generative modeling*, which maps low-dimensional representations z to high-dimensional instances x, and embedding, which maps high-dimensional instances x to low-dimensional representations z. Examples of generative models include Hidden Markov Models [24], Latent Dirichlet Allocation [9], and GANs [26]. Examples of embeddings include t-SNE [50] and the latent spaces of deep classification models. Examples of both simultaneously (i.e. approximately invertible mappings) include PCA [34] and autoencoders [32].

Disentangled representations and disentanglement measures. Disentangled representations [8, 18, 20, 31] seek mappings between high-dimensional inputs and low-dimensional representations such that representation dimensions correspond to the ground-truth factors that generated the data (which are presumed to be interpretable). To evaluate a model's disentanglement, many papers compute *disentanglement measures* with respect to known ground-truth factors. However, not only do there exist many competing

measures [17, 22, 30, 31, 49, 62], but, to our knowledge, there exists no work that compares them to human notions of understandability. Our work both performs this comparison and provides a way to evaluate representation learning methods on real-world datasets, where we cannot rely on disentanglement measures because the ground truth is generally unavailable to us.

Explaining and visualizing representations. To be understood, a representation must be visualized or explained. For generative models, this usually means explaining each dimension. Many works [17, 18, 31, 37, 38] show a "latent traversal" of instances with linearly varying values of a specific dimension and constant values of other dimensions. Others find or construct "exemplar" instances that maximize or minimize particular (combinations) of dimensions [3, 54, 55]. Another visualization technique, less common because it requires setting up an interactive interface, is to let users dynamically modify representation dimensions and see how corresponding instances change in real time, e.g. using sliders [27]. Related approaches have been explored for discriminative models with predefined meaningful features, e.g. Krause et al. [40], Wexler et al. [68], and Cai et al. [13], who use sliders to display exemplars matching user-defined concept activation vectors [36]. However, to our awareness, interactive manipulation of autonomously learned generative model dimensions has not been considered in an HCI context, especially to quantify interpretability. We use interactive slider-based visualizations as a subcomponent of our task, and test against baseline approaches using exemplars and traversals.

We do note there are other representation learning visualization methods that could be used in interpretability quantification but do not provide insight into the individual meanings of dimensions and are often specific to embeddings rather than generative models. Examples include Praxis [15], the embedding projector [65], and a parallel-coordinates-inspired extension by Arendt et al. [4], which further reduce embedding dimensionality down to 2D or 3D with PCA or t-SNE [50]. All of these methods are interactive and can help users understand the geometry of the data and what information the representation preserves, but they do not explain the dimensions of variation themselves. Bau et al. [6, 7] both visualize and quantify the interpretability of representations by relating dimensions to dense sets of auxiliary labels, but their approach (while highly effective) is not applicable to most datasets. Our focus is on quantifying how well users can understand representations on their own, in terms of their dimensions, without further projection or side-information.

3 APPROACH

We now define our proposed task for evaluating the interpretability of generative models, which we call "interactive reconstruction," and ground it as a measure of understanding.

3.1 The Interactive Reconstruction Task

Assume we are given a generative model $g(z)\colon \mathbb{R}^{D_z} \to \mathbb{R}^{D_x}$, which maps from representations z to instances x. Also assume we are given a distribution $p(Z)$ from which we can sample representations z (which may be approximated by sampling from a dataset $\{z^{(1)}, z^{(2)}, ..., z^{(N)}\}$), as well as a list of permitted domains $\mathcal{Z} = (\mathcal{Z}_1, \mathcal{Z}_2, ..., \mathcal{Z}_{D_z})$ for each dimension of z (e.g. a closed interval if

z_i is continuous, or a set of supported values if z_i is discrete). Finally, assume we are given some distance metric $d(x, x')$ and threshold ϵ.

The task consists of a sequence of N_q questions. For each, we sample two representation values z, z' iid from $p(Z)$, which have corresponding instance values $x = g(z)$ and $x' = g(z')$. In an interface, we visualize x and x' along with their distance $d(x, x')$ and its proximity to the threshold ϵ. We also present manipulable controls that allow the user to modify each component of z within $\mathcal{Z}$ to interactively change $x = g(z)$ and thus the distance $d(x, x')$. The user's goal is to manipulate the controls as efficiently as possible to bring $d(x, x') \leq \epsilon$, thereby reconstructing the target. Once this condition is achieved, we repeat the process for newly sampled z and z'. If a user actively attempts to solve a question for a threshold amount of time T to no avail, we permit them to move on.

During each question, we continuously record the values of z, the errors $d(x, x')$, the dimensions i being changed, and the directions of modification (increasing or decreasing). These records let us precisely replay user actions and from them, derive a rich set of performance metrics, which we enumerate in Section 4.4. We hypothesize that, while the task will be possible to complete without understanding the model, users will perform it more reliably and efficiently when they intuitively understand what representation dimensions mean. We also hypothesize that the process of performing it will *teach* users these intuitive meanings when they exist—that is, when the model is interpretable.

The above is a general definition of the interactive reconstruction task. To apply it to a specific problem, however, a number of implementation choices must be made:

- **Control of representations** z: There are many different ways of inputting values for representation dimensions (e.g., numeric fields vs. sliders for continuous dimensions, or radio buttons vs. dropdowns for discrete dimensions), and also many different ways of arranging and annotating them (e.g., allowing reordering, grouping, and labeling, which can be helpful for higher-dimensional z).
- **Visualization of instances** x: Although some instance modalities have canonical visualizations (e.g., images), others can be visualized in many ways (e.g., patient medical records). Visualizations should make it easy to recognize the effects of changing z and compare whether x and x' are becoming closer with respect to $d(\cdot, \cdot)$.
- **Choice of distance metric** $d(x, x')$, **distance threshold** ϵ, **and time threshold** T: These critical parameters are best chosen together. For our experiments, we relied on small studies for each dataset, seeking $d(\cdot, \cdot)$ and ϵ that captured when users subjectively felt they had manipulated x to match x' sufficiently closely, and setting T to a round number on the order of twice the median duration.

We describe our dataset-specific choices in Section 4.3, but recommend these be re-tuned for new applications.

3.2 Theoretical Grounding for the Interactive Reconstruction Task

A number of sources in the HCI literature motivate and ground our interactive reconstruction task as a meaningful measure of understanding. First, as recommended by Kulesza et al. [41], our

method attempts to present the "whole truth" and "nothing but the truth:" we visualize the entire model without any simplifying approximations. For generative models, visualizing the full model also means, to the greatest extent possible, visualizing dimensions jointly rather than separately (as is the case when using, e.g., latent traversals). We adopt this strategy in line with cognitive load theory as articulated by Sweller [66], who argues that "understanding" emerges from explaining interactions between elements of a schema ("the basic unit of knowledge"), and that explaining interacting elements separately will make material harder to understand, rather than easier to understand, because of the split attention effect [16].

We also attempt to avoid split attention effects between the visualization and the proxy task. In many interpretability measurement methods, these two components are visually separate, e.g., in different portions of the screen [33, 42, 43, 56]. Such physical separation often makes it possible or even preferable to complete the task without engaging with the visualization, e.g., by guessing multiple-choice answers to complete the task more quickly. In recent HCI studies of interpretability tools, both Buçinca et al. [11] and Kaur et al. [35] note an inconsistency between outcomes in thinkaloud studies, where cognitive engagement with a visualization is forced, and practice, where the visualization can be readily ignored. By closely integrating the visualization and the task, we attempt to avoid this pitfall.

Finally, per Norman's articulation of the importance of feedback for building understanding in human-centered design [53], our task is structured to provide immediate, interactive feedback. Just from looking at the screen, it is always readily apparent to users whether they have gotten the right answer, and whether they are getting closer or further away. In contrast, for most simulation-proxy interpretability measurement tasks [33, 42, 56, 63], if users receive feedback at all, it is sporadic, e.g., about whether a multiple choice answer was correct.

3.3 Baseline: the Single-Dimension Task

As a baseline to our interactive reconstruction task, we also consider a "single-dimension" task, inspired by existing interpretability measurement methods for discriminative models (Section 2.1). Here, users are given an instance $x = g(z)$ and asked to guess the value of some hidden dimension z_i. To help them, users may view visualizations of other instances as dimension i is varied, shown in a different section of the screen. For each model, users are asked N_q questions about each dimension, and receive feedback after each question about whether their answer was correct. Details for our dataset-specific implementations are in Section 4.5.

The single-dimension task is deliberately designed to violate our theoretical motivations; specifically, (1) it does not present "the whole truth" of the model, (2) it visualizes dimensions individually rather than jointly, (3) it spatially separates the visualization from the task, and (4) it provides feedback sporadically (after each question) rather than interactively.

3.4 Quantifying Quality of Interpretability Measurement Methods

Measuring the accuracy of any measurement method generally requires testing with a precisely known quantity. In this paper,

we *assume ground-truth knowledge* that a particular model is more interpretable than another by working with synthetic datasets constructed for the express purpose of making this assumption reasonable, and supported by qualitative studies. We then test how well different interpretability measurement methods detect this assumed ground-truth difference. Where possible, we support our assumptions with quantitative measures of models' correspondence to ground-truth, e.g. disentanglement measures.

Our approach contrasts with the strategy of Buçinca et al. [11], who evaluate interpretability evaluation "proxy tasks" by how well they predict performance on downstream tasks. Although the ultimate motivation for interpretability research is to improve performance on downstream tasks ranging from better human+AI collaboration [5] to auditing for safety [14], performance on these tasks has no explicit correspondence to understanding. Additionally, in research settings, we often lack access to the true downstream tasks that motivate our work; thus our downstream validation task is *also* a proxy. So in effect, we are not evaluating whether a proxy task measures interpretability, but whether one proxy predicts another.

4 IMPLEMENTATION

We now describe how we implement our approach for a variety of representation learning models and datasets.

4.1 Datasets

We considered three datasets, visualized in Figure 1:

- **dSprites** [51], a 64×64 binary image dataset with five ground-truth factors of generation z: shape, rotation, size, x-position, and y-position. We chose dSprites due to its popularity in the disentanglement literature (Section 2.2).
- **Sinelines**, a 64-dimensional timeseries dataset we developed for this study. Each instance $x = \langle x_1, x_2, ..., x_{64} \rangle$ is a mixture of a line and a sine wave, generated by mapping: $x_t = z_1 t + z_2 + z_3 \sin(z_4 t + z_5)$, where $z_1 \sim \text{Uniform}(-1, 1)$ is slope, $z_2 \sim \mathcal{N}(0, 1)$ is intercept, $z_3 \sim \text{Exponential}(1)$ is amplitude, $z_4 \sim \text{Exponential}(1)$ is frequency, and $z_5 \sim \text{Uniform}(0, 2\pi)$ is phase. We make z 5-dimensional for consistency with dSprites, but make x a timeseries rather than an image to probe sensitivity to instance modality.
- **MNIST** [44], a popular benchmark in the ML and interpretable representation learning literature consisting of images of handwritten digits from 0 to 9. Although the MNIST dataset lacks ground-truth representations, it does contain labels indicating depicted digits.

4.2 Models

On dSprites and Sinelines, we tested the following models:

- **"Ground-truth" (GT)** generative models, constructed to have the same relationship between x and z as the (intentionally interpretable) process which generated the examples.
- **Autoencoders (AE)** [32], a baseline nonlinear representation learning method often considered to learn uninterpretable representations.
- **Variational autoencoders (VAE)** [38], which are similar to autoencoders but learn distributions over z, with a prior

on z that can be interpreted as a regularizer. Because of this regularization effect, VAEs are widely reputed to learn more interpretable relationships between instances and representations than standard autoencoders.

Our primary goal was to validate that our task could distinguish entangled autoencoders from disentangled ground-truth models (both absolutely and relative to baselines).

Architecturally, for dSprites, we used the same 7-layer convolutional neural networks (CNNs) as Burgess et al. [12], one of the original papers introducing dSprites (with the GT model just using the decoder). For Sinelines, we used 256×256 fully connected networks with ReLU activations (except for the GT model, which was simple enough to implement in closed form). We tested all models at $D_z = 5$ to match ground-truth.

On MNIST, which has no ground-truth model, our goal instead was to test a broader set of popular interpretable representation learning methods that have never been tested with user studies. We again tested on AEs and VAEs, but also included the following models from the disentanglement literature:

- **β-TCVAEs (TC)** [17], a modification of the VAE that is trained to learn representation dimensions z_i that are statistically independent, generally considered near state of the art in the disentanglement literature.
- **Semi-supervised β-TCVAEs (SS)**, a modified β-TCVAE we explicitly train to disentangle digit identity from style (in a discrete dimension). Though we lack full ground truth, we expect SS to be less entangled than TC.
- **InfoGAN (IG)** [18], a generative adversarial [26] disentanglement method that also learns to disentangle digit identity from digit style, but imperfectly and without supervision.

We tested all of these MNIST models (AE, VAE, TC, SS, and IG) at $D_z = 5$ but additionally tested AE, TC, and SS at $D_z = 10$ to probe sensitivity to representation dimensionality. Architecturally, for all models, we use the same CNN architecture given in the papers introducing InfoGAN [18] and β-TCVAE [17] (Table 4), changing only the size of the latent dimension D_z. Training details were chosen to match original specifications where possible; see Section A.1 of the Appendix for additional details as well as model loss functions.

On dSprites and Sinelines, we quantified the extent to which our models matched ground truth with disentanglement measures. Specifically, we computed the DCI disentanglement score [22] and the mutual information gap (MIG) [17], which are commonly included in disentanglement papers. In addition to these overall scores, we computed pairwise mutual information to visualize dependence on a per-dimension basis. Figure 2 shows each of these metrics for each dimension and dataset. On both datasets, GT models are perfectly disentangled and AE models are heavily entangled. VAEs are somewhere in the middle, partially disentangling horizontal and vertical position from shape, scale, and rotation on dSprites, and partially disentangling linear from sinusoidal factors on Sinelines. As mentioned previously in Section 3.4, we use these metrics to support our assumption that GT models are more interpretable than AEs. On MNIST, we have no ground truth model, but we hypothesize that the semi-supervised (SS) model will be most interpretable due to disentanglement between its continuous dimensions and ground-truth digit identity.
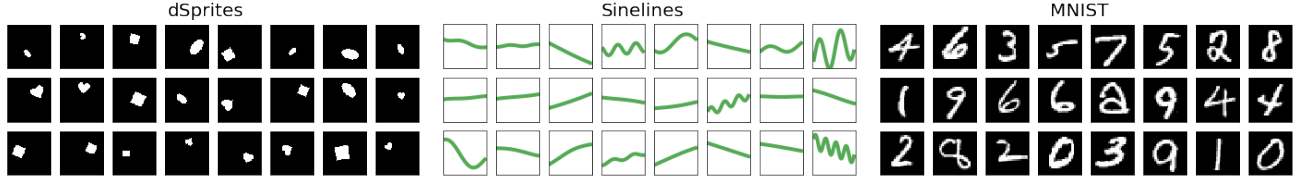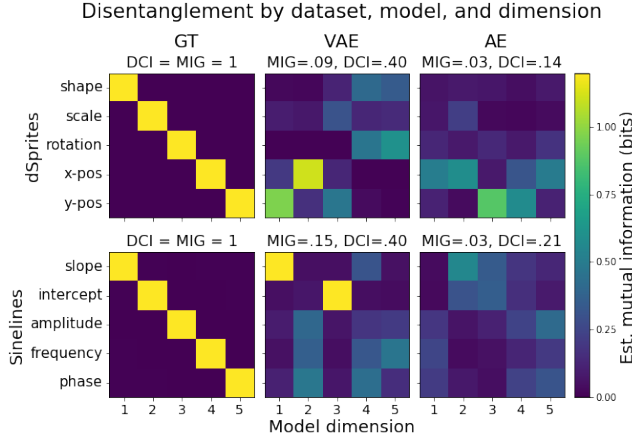
Figure 1: Examples from the dSprites, Sinelines, and MNIST datasets.

Figure 2: Disentanglement scores (in plot titles) and pairwise mutual information (in heatmaps, approximated using 2D histograms) between true generative factors and representation dimensions. By construction, ground-truth (GT) models are perfectly disentangled, while VAEs learn to partially concentrate information about certain ground-truth factors into individual representation dimensions. AEs exhibit less clear relationships and have the lowest disentanglement scores.

4.3 Interface

Figure 3 shows examples of the user-facing interface for this method, implemented for two different datasets. The following interface elements correspond to the interactive reconstruction task parameters described in Section 3.1:

Control of representations z: To specify closed-interval continuous dimensions of z, we use slider components, while to specify finite-support discrete dimensions, we use unlabeled radio buttons. We sampled initial z and target z' from $p(Z)$ set to the empirical distribution, or specifically a heldout testing split of the dataset. Slider ranges were determined by taking the empirical minimum and maximum values of each dimension z_i over this heldout split. The one exception to this procedure was for the InfoGAN model, where sampling and limits were determined from the prior.

Visualization of instances x: For images, visualization was straightforward, while for Sinelines, we used line charts with dashed lines at $y = 0$ and appropriate limits. In addition to visualizing x and x' side-by-side, we provided an option to overlay them with partial transparency, which we found was helpful in pilot experiments for fine-tuning. This defaulted to on for our synthetic experiments,

but was defaulted to off for MNIST after pilot users expressed the side-by-side view was more helpful. For MNIST in particular, to facilitate remembering (and recording) user impressions of dimension meanings, we allowed users to input custom labels next to the corresponding controls in the interface.

Distance and time thresholds: For each dataset, we defined $d(x, x')$ as a Jaccard distance [45], i.e., the fraction of disagreeing dimensions of x and x' to total active dimensions of x and x', with dataset-specific definitions of agreement and activity. Exact expressions are given in Section A.2 of the Appendix. Although these choices worked well for black and white images and consistently-scaled timeseries, different metrics might be necessary for other data modalities; we discuss this further in Section 7. Because $d(x, x')$ was between 0 and 1, we visualized it as an agreement percentage rather than a distance, and chose $\epsilon = 0.1$ (or a 90% agreement threshold) for synthetic datasets and $\epsilon = 0.25$ (or a 75% agreement threshold) for MNIST.

The time threshold T for skipping questions was set to 30 seconds for dSprites and Sinelines and 45 seconds for MNIST. We paused this hidden countdown whenever users were inactive for more than 3 seconds.

We implemented tasks as single-page, client-side web applications, with machine learning models running directly in users' web browsers after being converted to TensorFlow.js [64]. Despite the fairly large size of certain models (e.g. 7-layer convolutional neural networks), only two users out of hundreds reported problems running them interactively in-browser. Links to the task for all models can be found at http://hreps.s3.amazonaws.com/quiz/manifest.html.

4.4 Interactive Reconstruction Metrics

We computed the following performance metrics from our records of users performing the interactive reconstruction task:

- **Completion rate:** For each model, we measure the fraction of questions the user solves, i.e. moves the sliders and/or chooses radio buttons to make $d(x, x') \leq \epsilon$, rather than skipping after T seconds.
- **Response time:** For each model, we measure the average time it takes participants to complete each question.
- **Slide distance:** For each model, we measure the average total distance moved by sliders, in units of full slider widths, for each question. Discrete changes count as 1.
- **Error AUC:** We measure the "area under the curve" of mean squared difference between x and x' over the total time the user is attempting to solve the quiz.
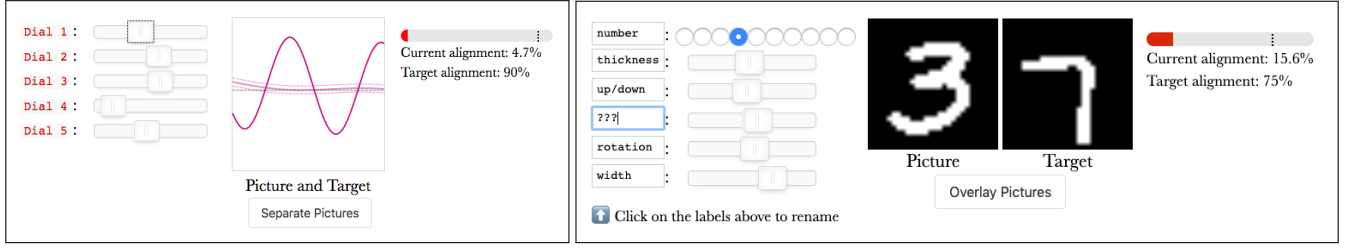
Figure 3: Screenshots of the interactive reconstruction task on Sinelines (left, with x and x' overlaid and dotted lines indicating the region of allowable alignment) and MNIST (right, with separated x and annotations for z).

- **Self-reported difficulty:** After answering a "stage" of questions about a model, users rate that difficulty from 1 to 7 using a Single Ease Question (SEQ) [59].

We store snapshots of x and z every 100ms while the value is changing in a constant direction, or whenever the active direction or dimension changes. While this list is not an exhaustive enumeration of all possible metrics applicable to the task, others can be defined and computed post-hoc as we store a nearly complete record of user actions over time.

4.5 Single-Dimension Task Parameters

In this section, we describe how we instantiated the single-dimension task for the synthetic datasets. This task has two primary implementation parameters: the prediction task and the visualization.

For the prediction task, we opted for a multiple-choice classification task where, for a particular example x visualized in the same way as in the interactive reconstruction task, users decided whether z_i is "Low," "Medium," or "High." These regimes were respectively defined in terms of the 1st-5th, 48th-52nd, and 95th-99th percentiles of the marginal distribution of encoded x in a held-out split of the original dataset. We sampled z by first sampling from the empirical joint distribution (i.e. a heldout split of the dataset), then overriding z_i to a value selected uniformly from one of these regimes. Users answered two questions for each dimension i, and received the correct answer as feedback.

For the visualization, which is the sole task component dependent on the model, we tested two versions of the task, one using *latent traversals* and the other using *synthetic exemplars*, as described in Section 2.2. For traversal visualizations, for 5 randomly sampled values of z, we plotted x values corresponding to overriding z_i to 7 linearly spaced values between "Low" and "High" as defined above. For exemplar visualizations, we showed "Low," "Medium," and "High" bins with 8 examples each. Both of these visualizations force users to generalize from finite samples, which can lead to ambiguities if randomly sampled z are not diverse. To mitigate this potential problem, we provided users with a "Show More Examples" button. Screenshots for the single-dimension task on the dSprites dataset are shown in Figure 4.

4.6 Single-Dimension Metrics

We recorded several performance metrics specific to the single-dimension task:

- **Correctness rate:** For each model, we measured the percentage of questions the participant answered correctly.
- **Self-reported confidence:** For each model, we measured whether users agreed with the statement "I'm confident I'm right" on a 5-point Likert scale.
- **Self-reported understanding:** Similar Likert measurement, but for "The dial makes sense." (We referred to dimensions as "dials" in the interface.)

In addition to these task-specific metrics, we also recorded response time and self-reported difficulty with the SEQ, which were shared in common with interactive reconstruction.

5 STUDY METHODOLOGY

With implementation decisions now specified, we describe the four studies we ran, two larger-scale studies on MTurk and two smaller-scale lab study sessions (see also Figure 5 for a graphical depiction). All experiments began with a consent form, a tutorial, and practice questions on an easy example. Experiments were deployed on the web and are available at http://hreps.s3. amazonaws.com/quiz/manifest.html. Code for our study is available at https://github.com/dtak/interactive-reconstruction.

5.1 Experimental Design

5.1.1 Synthetic Think-aloud Pilot. We began by running a small pilot study with $N = 3$ users (referred to as U1, U2, and U3) to test out each possible task on the dSprites and Sinelines synthetic datasets (interactive reconstruction, single-dimension with exemplars, and single-dimension with traversals), with each user completing two of the six task/dataset conditions with two of the six models (drawn randomly without replacement to ensure complete coverage). Participants were asked to think aloud and describe their strategies for solving each problem. Interviews were recorded and transcribed, and the feedback was used to make minor clarifying changes to the interface.

5.1.2 Synthetic MTurk Study. We then ran a larger version of the synthetic dataset study on MTurk, where each participant ($N = 15$ per dataset and task, and $N = 90$ total) completed one of the six possible dataset/task conditions, but for all three models (AE, VAE, and GT, with the order randomized). To keep overall quiz length manageable, N_q was set to 5 for interactive reconstruction and 10 for single-dimension tasks (that is, 2 questions per dimension). Differences were analyzed with repeated-measures ANOVA and paired t-tests.
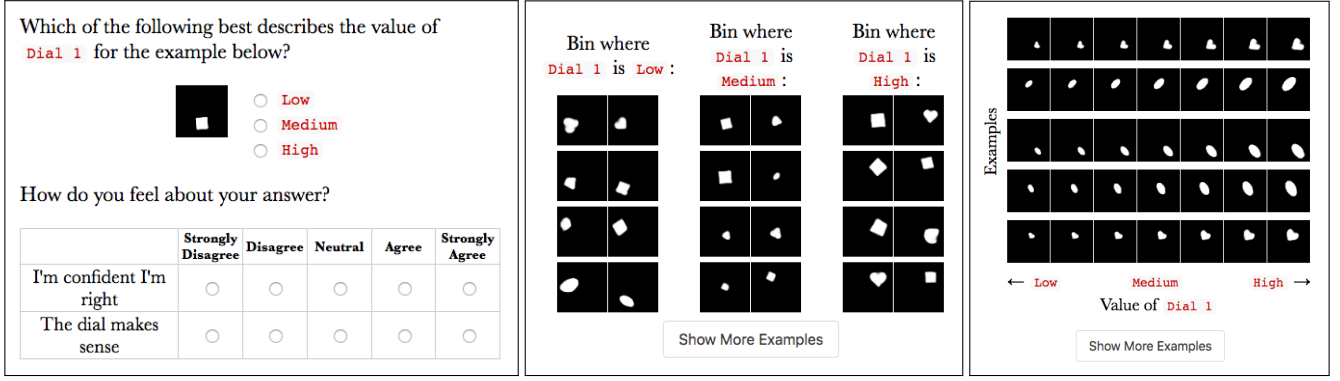
Figure 4: Screenshots of the dSprites single-dimension task (left), showing ground truth visualizations with exemplars (center) and traversals (right).
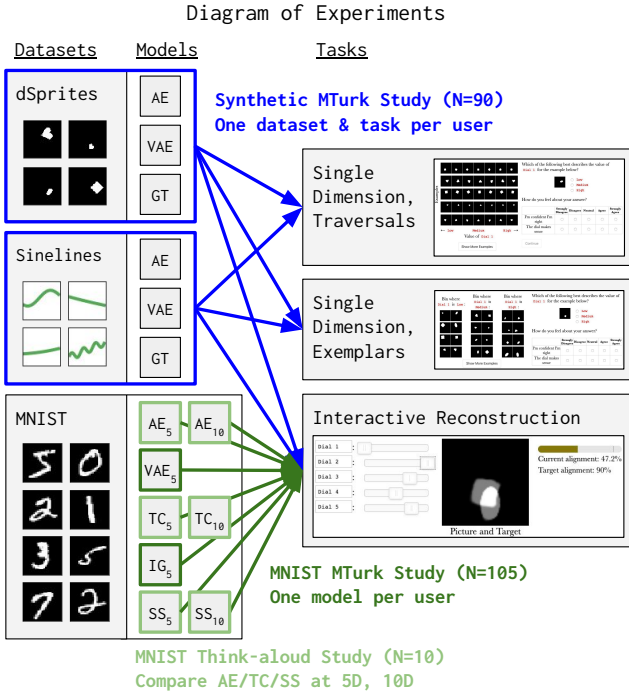

Figure 5: Diagram of experiments performed (except the small synthetic pilot). On synthetic datasets, experiments were designed to evaluate the best task (assuming the best model). On MNIST, experiments were designed to evaluate the best model (assuming the best task).

5.1.3 MNIST Think-aloud Study. We next ran a think-aloud study with $N = 10$ participants on MNIST, testing out a wider variety of models but only for the interactive reconstruction task. Each participant (whom we refer to as P1-10) completed interactive reconstruction for three models (AE, TC, and SS, order drawn randomly without replacement), with odd and even-numbered participants working with 5D and 10D representations respectively. Model-specific stages ended after 7 questions had been completed or 10

minutes had elapsed. After each stage, in addition to the single ease question [59], users entered raw NASA-TLX scores [29] and answered Likert scale questions about whether they felt they understood representation dimensions. Differences were analyzed with repeated-measures ANOVA and paired t-tests. During each stage, users were also encouraged to label dimensions if possible and continuously describe their impressions of the task, which we recorded.

5.1.4 MNIST MTurk Study. Finally, we ran a large-scale MTurk study on MNIST. As with the synthetic dataset study, we had $N = 15$ participants per condition (for $N = 105$ participants total). However, instead of showing all models and varying the task, we limited the task to interactive reconstruction and varied the model, specifically showing each participant one of AE_5, VAE_5, IG_5, TC_5, SS_5, AE_{10}, or SS_{10}. N_q was increased from 5 to 7 for additional signal and to make the task sufficiently long. Differences were analyzed with one-way ANOVA and independent t-tests.

5.2 Recruitment

5.2.1 Think-aloud Studies. For our think-aloud studies, we recruited undergraduate and graduate students from computer science mailing lists at an academic institution, and compensated participants with Amazon gift cards ($15/hour for synthetic pilot study sessions, which we increased to $20/hour for MNIST to incentivize recruitment for a larger study). We recruited $N = 3$ participants for the synthetic pilot and $N = 10$ for MNIST.

On the synthetic datasets, two participants were male, one was female, and all were graduate students. On MNIST, two were male, eight were female, three were undergraduates, and seven were graduate students. All participants in both studies were aged 18-34.

5.2.2 MTurk Studies. For each MTurk experimental condition (6 for synthetic datasets and 7 for MNIST), we recruited $N = 15$ participants on Amazon Mechanical Turk with unique worker IDs. This translates to a total of 90 participants on the synthetic dataset study and 105 participants for the MNIST-based study.

For single-dimension tasks, participants were excluded if they answered practice questions incorrectly. For traversal visualizations, the retention rate was 71%, while for exemplars, it was 91%.

For interactive reconstruction, we included all participants who successfully completed the study on synthetic datasets (as users could not proceed past the practice questions without answering them correctly), but added an additional inclusion criteria on MNIST requiring participants to have reconstructed at least one of seven instances successfully. We adopted this criteria in part because we noticed several participants did not realize they could use radio buttons (which were not included in the practice questions) on the IG and SS tasks (never changing them during any question), and in part to filter out users who gave up completely on extremely hard models (e.g. the 10D AE). Retention rates were highest (88%) for the SS_{10} model and lowest (42%) for the AE_{10}, with an overall average of 67%. In total, 156 participants were needed to get 105 retained participants on MNIST and 115 participants were needed to get 90 retained participants on synthetic tasks.

Compensation was set with the intention of ensuring that actively engaged MTurk users would earn at least $15/hr, with time estimates based on pilot experiments and interactive reconstruction time limits. On the synthetic datasets, users were paid $6 for both single-dimension and interactive reconstruction tasks, which took an average of 13 minutes ($27/hr). The interactive reconstruction experiments took a slightly longer average of 18 minutes ($20/hr), with 95% of participants finishing under 34 minutes ($11/hr). For the MNIST dataset, users were paid $3.75 for completing tasks, which took an average of 15 minutes ($15/hr), with 80% of participants finishing under 20 minutes ($12/hr) and 95% finishing under 38 minutes ($6/hr).

Demographic information was recorded in a post-quiz questionnaire. In the synthetic study, participants were generally young (58% between 18-34), North American (68%, with most others from Asia or South America), college-educated (79% had Bachelor's degrees or above) and male (64%), with gender recorded per guideline G-4 of [60]. Demographics were similar on MNIST (49% aged 18-34, 59% North American, 71% male, and 81% with Bachelor's or above).

6 RESULTS

Since the think-aloud and MTurk study participants worked with the same datasets and completed similar or identical tasks, we will first present results from all the synthetic dataset studies and then move on to results from the MNIST dataset studies. Adding up ANOVAs and t-tests across all our experiments and pairs of models, we ran 435 statistical tests, giving us a Bonferroni-corrected threshold of 0.00011 for an initial α of 0.05. Means, standard deviations, and pairwise p-values for all metrics across all experiments are given in Section A.3 of the Appendix.

6.1 Synthetic Study Results

6.1.1 MTurk, Interactive Reconstruction. As shown in Figure 6, **the interactive reconstruction task clearly differentiated ground truth (GT) models from AEs.** Completion rates were significantly different on both dSprites ($F_{2,28} = 32.3, p<0.0001$) and Sinelines ($F_{2,28}=23.4, p<0.0001$) in the directions we hypothesized. GT model completion rates on both dSprites (.75±.26) and Sinelines (.83±.19) were significantly higher than AE completion rates (.15±.19, $t_{14}=8.5, p<0.0001$ on dSprites and .34±.26, $t_{14}=6.4, p<0.0001$ on Sinelines). VAE models, though only marginally significantly

different, were in the middle (.51±.33 on dSprites and .63±.28 on Sinelines). These results closely mirror the disentanglement metrics in Figure 2.

We also saw significant differences in self-reported difficulty ($F_{2,28} = 32.3, p<0.0001$ on dSprites, $F_{2,28}=11.7, p=0.0002$ on Sinelines) and error AUC ($F_{2,28}=31.6, p<0.0001$ on dSprites, $F_{2,28}=14.1, p<0.0001$ on Sinelines). Between AE and GT models, these differences were large and least marginally significant. VAEs were no longer exactly in the middle, however. Instead, they more closely matched the AE model on dSprites and the GT model on Sinelines— e.g. for difficulty, average scores for the AE, VAE, and GT were 6.9, 6.4, and 4.2 on dSprites vs. 5.9, 4.1, and 4.0 on Sinelines.

6.1.2 Think-aloud, Interactive Reconstruction. **Interactive reconstruction task users felt they understood GT models, did not understand AEs, and partially understood VAEs** in a way that matched the disentanglement scores in Figure 2. On dSprites, U1 assigned meanings to GT model dimensions almost immediately, noting on the first question that *"[dial 1] is changing its shape, [...] dial 2 is changing the size, [...] and then 3 seems like it's changing the rotation, and 4, probably x-position. And then 5 is y-direction I guess?"* For the AE, however, U1 *"d[id]n't understand what the dials [we]re doing."* On Sinelines, U2 found the GT model *"much easier to figure out"* than the AE, where it was *"hard to tease out exactly what each [dimension] is doing."* The difficulty of understanding the AE model caused them to switch strategies from *"match[ing] [dimensions] sequentially"* (which they could do with the GT model) and instead *"look[ing] at the current alignment and then mak[ing] sure that number is increasing"* by *"tweak[ing] the dials and see[ing] what happens."*

Users partially understood VAEs. On the Sinelines VAE, U2 reported they could *"quickly figure out"* there *"was one dial that [controlled] slope"* and another that *"did [vertical] translation"*, but that *"squiggliness"* was *"quite hard."* These reports match the dependence plots in Figure 2, which show that the Sinelines VAE disentangled linear slope and intercept from sine wave parameters. On the dSprites VAE, U1 felt *"dials 1, 2 and 3 are more predictable"* after determining *"dial 3 is changing the size"* while *"dials 1 and 2 are moving the object in the diagonal way."* These comments again matched the dependence plots in Figure 2, which suggest position and size are controlled by VAE dimensions 1-3, while shape and rotation are controlled by dimensions 4-5. However, U1 did find the *"diagonal"* relationship (which was also nonlinear) confusing, which forced them to randomly experiment more within each group of dimensions. In contrast, for U2, linear slope and intercept were disentangled both from sine wave dimensions and each other. This difference may explain why, relative to the GT, VAEs had higher MTurk difficulty ratings and error AUCs on dSprites than Sinelines (despite having similar DCI scores).

6.1.3 MTurk, Single-Dimension. **The single-dimension task did not clearly differentiate models.** In the MTurk results, no differences in any metrics were significant at $\alpha = 0.00011$, though correctness rates for the exemplar-based version of the task were closest ($F_{2,28}=5.8, p=0.008$ for dSprites and $F_{2,28}=6.6, p=0.005$ for Sinelines, though the most significant relationships were in the wrong direction). Response time and self-reported confidence/understanding were almost identical across models, datasets, and visualizations.
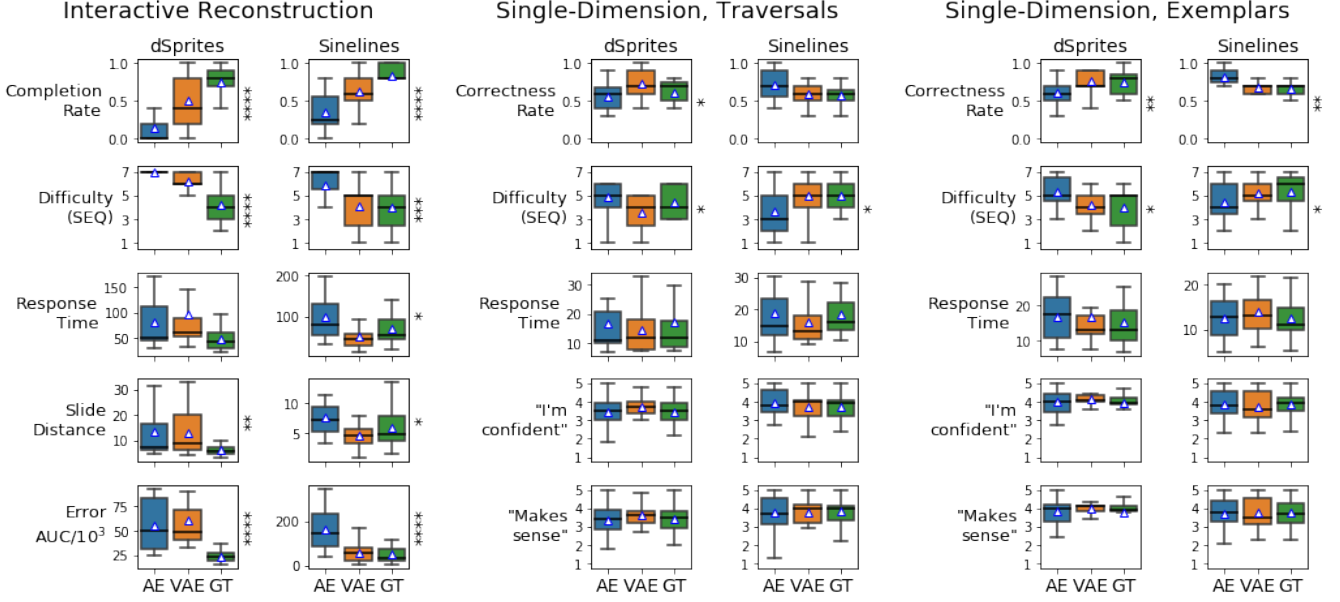
Figure 6: Boxplots of MTurk dependent variables across the 15 participants for each synthetic data task condition. Blue triangles indicate means, black lines indicate medians, and stars indicate p-values for differences between means (****=p<0.0001, ***=p<0.001, **=p<0.01, *=p<0.05). Expected differences between AE and GT models emerge clearly from interactive reconstruction results, but not single-dimension results. The same results are repeated as tables in Table A.1 of the Appendix.

On dSprites, we did observe that GT models had higher correctness rates (.75±.14) and lower difficulty ratings (4.0±1.6) than AEs, which had .61±.18, p=.01 for correctness and 5.3±1.3, p=.03 for difficulty (on exemplars). However, on Sinelines, AEs actually emerged with the *highest* average correctness rates in the exemplar-based MTurk study (AE=.81±.15 vs. GT=.65±.11, p=0.003), as well as the lowest difficulty (AE=4.5±1.6 vs. GT=5.3±1.5, p=0.003). Our qualitative results below (as well as the dimension-by-dimension correctness rate breakdown in Figure 7) suggest this is not because users understood the AE, but because of helpful AE pathologies and unhelpful GT symmetries.

6.1.4 Think-aloud, Single-Dimension. **Users performed the task without trying to understand the model.** Although U1 initially tried to formulate *"hypotheses"* about the meanings of dimensions, they found these *"irrelevant for [them] to make decisions."* U2 described their strategy as *"match[ing] directly"* without trying to *"tease apart the different dimensions,"* and P3 called it *"blind trust"* in *"visual matching."*

Visualization pathologies also biased the results in directions unrelated to understanding. On Sinelines, we noticed a lack of diversity in "High" and "Low" samples from AEs, which sometimes made the matching problem trivially easy for all participants. For GT models, we also noticed questions for particular were pathologically hard due to symmetries in the generative process. One example is that for the rotation feature on dSprites and the phase feature on Sinelines, "Low" values near 0 and "High" values near 2π were nearly indistinguishable. For GT models on Sinelines, participants also found it hard to detect changes in amplitude at low frequencies, as these could be alternately explained by slope, and harder still
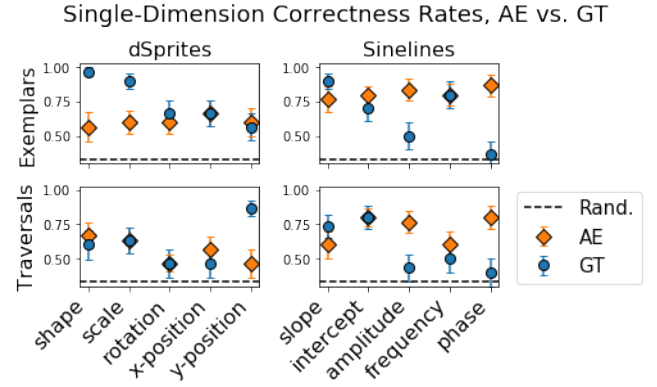
Figure 7: AE and GT correctness rates by representation dimension for MTurk single-dimension tasks. Dots show means with standard errors, x-axis shows GT dimensions (AE order is arbitrary). Compared to AEs, some GT dimensions had much lower or higher correctness rates than others, rising to near 100% for attributes like shape, scale, and slope and falling to near 33% (the rate of random guessing) for others, especially periodic attributes like rotation and phase. These differences suggest that questions about certain conceptually simple dimensions were pathologically difficult to answer from static visualizations.

to detect changes in frequency at low amplitudes, as near-linear examples were effectively unaltered.

6.1.5 Exemplars vs. Traversals. Though not central to our narrative, we found some evidence that exemplars may have been more effective than traversals for single-dimension performance. U3, who completed versions with both visualizations, reported that the task was easier with exemplars because seeing instances clustered into bins *"visually help[ed them] understand the three categories,"* whereas with the traversal visualization, it was necessary to *"mentally create"* those categories by imagining spatial *"divisions."* U3 also hypothesized it was easier for them to detect patterns on the right-hand side of traversal visualizations due to their experience *"reading from left to right,"* which they were concerned might introduce *"a bias"* in their answers. We see such evidence of spatial bias in Figure 7, whose bottom-left plot shows that MTurk users struggled to answer traversal questions about dSprites x-position, where the movement of the shape was in the same direction as the movement of the images, but answered questions about y-position almost perfectly, where the movement of the shape was orthogonal to the movement of images. Meanwhile, accuracy on these questions for exemplars (top-left plot) was intermediate in both cases, showing less negative bias for x-position though also less positive bias for y-position. Overall, these results suggest that visualization details matter for performance, but also that it may be better to utilize time rather than space when visualizing changes in spatial features, which is effectively what occurs with interactivity.

6.2 MNIST Study Results

6.2.1 MTurk, Interactive Reconstruction. As shown in Figure 8, **Interactive reconstruction metrics clearly distinguished models, especially AE vs. SS.** In general, we found significant differences between models (F_{28}=4.7, p=0.0003 for completion rate, F_{28}=4.9, p=0.0002 for difficulty, and F_{28}=11.4, p<0.0001 for error AUC), with the sharpest pairwise contrasts being significant or nearly significant differences between AE and SS models. For example, at 5D for the AE vs. SS, completion rates were 0.38 ± 0.23 vs. 0.72 ± 0.24 (t_{28}=−3.9, p=0.0006), subjective difficulty was 5.6 ± 1.6 vs. 3.4 ± 1.5 (t_{28}=3.7, p=0.001), and error AUC (given in units of 1000s for conciseness) was 15.7 ± 4.3 vs. 6.3 ± 4.7 (t_{28}=5.5, p<0.0001). The next-best performing model was the β-TCVAE (TC), with an average completion rate of 0.61 ± 0.27, subjective difficulty of 5.0 ± 1.5, and error AUC of 16.0 ± 14.0. Although the average error AUC was higher for the TC model than the AE, this was largely due to a small number of extreme outliers (note the high variance). Comparing medians, we have 10.5 for the TC model and 14.5 for the AE model, which matches the overall trend of AE < TC < SS. Results for the IG and especially the VAE were generally slightly worse and closer to AE performance.

Performance degraded when increasing dimensionality, but not as badly for methods thought to be interpretable. When we increased D_z from 5 to 10, average completion rate fell (0.38→0.35 for the AE, 0.73→0.71 for the SS), error AUC rose (15.7→47.8 for the AE, 6.3→7.6 for the SS), and subjective difficulty generally rose (3.4→4.4 for the SS, though it fell slightly from 5.6→5.3 for the AE), though none of these results were near significance except the AE's increase in error AUC (t=−3.7, p=0.002). Examining medians, however, we find that median completion rate fell dramatically for the AE (3/7→1/7, the minimum possible

value for retention), while for the SS it actually *rose* (5/7→6/7), suggesting that degradation from increasing dimensionality was more drastic for the AE than the SS.

6.2.2 Think-aloud, Interactive Reconstruction. **Multiple sources of evidence suggest users understood the SS well and the AE poorly**. The first source is subjective measures, which are plotted in Figure 9. Though not always significant at $N = 5$, they suggest not just that the difficulty and cognitive load of the task varied across models (with AE generally rated hardest and SS easiest), but also that dimension understandability varied in the same way, e.g., with SS rated most understandable at 5D with 3.6±0.5 and AE rated least understandable with 1.4±0.5 (t_4=11.0, p=0.0004). When increasing D_z to 10, frustration grew and subjective understanding of SS models fell (to 2.4±0.5, t_4=6, p=0.004), but relative differences remained fairly consistent.

The second source of evidence is agreement between user-entered dimension labels, shown for $D_z = 5$ in Figure 1. On average, only 20% of users were able to assign labels to any AE dimensions, but 100% were able to do so for at least one SS or TC dimension, with all dimensions at $D_z = 5$ and many at $D_z = 10$ having at least 80% coverage. Additionally, for many SS dimensions, these independently-assigned labels agreed closely. TC model labels were less consistent, suggesting more entanglement with the digit.

The third source of evidence is users' verbal descriptions. None of the 10 participants made any comments indicating that they found the AE comprehensible, with P3 commenting that "*the dials didn't have any discernible meaning*", and P8 concluding that "*the meanings of the dials aren't helpful in this one for solving the problem most efficiently.*" In contrast, 9/10 participants felt that the SS model was "*the easiest to understand and label*" (P3), with 100% agreeing it was more comprehensible than the AE. This was in large part "*because it [...] let you choose the number*" (P4, with all participants commenting on this in some manner), though 8/10 participants also expressed that the continuous dimensions were at least partially understandable. For example, P1 noted that they "*understood what most of the dimensions were doing, especially rotating.*" However, unlike with the synthetic GT models, no participants claimed complete understanding; for example, P4 felt "*there were some dials that were easy to tell, but there were others that were more obscure.*"

The TC model consistently fell between the others, with 8/10 commenting that it was more comprehensible than the AE, and 9/10 commenting that it was less comprehensible than the SS. Subjective descriptions varied; compared to the AE, P9 felt able to "*understand and write down and remember*" what TC dials meant, despite being "*worried*" that "*the framework [they] built might not be accurate.*" P7 felt they could "*understand physically*" what certain dimensions were "*trying to do*" but had trouble expressing it verbally; they described it as being like a "*matching game*" where "*you know what you mean but there's not an agreed upon word for it.*"

Differences in understanding led to differences in strategy. When users understood dimensions, they used that understanding to perform the task more efficiently. On the SS model, P5 "*used the dials that [they] understood first*" and "*had an idea of which way to move*" them, while P8 felt it was "*pretty easy to know which [SS] dial to pick.*" P7 was able "*to be intuitive*" when they "*could figure out what the dials meant.*" P1 even felt a degree of mastery,
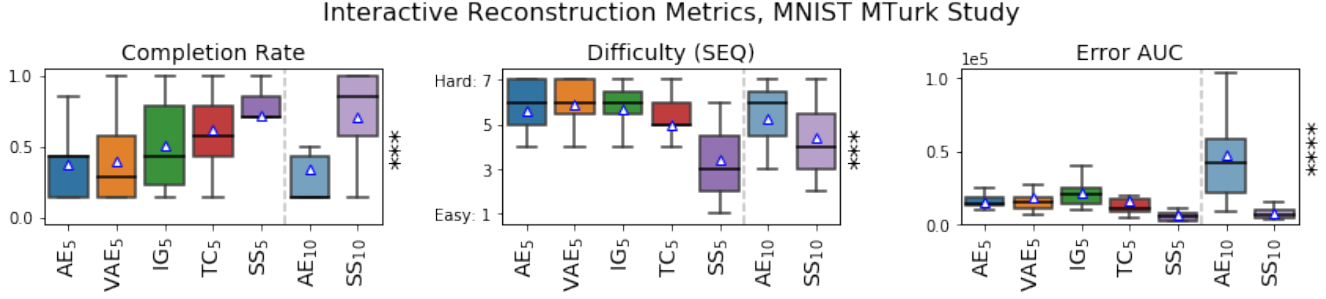
Figure 8: Boxplots across the 15 MTurk participants for each MNIST model, for dependent variables that were significant in the synthetic tasks. Blue triangles indicate means, black lines indicate medians, and stars indicate significance as in Figure 6. As with the qualitative studies, semi-supervised (SS) models performed best by each measure, while standard autoencoders (AE) performed near the worst, especially at higher representation dimensions (right). Pairwise comparisons can be found in Figure A.1 of the Appendix.
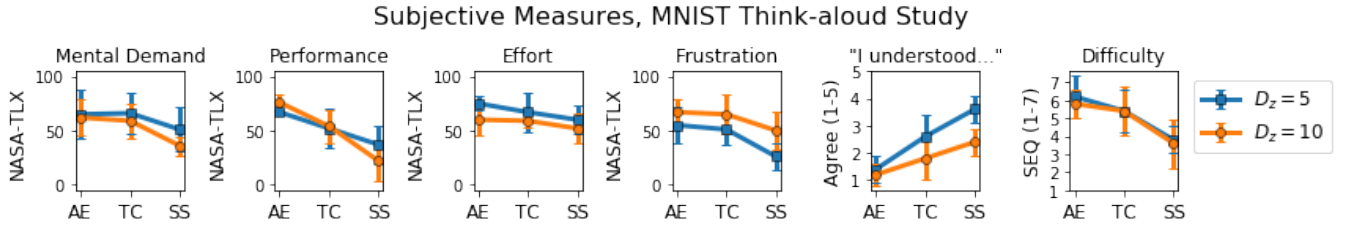

Figure 9: Subjective measures for the MNIST think-aloud study (means ± standard deviations). The first four measures are NASA-TLX scores [29], the fifth is users' subjective agreement (from 1-5) with the statement "I understood what many of the dimensions meant," and the last is a single ease question (SEQ) [59] assessment of difficulty from 1-7. Users rated SS models easiest across all measures, though for higher D_z, subjective understandability was lower, frustration was higher, and perceived performance gaps were greater. An alternate presentation can be found in Figure A.1 of the Appendix.

saying that for the SS model, they "*had it down to a more exact science.*"

In contrast, when users did not understand dimensions, they gravitated towards an inefficient but less cognitively taxing strategy similar to gradient ascent. P3 described this strategy as trying to "*watch one slider and look at the alignment number of see if it reaches a local max,*" with some participants "*proceeding through dials one at a time*" (P10) and others selecting them in "*a very random order*" (P8). Users generally did not generally want to resort to this strategy; for the AE, P9 started off by putting in significant effort "*trying to see if [they] could find any understanding of what the dials meant*" but "*it didn't work,*" and with reluctance, their "*strategy changed*" to "*finicking with the dials.*" However, P8 felt that for complicated models, the gradient ascent strategy was actually "*less mentally demanding*" because "*not even trying to figure out*" dimension meanings meant the task was sufficiently mindless that they could "*do something else at the same time,*" such as "*hold a conversation.*" Users felt this was inefficient, with P2 commenting that "*this is definitely not the fastest strategy,*" but when "*there wasn't any easy way to define anything [...] to do it systematically feels really annoying*" (P8). These comments may help to explain why the differences in subjective measures of effort were less significant than differences in frustration, understanding, and performance.

7 DISCUSSION

As shown in Figure 6 and Table 2, **interactive reconstruction metrics differentiated entangled and disentangled models, both absolutely and relative to baselines.** On both synthetic datasets, these metrics (specifically completion rate, difficulty rating, and error AUC) clearly distinguished the entangled AE from the disentangled GT model. Single-dimension task metrics, on the other hand, showed a much less clear relationship with disentanglement, sometimes predicting that the highly entangled autoencoder was more interpretable than ground truth. On MNIST, interactive reconstruction metrics continued to meaningfully differentiate models in a manner consistent with hypotheses from the disentangled representations literature (e.g. β-TCVAEs especially semi-superivsed β-TCVAEs were more interpretable than AEs and standard VAEs).

Interactive reconstruction metrics measured understanding. This is a major claim, but our results suggest that performing the task helped users understand models, and that understanding models helped users perform the task. First, we know that after performing interactive reconstruction, users felt they understood the ground-truth interpretable models much more than unregularized autoencoder models. On MNIST, where there was no GT model, users held similar feelings towards the SS vs. the AE. Although one could attribute these feelings to an illusion of explanatory

| Model	Dimension	User-Assigned Labels
AE$_5$	Continuous 1	"791"
	Continuous 2	"thickness"
	Continuous 3	"rotate"
	Continuous 4	"stretch"
	Continuous 5	—
TC$_5$	Continuous 1	"thickness", "0-1", "closedness", "2-0"
	Continuous 2	"shrinking horizontally", "paint white", "0-1"
	Continuous 3	**"diagonal"**, **"rotate"**, "4-5-3"
	Continuous 4	"add bottom left", **"3-5-6"**, **"3-5-blob"**
	Continuous 5	"ccw", **"6-7"**, **"7-6?"**
SS$_5$	Discrete 1	**"number"**, **"Digit"**, **"Class"**, **"digit"**
	Continuous 1	"up", "Widthish", "paint white", "rotate cw, change focus"
	Continuous 2	**"ccw"**, **"LR Skew"**, **"Rotation"**, "rotate", "skew rotate cw/ccw"
	Continuous 3	"curve", **"TB Skew"**, "up down bias", "focus up/dwn"
	Continuous 4	"thickness", **"Wide"**, **"width"**, "horiz thickness somewhat"
	Continuous 5	"left", **"Thick"**, **"paint black"**, "line thickness/focus but white"

Table 1: All labels assigned to $D_z = 5$ models by participants in the MNIST thinkaloud study. Bold text shows labels experimenters identified as consistent between participants. The semi-supervised (SS) model had both the most labels and the most consistent labels, while the autoencoder (AE) had the fewest. Labels for $D_z = 10$ are given in Table A.2 in the Appendix.

Model Information				Disentanglement		Interactive Reconstruction			
Dataset	Model	D_z	MSE	DCI [22]	MIG [17]	Completion Rate	Difficulty	Error AUC/10^3	Link
dSprites	AE	5	**6.5**	0.14	0.03	0.15±0.19	6.93±0.25	55.8±25.6	➡
dSprites	VAE	5	8.1	0.40	0.09	0.51±0.33	6.20±0.91	60.1±26.6	➡
dSprites	GT	5	8.9	**1.00**	**1.00**	**0.75±0.26**	**4.20±1.60**	**23.6±5.9**	➡
Sinelines	AE	5	0.6	0.21	0.03	0.34±0.26	5.87±1.67	163.5±90.0	➡
Sinelines	VAE	5	3.3	0.40	0.15	0.63±0.28	4.07±1.84	59.6±47.8	➡
Sinelines	GT	5	**0.0**	**1.00**	**1.00**	**0.83±0.19**	**4.00±1.71**	**52.0±48.9**	➡
MNIST	AE	5	**15.5**	—	—	0.38±0.23	5.60±1.62	15.7±4.3	➡
MNIST	VAE	5	16.2	—	—	0.40±0.31	5.87±1.15	18.6±13.7	➡
MNIST	IG	5	—	—	—	0.51±0.31	5.67±1.30	21.3±8.6	➡
MNIST	TC	5	25.4	—	—	0.62±0.26	5.00±1.41	16.0±13.1	➡
MNIST	SS	5	20.8	—	—	**0.73±0.24**	**3.40±1.54**	**6.3±4.7**	➡
MNIST	AE	10	**7.2**	—	—	0.35±0.29	5.27±1.73	47.8±34.0	➡
MNIST	TC	10	24.9	—	—	—	—	—	➡
MNIST	SS	10	20.7	—	—	**0.71±0.29**	**4.40±1.40**	**7.6±3.3**	➡

Table 2: MTurk interactive reconstruction metric means and standard deviations along with general metrics for each of the models used in all experiments. Mean squared error (MSE) measures autoencoders' errors reconstructing inputs not seen during training. DCI and MIG are measures of disentanglement applicable to synthetic datasets. Bold entries indicate least error / greatest interpretability or disentanglement in each group. Links go directly to tasks for each model (skipping instructions).

depth [58], that would not explain why user feelings were consistent for the same pairs of models or why users assigned similar labels to dimensions of models they felt they understood (Table 1). It also would not explain evidence of learning across trials, e.g. why performance increased most consistently for interpretable models (Figure 10). Overall, the evidence suggests that users achieved meaningfully different levels of model understanding by performing the interactive reconstruction (and not the single-dimension) task.

Second, we found that when users felt they understood the meanings of dimensions, they could modify them in a single pass through the list, knowing in advance in which direction and how much to change them. This allowed them to solve problems efficiently, leading directly to high completion rates and low error AUC. When users did not understand dimensions, we found that they tended to adopt less efficient gradient ascent or random experimentation
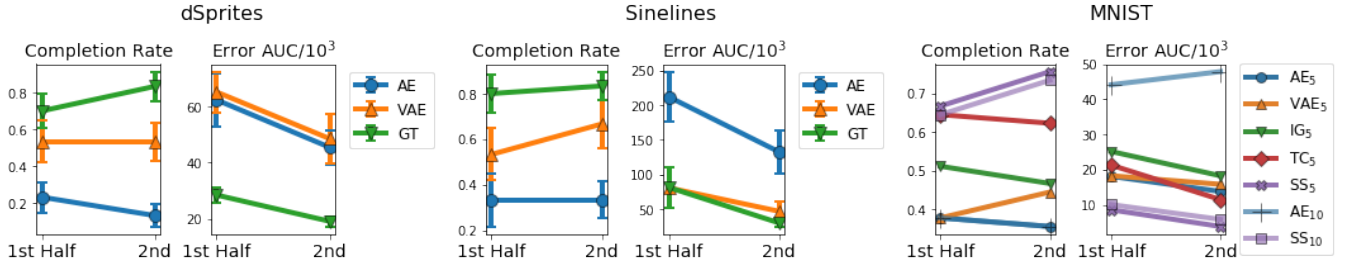
Figure 10: Changes in average MTurk interactive reconstruction metrics between the first and last $(N_q - 1)/2$ questions (error-bars show standard error and are omitted on MNIST for readability). GT/SS models show the most consistent improvements over questions (suggesting conscious effort / learning) while AEs sometimes show degradation (suggesting users give up).

procedures with numerous loops through the full set of dimensions (or a partial set, in the intermediate-understanding case).

A competing hypothesis could be that, rather than gaining understanding, users *always* used gradient ascent or experimented randomly, but just happened to more easily stumble upon solutions with the models we assumed were "interpretable" than the models we assumed were "uninterpretable." In certain cases, this effect may be partially operative. For example, on MNIST, AEs consistently have the lowest reconstruction error (shown in Table 2), implying they can reconstruct a wider variety of images and therefore have a larger "search space." Meanwhile, TCs consistently have the highest reconstruction error (25.4 vs. 15.5 at $D_z = 5$) and therefore the smallest search space, so randomly experimenting users ought to happen upon solutions more quickly.

However, this hypothesis is at odds with our qualitative observations, and also conflicts with our quantitative results in many cases. For example, on MNIST, the SS model has lower reconstruction error (20.8 at $D_z = 5$) and thus a larger search space than the TC model, but it performs much better. On Sinelines, the GT model has precisely zero reconstruction error but still performs best. Although having a smaller search space may be helpful (and arguably less expressive models may often be easier to understand), we posit that task performance depends more strongly on the understandability of a search space, rather than its size.

Generalizability. It is worth emphasizing that interactive reconstruction differentiated models in qualitatively similar ways across multiple datasets (dSprites, Sinelines, and MNIST) and user groups (workers on Amazon Mechanical Turk and students in Computer Science). This consistency suggests that interactive reconstruction can be useful as a research tool for comparing interpretable representation learning methods in varied contexts. Some innovation (beyond choosing appropriate parameters per Section 3.1) may still be required to adapt the method to non-visual or discontinuous data modalities (e.g. audio, text, or medical records) or to specific user groups in application-grounded contexts (per Doshi-Velez and Kim [21]). However, while it is desirable for interpretable representation learning algorithms to generalize to all contexts, interpretability *measurement* tools can afford to be a little more domain-specific, as long as they can still identify the algorithms that output the most interpretable models across contexts.

Limitations. Interactive reconstruction has limitations not shared by the single-dimension task which our experiments do not fully explore. First, these tasks require interactive visualization, which may be technically challenging for practitioners to implement, though the increasing usability of efficient web-based machine learning frameworks [64] helps. Second, interacting with all dimensions simultaneously may be overwhelming for models with many tens or hundreds of representation dimensions. Potential workarounds include allowing users to annotate, as with MNIST, and/or group dimensions, or defining tasks over subgroups rather than the full set. However, we note that 100D representations may be inherently uninterpretable due to limits on working memory [66]—unless the models are structured so that only sparse subsets of dimensions need to be considered simultaneously.

Our task also requires setting several parameter choices sensibly. For example, despite our initial pilots on MNIST, 8/10 think-aloud study participants made at least one comment that our distance metric, the intersection-over-union alignment percentage, did not match their intuitive notion of perceptual similarity, with P3 commenting that sometimes the images "*look similar but the alignment [percentage] doesn't reflect that.*" Although it was uncommon for users to reach the threshold alignment ϵ without feeling that the images were similar, they felt frustrated that changes in $d(x, x')$ seemed unrelated to progress early on. Although we feel confident our method will generalize to many datasets and data types, our experience suggests practitioners will need to take care when selecting metrics $d(x, x')$ and thresholds ϵ. Future work could explore choosing example-specific thresholds or, on visual data, using metrics explicitly designed to model perceptual similarity [19, 67, 69] (if they can be evaluated efficiently in-browser).

Finally, interactive reconstruction is specific to generative models, e.g. autoencoders and GANs. Single-dimension tasks, however, can be made to support the other main category of representations, embeddings, via feature visualization [54]. Evaluating embedding interpretability is an area for future work. Cavallo et al. [15], Smilkov et al. [65], and Arendt et al. [4] may provide a foundation.

8 CONCLUSION

Developing reliable methods of evaluating interpretability is important for progress in interpretable ML. In this study, we introduced an interactive reconstruction task for evaluating the interpretability of generative models, which have largely gone unstudied in

the growing literature on human factors in ML. We validated our method by verifying it was effective at identifying ground-truth differences in model interpretability—both absolutely and relative to baselines—and that differences in objective performance metrics corresponded to meaningful differences in subjective understanding, which was measured in multiple independent ways. We then applied it to a wide range of representation learning methods from the disentanglement literature, and found evidence that methods which have been shown to improve disentanglement on synthetic data, e.g., [17], also improve interpretability on real data. To our awareness, ours is the first study providing such evidence.

ACKNOWLEDGMENTS

REFERENCES

[1] Ashraf Abdul, Christian von der Weth, Mohan Kankanhalli, and Brian Y Lim. 2020. COGAM: Measuring and Moderating Cognitive Load in Machine Learning Model Explanations. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–14.

[2] Hiva Allahyari and Niklas Lavesson. 2011. User-oriented assessment of classification model understandability. In *11th scandinavian conference on Artificial intelligence*. IOS Press.

[3] David Alvarez-Melis and Tommi S Jaakkola. 2018. Towards robust interpretability with self-explaining neural networks. *arXiv preprint arXiv:1806.07538* (2018).

[4] Dustin L Arendt, Nasheen Nur, Zhuanyi Huang, Gabriel Fair, and Wenwen Dou. 2020. Parallel embeddings: a visualization technique for contrasting learned representations. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*. 259–274.

[5] Gagan Bansal, Besmira Nushi, Ece Kamar, Walter S Lasecki, Daniel S Weld, and Eric Horvitz. 2019. Beyond accuracy: The role of mental models in human-AI team performance. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*. 2–11.

[6] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2017. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 6541–6549.

[7] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B Tenenbaum, William T Freeman, and Antonio Torralba. 2019. Visualizing and Understanding GANs. (2019).

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

[9] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[10] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.

[11] Zana Buçinca, Phoebe Lin, Krzysztof Z Gajos, and Elena L Glassman. 2020. Proxy tasks and subjective measures can be misleading in evaluating explainable AI systems. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*. 454–464.

[12] Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. 2018. Understanding disentangling in beta-VAE. *arXiv preprint arXiv:1804.03599* (2018).

[13] Carrie J Cai, Emily Reif, Narayan Hegde, Jason Hipp, Been Kim, Daniel Smilkov, Martin Wattenberg, Fernanda Viegas, Greg S Corrado, Martin C Stumpe, et al. 2019. Human-centered tools for coping with imperfect algorithms during medical decision-making. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–14.

[14] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. 2015. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 1721–1730.

[15] Marco Cavallo and Çağatay Demiralp. 2018. A visual interaction framework for dimensionality reduction based data exploration. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–13.

[16] Paul Chandler and John Sweller. 1992. The split-attention effect as a factor in the design of instruction. *British Journal of Educational Psychology* 62, 2 (1992),

233–246.

[17] Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. 2018. Isolating sources of disentanglement in variational autoencoders. In *Advances in neural information processing systems*. 2610–2620.

[18] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*. 2172–2180.

[19] Steffen Czolbe, Oswin Krause, Ingemar Cox, and Christian Igel. 2020. A Loss Function for Generative Neural Networks Based on Watson's Perceptual Model. *Advances in Neural Information Processing Systems* 33 (2020).

[20] Guillaume Desjardins, Aaron Courville, and Yoshua Bengio. 2012. Disentangling factors of variation via generative entangling. *arXiv preprint arXiv:1210.5474* (2012).

[21] Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608* (2017).

[22] Cian Eastwood and Christopher KI Williams. 2018. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*.

[23] Shi Feng and Jordan Boyd-Graber. 2019. What Can AI Do for Me? Evaluating Machine Learning Interpretations in Cooperative Play. In *Proceedings of the 24th International Conference on Intelligent User Interfaces (IUI '19)*. Association for Computing Machinery, New York, NY, USA, 229–239. https://doi.org/10.1145/3301275.3302265

[24] Zoubin Ghahramani and Michael I Jordan. 1996. Factorial hidden Markov models. In *Advances in Neural Information Processing Systems*. 472–478.

[25] Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. 2018. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*. IEEE, 80–89.

[26] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

[27] David Ha and Jürgen Schmidhuber. 2018. Recurrent World Models Facilitate Policy Evolution. In *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc., 2451–2463. https://papers.nips.cc/paper/7512-recurrent-world-models-facilitate-policy-evolution https://worldmodels.github.io.

[28] MM Hansen, T Miron-Shatz, AYS Lau, and C Paton. 2014. Big data in science and healthcare: a review of recent literature and perspectives. *Yearbook of medical informatics* 23, 01 (2014), 21–26.

[29] Sandra G Hart and Lowell E Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in psychology*. Vol. 52. Elsevier, 139–183.

[30] Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. 2018. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230* (2018).

[31] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, Vol. 3.

[32] Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *science* 313, 5786 (2006), 504–507.

[33] Johan Huysmans, Karel Dejaeger, Christophe Mues, Jan Vanthienen, and Bart Baesens. 2011. An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decision Support Systems* 51, 1 (2011), 141–154.

[34] Ian T Jolliffe. 1986. Principal components in regression analysis. In *Principal component analysis*. Springer, 129–155.

[35] Harmanpreet Kaur, Harsha Nori, Samuel Jenkins, Rich Caruana, Hanna Wallach, and Jennifer Wortman Vaughan. 2020. Interpreting Interpretability: Understanding Data Scientists' Use of Interpretability Tools for Machine Learning. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376219

[36] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. 2018. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*. PMLR, 2668–2677.

[37] Hyunjik Kim and Andriy Mnih. 2018. Disentangling by Factorising. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Jennifer Dy and Andreas Krause (Eds.), Vol. 80. PMLR, Stockholmsmässan, Stockholm Sweden, 2649–2658. http://proceedings.mlr.press/v80/kim18b.html

[38] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[39] Josua Krause, Adam Perer, and Kenney Ng. 2016. Interacting with predictions: Visual inspection of black-box machine learning models. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 5686–5697.

[40] Josua Krause, Adam Perer, and Kenney Ng. 2016. Interacting with Predictions: Visual Inspection of Black-Box Machine Learning Models. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. Association for Computing Machinery, New York, NY, USA, 5686–5697. https://doi.org/10.1145/2858036.2858529

[41] Todd Kulesza, Simone Stumpf, Margaret Burnett, Sherry Yang, Irwin Kwan, and Weng-Keen Wong. 2013. Too much, too little, or just right? Ways explanations impact end users' mental models. In *2013 IEEE Symposium on Visual Languages and Human Centric Computing*. IEEE, 3–10.

[42] Isaac Lage, Emily Chen, Jeffrey He, Menaka Narayanan, Been Kim, Sam Gershman, and Finale Doshi-Velez. 2019. An evaluation of the human-interpretability of explanation. *arXiv preprint arXiv:1902.00006* (2019).

[43] Isaac Lage, Andrew Ross, Samuel J Gershman, Been Kim, and Finale Doshi-Velez. 2018. Human-in-the-loop interpretability prior. In *Advances in neural information processing systems*. 10159–10168.

[44] Yann LeCun. 1998. The MNIST database of handwritten digits. *http://yann. lecun. com/exdb/mnist/* (1998).

[45] Michael Levandowsky and David Winter. 1971. Distance between sets. *Nature* 234, 5323 (1971), 34–35.

[46] Clayton Lewis. 1982. *Using the" thinking-aloud" method in cognitive interface design*. IBM TJ Watson Research Center Yorktown Heights, NY.

[47] Brian Y Lim, Anind K Dey, and Daniel Avrahami. 2009. Why and why not explanations improve the intelligibility of context-aware intelligent systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2119–2128.

[48] Zachary C Lipton. 2018. The mythos of model interpretability. *Queue* 16, 3 (2018), 31–57.

[49] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. 2019. Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.), Vol. 97. PMLR, Long Beach, California, USA, 4114–4124. http://proceedings.mlr.press/v97/locatello19a.html

[50] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[51] Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. 2017. dSprites: Disentanglement testing Sprites dataset. https://github.com/deepmind/dsprites-dataset.

[52] Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence* 267 (2019), 1–38.

[53] Don Norman. 2013. *The design of everyday things: Revised and expanded edition*. Basic books.

[54] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. 2017. Feature visualization. *Distill* 2, 11 (2017), e7.

[55] Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. 2018. The building blocks of interpretability. *Distill* 3, 3 (2018), e10.

[56] Forough Poursabzi-Sangdeh, Daniel G Goldstein, Jake M Hofman, Jennifer Wortman Vaughan, and Hanna Wallach. 2018. Manipulating and measuring model interpretability. *arXiv preprint arXiv:1802.07810* (2018).

[57] Karl Ridgeway. 2016. A survey of inductive biases for factorial representation-learning. *arXiv preprint arXiv:1612.05299* (2016).

[58] Leonid Rozenblit and Frank Keil. 2002. The misunderstood limits of folk science: An illusion of explanatory depth. *Cognitive science* 26, 5 (2002), 521–562.

[59] Jeff Sauro and Joseph S Dumas. 2009. Comparison of three one-question, post-task usability questionnaires. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 1599–1608.

[60] Morgan Klaus Scheuerman, Katta Spiel, Oliver L Haimson, Foad Hamidi, and Stacy M Branham. 2019. HCI guidelines for gender equity and inclusivity. https://www.morgan-klaus.com/gender-guidelines.html.

[61] Ute Schmid, Christina Zeller, Tarek Besold, Alireza Tamaddoni-Nezhad, and Stephen Muggleton. 2016. How does predicate invention affect human comprehensibility?. In *International Conference on Inductive Logic Programming*. Springer, 52–67.

[62] Anna Sepliarskaia, Julia Kiseleva, and Maarten de Rijke. 2019. Evaluating Disentangled Representations. *arXiv preprint arXiv:1910.05587* (2019).

[63] Dylan Slack, Sorelle A Friedler, Carlos Scheidegger, and Chitradeep Dutta Roy. 2019. Assessing the Local Interpretability of Machine Learning Models. *arXiv preprint arXiv:1902.03501* (2019).

[64] Daniel Smilkov, Nikhil Thorat, Yannick Assogba, Ann Yuan, Nick Kreeger, Ping Yu, Kangyi Zhang, Shanqing Cai, Eric Nielsen, David Soergel, et al. 2019. Tensorflow.js: Machine learning for the web and beyond. *arXiv preprint arXiv:1901.05350* (2019).

[65] Daniel Smilkov, Nikhil Thorat, Charles Nicholson, Emily Reif, Fernanda B Viégas, and Martin Wattenberg. 2016. Embedding projector: Interactive visualization and interpretation of embeddings. *arXiv preprint arXiv:1611.05469* (2016).

[66] John Sweller. 1994. Cognitive load theory, learning difficulty, and instructional design. *Learning and instruction* 4, 4 (1994), 295–312.

[67] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 13, 4 (2004), 600–612.

[68] James Wexler, Mahima Pushkarna, Tolga Bolukbasi, Martin Wattenberg, Fernanda Viégas, and Jimbo Wilson. 2019. The what-if tool: Interactive probing of machine learning models. *IEEE transactions on visualization and computer graphics* 26, 1 (2019), 56–65.

[69] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 586–595.