# A. Optimization Algorithm

**Algorithm 1** Alternating optimization algorithm

Initialize $\theta = \theta_0$
**repeat**
  $x_1, \ldots, x_n \sim \mathcal{S}$ {Sample $n$ train examples}
  $z_i \leftarrow \phi_\theta(x_i) \ \forall i \in [n]$ {Generate representations}
  $a_i \leftarrow h(\rho(z_i)) \ \forall i \in [n]$ {Query human decisions}
  $\mathcal{T} = \{(z_i, a_i)\}_{i=1}^n$
  $\eta \leftarrow \text{argmin}_{\eta'} \mathbb{E}_{\mathcal{T}}[\ell(a, \hat{h}_{\eta'}(z))]$ {Train $\hat{h}$}
  $\theta \leftarrow \text{argmin}_{\theta'} \mathbb{E}_{\mathcal{S}}[\ell(y, \hat{h}_\eta(\phi_{\theta'}(x)))]$ {Train $\phi$}
**until** convergence

# B. General Optimization Issues

## B.1. Initialization

Because acquiring human labels is expensive, it is important to initialize $\phi$ to map to a region of the representation space in which there is variation and consistency in human reports, such that gradients lead to progress in subsequent rounds.

In some representation spaces, such as our 2D projections of noisy 3D rotated images, this is likely to be the case (almost any 3D slice will retain some signal from the original 2D image). However, in 4+ dimensions, as well as with the subset selection and avatar tasks, there are no such guarantees.

To minimize non-informative queries, we adopt two initialization strategies:

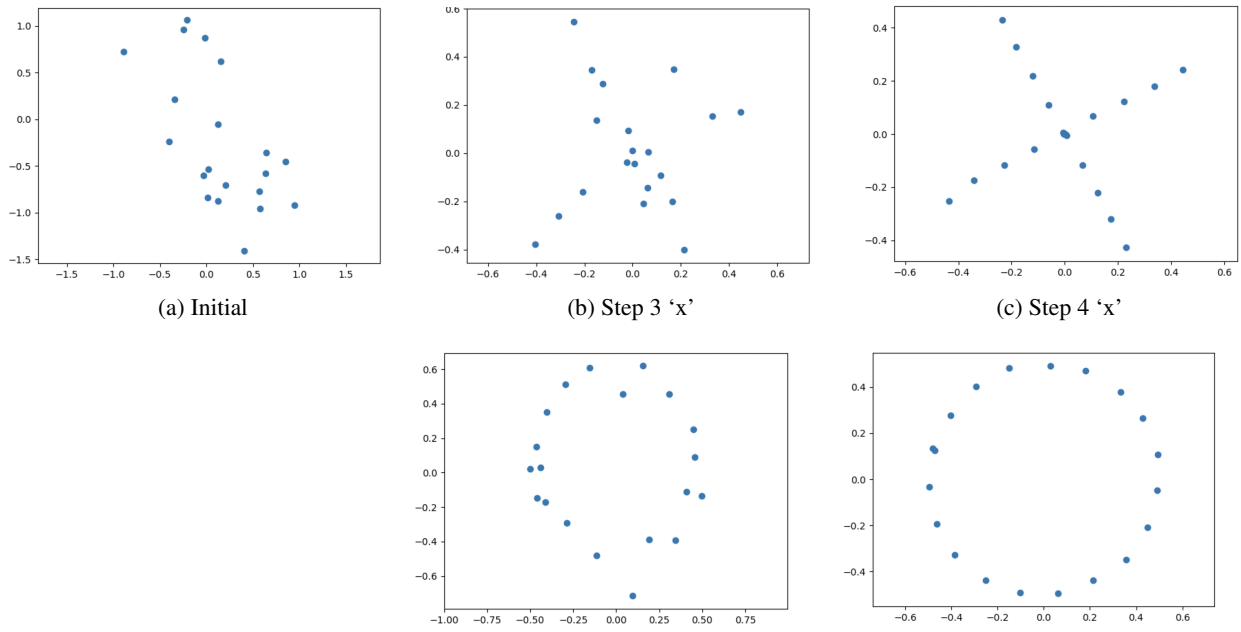
1. **Initialization with a computer-only model:** In scenarios in which the representation space is a (possibly discrete) subset of input space, such as in subset selection, the initialization problem is to isolate the region of the input space that is important for decision-making. In this situation, it can be useful to initialize with a computer-only classifier. This classifier should share a representation-learning architecture with $\phi$ but can have any other classifying architecture appended (although simpler is likely better for this purpose). This should result in some $\phi$ which at least focuses on the features relevant for classification, if not necessarily in a human-interpretable format.
2. **Initialization to a desired distribution with a WGAN:** In scenarios in which the initialization problem is to isolate a region of representation space into which to map all inputs, as in the avatar example, in which we wish to test a variety of expressions without creating expression combinations which will appear overly strange to participants, it can be useful to hand-design a starting distribution over representation space and initialize $\phi$ with a Wasserstein GAN (Arjovsky et al., 2017). In this case, we use a Generator Network with the same architecture as $\phi$ but allow the Discriminator Network to be of any effective architecture. As with the previous example, this results in an $\phi$ in which the desired distribution is presented to users, but not necessarily in a way that reflects any human intuitive concept.

## B.2. Convergence

As is true in general of gradient descent algorithms, the M∘M framework is not guaranteed to find a global optimum but rather is likely to end up at a local optimum dependent on both the initialization of $\phi$ and $\hat{h}$. In our case, however, the path of gradient descent is also dependent on the inherently stochastic selection and behavior of human users. If users are inconsistent or user groups at different iterations are not drawn from the same behavior distribution, it is possible that learning at one step of the algorithm could result in convergence to a suboptimal distribution for future users. It remains for future work to test how robust machine learning methods might be adapted to this situation to mitigate this issue.

## B.3. Regularization/Early Stopping

As mentioned in Section 3, training $\phi$ will in general shift the distribution of the representation space away from the region on which we have collected labels for $\hat{h}$ in the previous iterations, resulting in increasing uncertainty in the predicted outcomes. We test a variety of methods to account for this, but developing a consistent scheme for choosing how best to maximize the information in human labels remains future work.



(a) Initial (b) Step 3 'x' (c) Step 4 'x'

(d) Step 3 'o' (e) Step 4 'o'

*Figure 5.* Images of x-o interface

- **Regularization of $\hat{h}$:** We test regularization of $\hat{h}$ both with Dropout and L2 regularization, both of which help in preventing overfitting, especially in early stages of training, when the representation distribution is not yet refined. As training progresses and the distribution $\phi_\theta(x)$ becomes more tightly defined, decreasing these regularization parameters increases performance.

- **Training $\hat{h}$ with samples from previous iterations**: We also found it helpful in early training iterations to reuse samples from the previous human labeling round in training $\hat{h}$, as inspired by [Bobu et al. 2018]. [12] We weight these samples equally and use only the previous round, but it may be reasonable in other applications to alter the weighting scheme and number of rounds used.

- **Early stopping based on Bayesian Linear Regression:** In an attempt to quantify how the prediction uncertainly changes as $\theta$ changes, we also implement Bayesian Linear Regression, found in [Riquelme et al., 2018] [13] to be a simple but effective measure of uncertainty, over the last layer of $\hat{h}(\phi_\theta)$ as we vary $\theta$ through training. We find that in early iterations of training, this can be an effective stopping criterion for training of $\phi$. Again, as training progresses, we find that this mostly indicates only small changes in model uncertainty.

### B.4. Human Input

Testing on mTurk presents various challenges for testing the M○M framework:

- In some applications, such as loan approval, mTurk users are not experts. This makes it difficult to convince them that anything is at stake (we found that bonuses did not meaningfully affect performance). It is also difficult to directly measure effort, agency, trust, or autonomy, all of which result in higher variance in responses.

- In many other applications, the ground truth is generated by humans to begin with (for example, sentiment analysis). Since we require ground truth for training, in these task it cannot be expected of humans to outperform machines.

[12] Bobu, Andreea, et al. "Adapting to continuously shifting domains." (2018).

[13] Riquelme, Carlos, George Tucker, and Jasper Snoek. "Deep bayesian bandits showdown." *International Conference on Learning Representations*. 2018.

- As the researchers found in (Lage et al., 2018), there can be a large variance in the time users take to complete a given task. Researchers have found that around 25% of mTurk users complete several tasks at once or take breaks during HITs [Moss and Litman, 2019].[14] making it difficult to determine how closely Turkers are paying attention to a given task. We use requirements of HIT approval rate greater than 98%, US only, and at least 5,000 HITs approved, as well as a simple comprehension check.

- Turker populations can vary over time and within time periods, again leading to highly variable responses, which can considerably effect the performance of learning.

- Recently, there have been concerns regarding the usage of automated bots within the mTurk communiy. Towards this end, we incorporated in the experimental survey a required reading comprehension task and as well as a CAPTCHA task, and filtered users that did not succeed in these.

## C. Experimental Details

### C.1. Decision-compatible 2D projections

In the experiment, we generate 1,000 examples of these point clouds in 3D. The class of $\phi$ is a 3x3 linear layer with no bias, where we add a penalization term on $\phi^T\phi - \mathbb{I}$ during training to constrain the matrix to be orthogonal. Humans are shown the result of passing the points through this layer and projecting onto the first two dimensions. The class of $\hat{h}$ is a small network with 1 3x3 convolutional layer creating 3 channels, 2x2 max pooling, and a sigmoid over a final linear layer. The input to this network is a soft (differentiable) 6x6 histogram over the 2D projection shown to the human user.

We tested an interactive command line query and response game on 12 computer science students recruited on Slack and email. Users filled out a consent form online, watched an instructional video, and then completed a training and testing round, each with up to 5 rounds of 15 responses. Due to the nature of the training process, achieving 100% accuracy results in $\phi$ not updating in the following round. With this in mind, if a user reached 100% accuracy in training, they immediately progressed to testing. If a user reached 100% accuracy in testing, the program exited. $\phi$ was able to find a representation that allowed for 100% accuracy 75% of the time, with an average 5 round improvement of 23% across all participants. Many times the resulting projection appeared to be an 'x' and 'o', as in Figure 5, but occasionally it was user-specific. For example, a user who associates straight lines with the 'x' may train the network to learn any projection for 'x' that includes many points along a straight line.

The architecture of $\phi$ and $\hat{h}$ are described in Section 4. For training, we use a fixed number of epochs (500 for $\hat{h}$ and 300 for $\phi$) with base learning rates of .07 and .03, respectively, that increase with lower accuracy scores and decrease with each iteration. We have found these parameters to work well in practice, but observed that results were not sensitive to their selection. The interface allows the number of rounds and examples to be determined by the user, but often 100% accuracy can be achieved after about 5 rounds of 15 examples each.

### C.2. Decision-compatible algorithmic avatars

#### C.2.1. Data Preprocessing.

We use the *Lending Club* dataset, which we filter to include only loans for which we know the resolution (either default or paid in full, not loans currently in progress) and to remove all features that would not have been available at funding time. We additionally drop loans that were paid off in a single lump sum payment of at least 5 times the normal installment. This results in a dataset that is 49% defaulted and 51% repaid loans. Categorical features are transformed to one-hot variables. There are roughly 95,000 examples remaining in this dataset, of which we split 20% into the test set.

#### C.2.2. Learning architecture and pipeline.

The network $\phi$ takes as input the standardized loan data. Although the number of output dimension are $\mathbb{R}^9$, $\phi$ outputs vectors in $\mathbb{R}^{11}$. This is because the some facial expressions do not naturally coexist as compound emotions, i.e., happiness and sadness [Du et al., 2014]. [15] Hence, we must add some additional constraints to the output space, encoded in the extra

[14] A. J. Moss and L. Litman. How do most mturk workers work?, Mar 2019.

[15] Shichuan Du, Yong Tao, and Aleix M Martinez. Compound facial expressions of emotion. *Proceedings of the National Academy of Sciences*, 111(15):E1454–E1462, 2014.

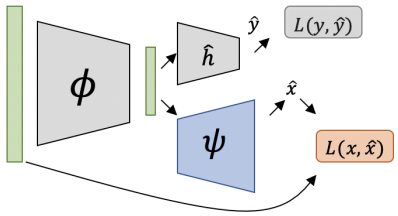
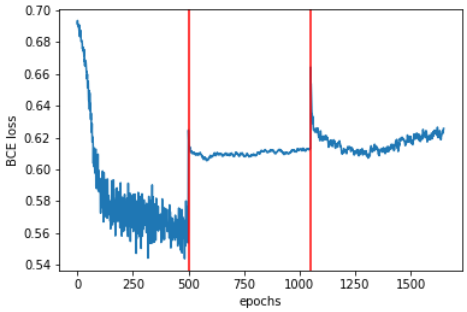
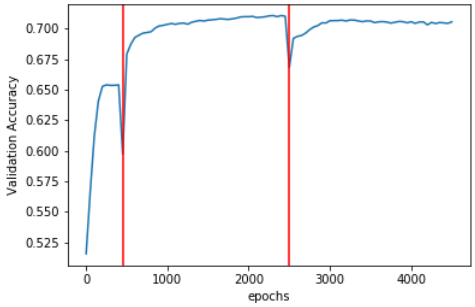



*Figure 6.* Visualization of reconstruction component




(a) Loss in training $\hat{h}$ over 3 rounds




(b) Validation Accuracy in training $\phi$ over 3 rounds

*Figure 7.* $\hat{h}$ does not necessarily have to match $h$ well to lead to an increase in accuracy

dimensions. For example, happiness and sadness are split into two separate parameters (rather than using one dimension with positive for happiness and negative for sadness). The same is true of "happy surprise", which is only allowed to coincide with happiness, as opposed to "sad surprise." For parameters which have positive and negative versions, we use a tanh function as the final nonlinearity, and for parameters which are positive only, we use a sigmoid function as the final nonlinearity.

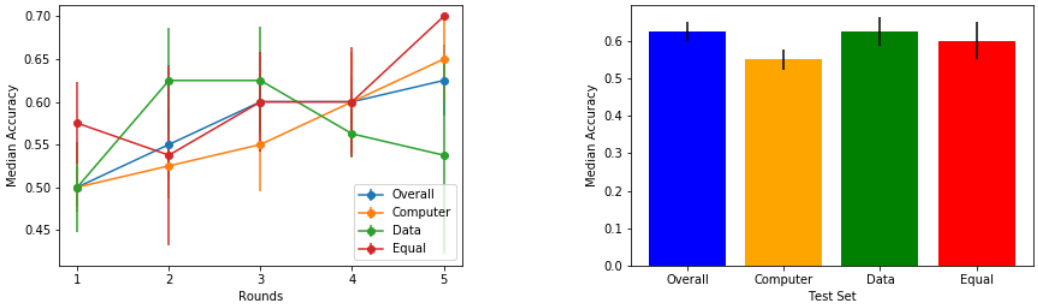
These parameters are programmatically mapped to a series of Webmorph (DeBruine & Tiddeman, 2016) transformation text files, which are manually loaded into the batch transform/batch edit functions of Webmorph. We use base emotion images from the CFEE database [Du et al., 2014] and trait identities from [Oosterhof and Todorov, 2008].[16] This forms $\rho$ for this experiment.

The network $\phi$ is initialized with a WGAN to match a distribution of parameters chosen to output a fairly uniform distribution of feasible faces. To achieve this, each parameter was chosen to be distributed according to one of the following: a clipped $\mathcal{N}(0, 4), \mathcal{U}[0, 1]$, or Beta(1,2). The choice of distribution was based on inspection as to what would give reasonable coverage over the set of emotional representations we were interested in testing. In this initial version of $\phi$, $x$ values end up mapped randomly to representations, as the WGAN has no objective other than distribution matching.

The hidden layer sizes of $\phi$ and $\hat{h}$ were chosen via cross validation. For $\phi$, we use the smallest architecture out of those tested capable of recreating a wide distribution of representations $z$ as the generator of the WGAN. For $\hat{h}$, we use the smallest architecture out of those tested that achieves low error both in the computer-only simulation and with the first round of human responses.

In the first experiment, we collect approximately 5 labels each (with minor variation due to a few mTurk users dropping out mid-experiment) for the LASSO feature subset of 400 training set $x$ points and their $\phi_0$ mappings (see Figure 9). $a$ is taken to be the percentage of users responding "approve" for each point.

[16] Nikolaas N Oosterhof and Alexander Todorov. The functional basis of face evaluation. *Proceedings of the National Academy of Sciences*, 105(32):11087–11092, 2008.




(a) Training Rounds ('Overall' here is average *per user* score, rather than the score of the average response per question)

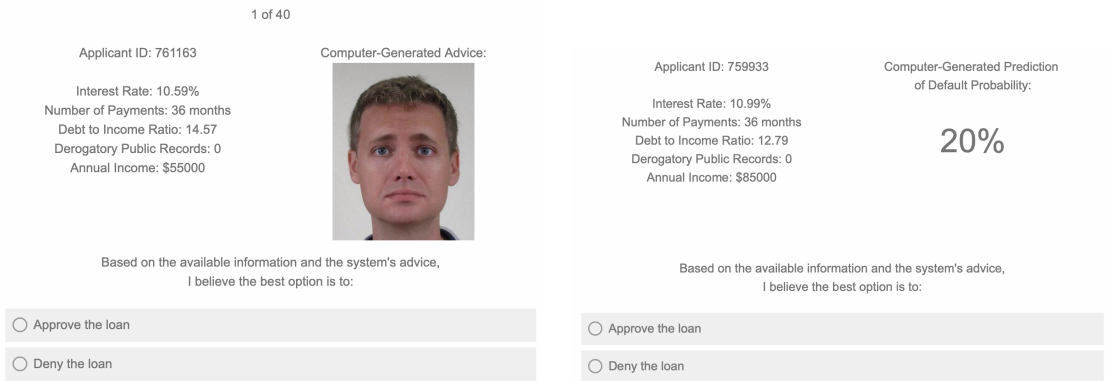
(b) Test Round

*Figure 8.* Results by Reported User Type

To train $\hat{h}$, we generate 15 different training-test splits of the collected $\{z, a\}$ pairs and compare the performance of variations of $\hat{h}$ in which it is either initialized randomly or with the $\hat{h}$ from the previous iteration, trained with or without adding the samples from the previous iteration, and ranging over different regularization parameters. We choose the training parameters and number of training epochs which result in the lowest average error across the 15 random splits. In the case of random initialization, we choose the best out of 30 random seeds over the 15 splits.

To train $\phi$, we fix $\hat{h}$ and use batches of 30,000 samples per epoch from the training set, which has 75,933 examples in total. To prevent mode collapse, wherein faces "binarize" to two prototypical exemplars, we add a reconstruction regularization term $R(x) = \|x - \psi(\phi(x))\|_2^2$ to the binary cross entropy accuracy loss, where $\psi$ is a decoder implemented by an additional neural network (see Figure 6). $\phi$ here also features a constraint penalty that prevents co-occurrence of incompatible emotions.

We train $\phi$ for 2,000 epochs with the Adam optimizer for a variety of values of $\alpha$, where we use $\alpha$ to balance reconstruction and accuracy loss in the form $\mathcal{L}_{total} = \alpha\mathcal{L}_{acc} + (1-\alpha)\mathcal{L}_{rec}$. We choose the value of $\alpha$ per round that optimally retains $x$ information while promoting accuracy by inspecting the accuracy vs. reconstruction MSE curve. We then perform Bayesian Linear Regression over the final layer of the current $\hat{h}$ for every 50th epoch of $\phi$ training and select the number of epochs to use by the minimum of either 2,000 epochs or the epoch at which accuracy uncertainty has doubled. In all but the first step, this resulted in using 2,000 epochs. At each of the 2-5th epochs, we choose only 200 training points to query. In the 6th epoch we use 200 points from the test set.




(a) (b)

*Figure 9.* Images from mTurk questionnaire

### C.2.3. SELF-REPORTED USER TYPE.

In the end of the survey, we ask users to report their decision method from among the following choices:

- I primarily relied on the data available
- I used the available data unless I had a strong feeling about the advice of the computer system
- I used both the available data and the advice of the computer system equally
- I used the advice of the computer system unless I had a strong feeling about the available data
- I primarily relied on the advice of the computer system
- Other

The percentage of users in each of these groups varied widely from round to round.

We consider the first two conditions to be the 'Data' group, the third to be the 'Equal' group, and the next two to be the 'Computer Advice' group. Although the trend is not statistically significant (at $p = 0.05$), likely due to the small number of subjects per type per round, we find it interesting that the performance improved on average over training rounds for all three types, of which the equal-consideration type performed best. For the data-inclined users, whose performance improved to surpass that of the no-advice condition in as early as round two, this implies at least one of the following: users misreport their decision method; users believe they are not influenced by the advice but are in fact influenced; or, as the algorithmic evidence becomes apparently better, only the population of users who are comparatively skilled at using the data continue to do so.

### C.2.4. DIVERSITY IN AVATAR REPRESENTATION.

Figure 10 presents examples of visualized avatars. Avatars correspond to examples having either low or high human-predicted probability (averaged across users) (top figure), and either low or high machine-predicted probability (lower figure). For visualization purposes, avatars are aligned according to a uni-dimensional PCA projection of the inputs, so that their spatial positioning captures the variance in the data. As can be seen, avatars are different for each predictive category (positive or negative; human or machine), but also vary considerably within each predictive category, with variance eminent across multiple facial dimensions.

We believe the additional dimensionality of the avatar representation relative to a numerical or binary prediction of default is useful for two reasons. Most importantly, high dimensionality allows users to retain an ability to reason about their decisions. In particular, avatars are useful because people likely have shared, mental reference points for faces. Moreover, users with a more sophisticated mental reference space may be able to teach the advising system over time to match specific reasoning patterns to specific characteristics. Additionally, when the advising system does not have a strong conviction about a prediction, presenting neutral advice should encourage the user to revisit the data, whereas percentages above or below the base rate of default (or 50%) may suffer from the anchoring effect.

### C.2.5. FURTHER DETAILS ON INFORMATION LEARNED BY $z$.

Using cross-validated ridge regression to predict individual $x$ variables from individual $z$ variables results in the coefficients of determination $R^2$ (to 2 significant figures) shown in Table 2.

Using cross-validated ridge regression to predict individual $x$ variables from all $z$ variables (both standardized to mean 0, std 1) results in the *variable coefficients* (to 2 significant figures) shown in Table 3.

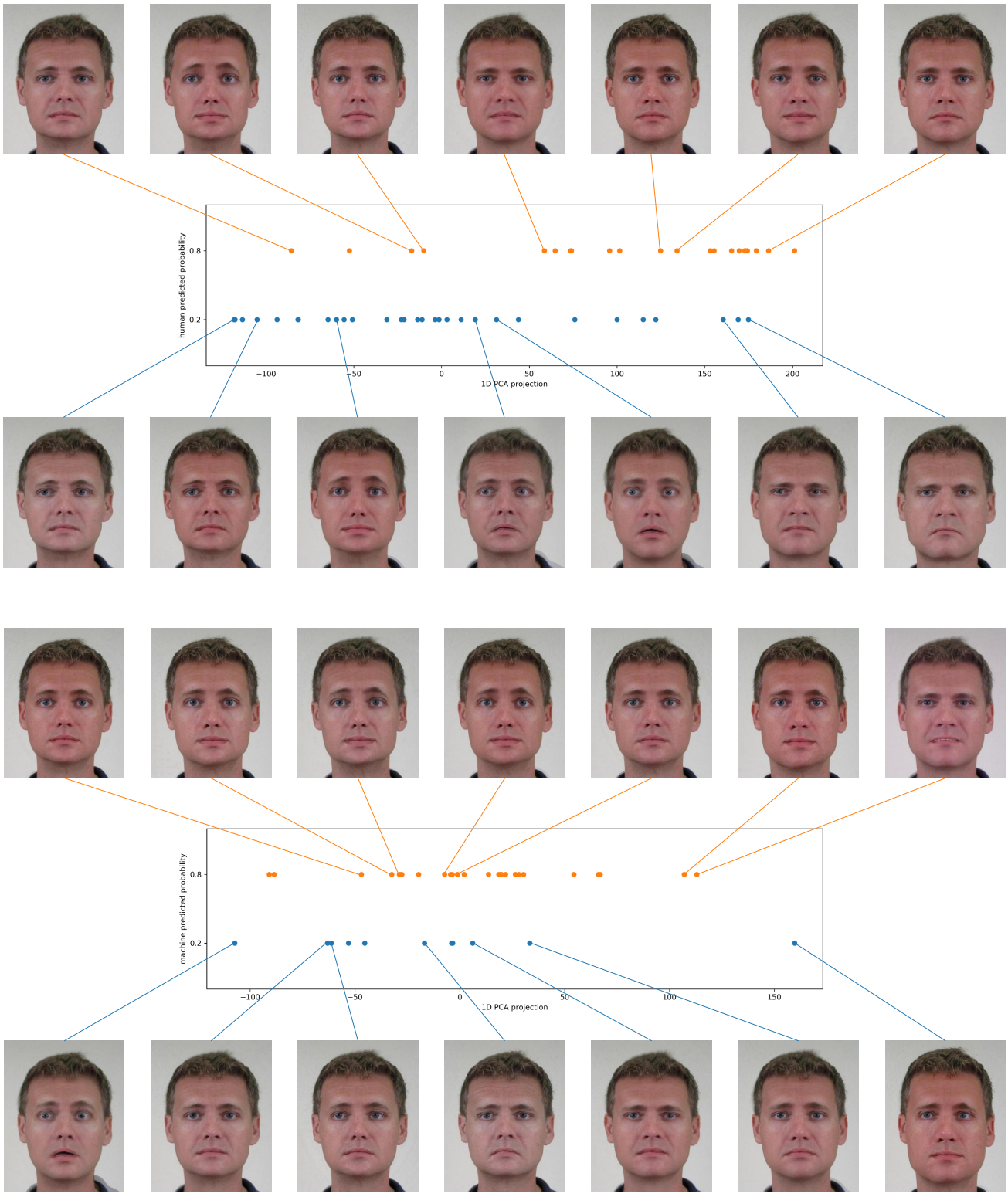



*Figure 10.* Richness of avatar representation. A visualization of 200 avatars randomly sampled from the held-out test set, grouped by either human (top) or machine (bottom) predictive probability (0.2 in blue, 0.8 in orange, with a tolerance of 0.05). Avatars are positioned based on a 1D PCA dimensionality reduction of their corresponding feature vectors $z$, along which a 'gradient' of facial changes can be observed. **Top**: Here avatars are grouped by human predictive probability. The figure shows how for the same human decisions, learning results in avatars of varied and complex facial expressions, conveying rich high-dimensional information. Interestingly, avatars corresponding to loan denial exhibit more variance, suggesting that there may be more 'reasons' for denying a loan than for approving one. **Bottom**: Here avatars are grouped by machine predictive probability. Since all examples in each group have the same predictive probability, they are equally similar, which does not facilitate a clear notion for reasoning. In contrast, avatars maintain richness in variation, and can be efficiently used for reasoning (e.g., via similarity arguments) and other downstream tasks.

## C.3. Incorporating Side Information

### C.3.1. Data Generation.

A directed graph showing the variable correlations is shown in Figure 11. The data in the side-information experiment is generated as follows: A latent variable $l_0 \sim \mathcal{N}(.3, .1)$ introduces a low correlation between $x_i$ and $x_r$ by setting a common mean for their Bernoulli probabilities $l_1, l_2$:

- $l_1, l_2 \sim \text{Unif}(\max(l_0 - .3, 0), \min(l_0 + .3, 1))$
- $x_i \sim \text{Bernoulli}(1 - l_1)$
- $x_r \sim \text{Bernoulli}(1 - l_2)$

An additional latent variable $l_3$ provides a similar correlation between $x_c$ and $x_d$, which also correlate, respectively, with $x_i$ and $x_r$:

- $l_3 \sim \text{Unif}(.5, .7)$
- $x_c \sim \text{Bernoulli}(l_3 + x_i)$
- $x_d \sim \text{Bernoulli}(l_3 + x_r)$

Side information $s$ is highly correlated with $x_r$ and $x_i$ but noisy: $s$ is drawn from a normal distribution centered at $x_r + x_i$ before rounding to an integer value between 0 and 3.

- $s_{cont} \sim \mathcal{N}(x_r + x_i, .5)$
- $s = \max(0, \min(3, \text{round}(s_{cont})))$

The integer outcome variable $y$ is the sum of $x_c$, $x_d$, and $s$. The binary outcome variable $y_{bin}$ is thresholded at $y > 3$.

$y = x_c + x_d + s; ; \; y_{bin} = \mathbb{1}\{y > 3\}$

### C.3.2. Learning Architecture.

The network $\phi$ contains a single linear layer with no bias which takes a constant (1) as an input and outputs a number $z_i$ for each data dimension $i$.

The network $\hat{h}$ takes as input $(x, w, y)$. It contains one linear layer with no bias which takes as input $[x, y]$ and outputs a single number $\hat{s}$. The second linear layer (with bias) takes as input $w$ and outputs the sigmoid activation of a single number, $switch$, representing the propensity to incorporate $s$ at $w$. It then outputs $w^\intercal x + switch \cdot \hat{s}$.

*Table 2.* Coefficients of Determination $R^2$, predicting each $x$ variable from each final $z$ variable.

| | RATE | TERM | DT | REC | INC | EMP |
|---|---|---|---|---|---|---|
| happiness | 0.00 | -0.15 | -0.14 | 0.00 | -0.01 | 0.00 |
| sadness | -0.01 | -0.06 | -0.10 | 0.00 | -0.04 | -0.07 |
| trustworthiness | 0.57 | 0.17 | 0.01 | 0.00 | -0.01 | -0.01 |
| dominance | 0.00 | -0.01 | 0.03 | -0.01 | 0.01 | -0.01 |
| hue | 0.48 | 0.29 | -0.02 | 0.00 | -0.04 | -0.02 |
| eye gaze | 0.42 | 0.46 | -0.04 | -0.40 | -0.04 | -0.17 |
| age | 0.23 | 0.22 | -0.12 | -0.21 | 0.17 | 0.04 |
| anger | -0.01 | -0.02 | -0.05 | -0.02 | -0.01 | 0.00 |
| fear | 0.04 | 0.00 | -0.03 | 0.00 | -0.01 | -0.01 |
| surprise | -0.18 | 0.04 | -0.01 | -0.02 | 0.00 | -0.04 |

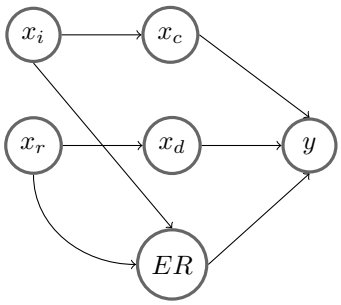



*Figure 11.* Relationship of variable correlations in the side information experiment

### C.3.3. BASELINES.

- **Machine Only**: The best possible linear model (with bias) trained to predict $y$ from $x_1 \dots x_4$.
- $h$(**Machine**): The human model $h$ applied to the best possible linear model (with bias) trained to predict $y$ from $x_1 \dots x_4$.

$$h(\text{Machine}) = \beta_0 + h(x, \beta_1, \dots, \beta_4, s)$$

where $\beta$ are the coefficients selected by the machine-only regression.

### C.3.4. HUMAN MODELS

- **Always**: The human always fully incorporates the side information,

$$h(x, w, s) = w^\intercal x + s$$

- **Never**: The human never incorporates the side information,

$$h(x, w, s) = w^\intercal x$$

- **Or**: The human becomes less likely to incorporate side information as weight is put on $x_i, x_r$,

$$h(x, w, s) = w^\intercal x + \sigma(1/\max(\max(x_i, x_r), .0001) - 2). \cdot s$$

Note that max(.0001) is required to prevent numerical overflow, and -2 recenters the sigmoid to allow for values $< .5$.

- **Coarse**: The human incorporates $s$ as in Or, but uses a coarse, noisy version of $s$, $s' = 2 \cdot \mathbb{1}\{s \geq 2\}$

$$h(x, w, s) = w^\intercal x + \sigma(1/\max(\max(x_i, x_r), .0001) - 2). \cdot s'$$

*Table 3.* Coefficients of Ridge Regression, predicting each $x$ variable from all final $z$ variables.

| | RATE | TERM | DT | REC | INC | EMP |
|---|---|---|---|---|---|---|
| happiness | -0.07 | -0.29 | -0.10 | -0.06 | 0.21 | -0.07 |
| sadness | 0.16 | 0.07 | 0.07 | -0.01 | 0.13 | 0.07 |
| trustworthiness | -0.62 | -0.28 | -0.05 | -0.23 | 0.31 | 0.16 |
| dominance | 0.05 | 0.16 | 0.12 | -0.13 | -0.02 | 0.04 |
| hue | 0.27 | 0.20 | 0.19 | 0.03 | 0.01 | -0.08 |
| eye gaze | 0.13 | 0.28 | -0.10 | 0.13 | -0.29 | -0.04 |
| age | -0.09 | 0.14 | 0.12 | -0.09 | 0.67 | 0.40 |
| anger | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| fear | 0.19 | 0.12 | 0.08 | -0.07 | 0.04 | 0.00 |
| surprise | 0.07 | 0.12 | 0.03 | -0.07 | -0.06 | 0.13 |

## D. Select Turker quotes

- "I wasn't always looking at just happiness or sadness. Sometimes the expressions seemed disingenuously happy, and that also threw me off. I don't know if that was intentional but it definitely effected my gut feeling and how I chose."
- "In my opinion, the level of happiness or sadness, the degree of a smile or a frown, was used to represent applications who were likely to be payed back. The more happy one looks, the better the chances of the client paying the loan off (or at least what the survey information lead me to believe)."
- "I was more comfortable with facial expressions than numbers. I felt like a computer and I didn't feel human anymore. Didn't like it at all."