Pragmatic Image Compression for Human-in-the-Loop Decision-Making

Anonymous Author(s)
Affiliation
Address
email

Abstract

Standard lossy image compression algorithms aim to preserve an image's appearance, while minimizing the number of bits needed to transmit it. However, the amount of information actually needed by the user for downstream tasks – e.g., deciding which product to click on in a shopping website – is likely much lower. To achieve this lower bitrate, we would ideally only transmit the visual features that drive user behavior, while discarding details irrelevant to the user's decisions. We approach this problem by training a compression model through human-in-the-loop learning as the user performs tasks with the compressed images. The key insight is to train the model to produce a compressed image that induces the user to take the same action that they would have taken had they seen the original image. To approximate the loss function for this model, we train a discriminator that tries to distinguish whether a user's action was taken in response to the compressed image or the original. We evaluate our method through experiments with human participants on four tasks: reading handwritten digits, verifying photos of faces, browsing an online shopping catalogue, and playing a car racing video game. The results show that our method learns to match the user's actions with and without compression at lower bitrates than baseline methods, and adapts the compression model to the user's behavior: it preserves the digit number and randomizes handwriting style in the digit reading task, preserves hats and eyeglasses while randomizing faces in the photo verification task, preserves the perceived price of an item while randomizing its color and background in the online shopping task, and preserves upcoming bends in the road in the car racing game.

1 Introduction

Modern web platforms serve billions of images every day, and typically rely on lossy compression algorithms to store and transmit this data efficiently. Recent work on machine learning methods for lossy image compression [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11] improves upon standard methods like JPEG [12] by training neural networks to minimize the number of bits needed to generate high-fidelity reconstructions. In this paper, we explore the idea of compressing images to even smaller sizes by intentionally allowing reconstructions to deviate drastically from the visual appearance of their originals, and instead optimizing reconstructions for the specific, downstream tasks that the user wants to perform with them, such as video conferencing, online shopping, or remotely operating space robots [13].

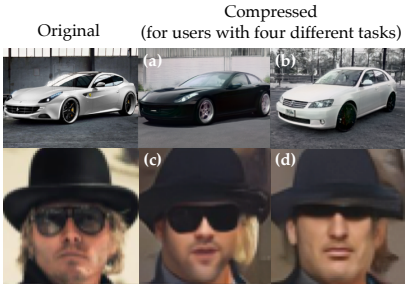

Figure 1: Images compressed 2-4x smaller than JPEG retain information for tasks like shopping for cars in a perceived price range (a), surveying car colors (b), and checking photos for eyeglasses (c) or hats (d).

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

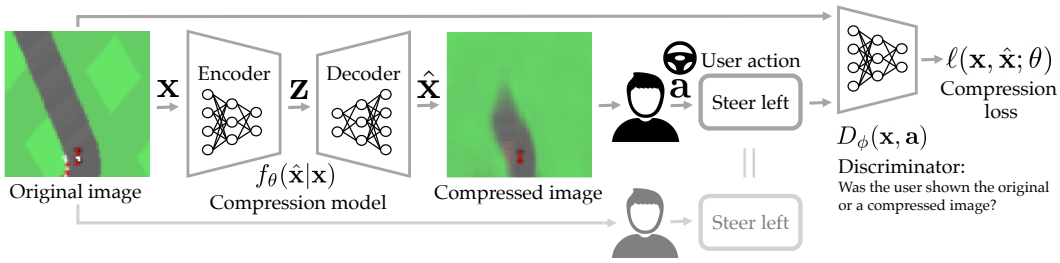

Figure 2: Given the original image $\mathbf{x}$, we would like to generate a compressed image $\hat{\mathbf{x}}$ such that the user's action $\mathbf{a}$ upon seeing the compressed image is similar to what it would have been had the user seen the original image instead. In a 2D top-down car racing video game, our compression model learns that, in order to match the user's steering with and without compression, it must preserve bends, but can discard the road farther ahead.

Our main observation in this work is that, instead of optimizing the compression model for a task-agnostic perceptual loss function, we can instead optimize the compression model for *behavioral equivalence*: producing compressed images that, when shown to the user, induce the user to take the same actions that they would have taken had they observed the original, uncompressed images. We call this *pragmatic compression* (PICO). PICO adapts compression to user behavior, enabling the user to perform their individually-desired tasks with compressed images instead of the original images. For example, consider two users with distinct tasks: one flying a quadcopter, and the other driving a ground robot. On a network with an extremely low bitrate, we would like the compressed video feed of the ground robot to preserve ground-level obstacles and terrain while discarding details about power lines and tree canopies, and the quadcopter feed to do the opposite.

To this end, we formulate compression as a human-in-the-loop learning problem, in which the compression model is represented as an encoder-decoder neural network that takes the original image as an input and outputs the compressed image. The user sees the compressed image, and takes an action to perform their desired task (see Figure 2). The main challenge in this work is designing a loss function for the compression model that evaluates the quality of the compressed image in the context of the original image and the user's action. We do not assume knowledge of the user's desired task, so we cannot directly evaluate the quality of the compressed image by evaluating the fitness of the user's action upon seeing the compressed image. We also do not assume access to ground-truth action labels for the original images in the streaming setting, so we cannot compare the user's action upon seeing the compressed image to some ground-truth action. Instead, we define the loss function through adversarial learning. For example, consider a user browsing an online shopping catalogue, observing photos and clicking on appealing items. To collect positive and negative examples of user behavior, we simply randomize whether a user sees the original or compressed version of an image while they are shopping, and record their actions. We then train a discriminator to predict the likelihood that a user's action was taken in response to the original rather than a compressed image, and train the compression model to maximize this predicted likelihood.

Our primary contribution is the PICO algorithm for human-in-the-loop learning of data compression models. We validate PICO through three user studies on Amazon Mechanical Turk, in which we train and evaluate our compression models on data from human participants. In the first study, we asked participants to read handwritten digits and identify the numbers – PICO learned to preserve the number and discard handwriting style (Figure 3). In the second study, we asked users to browse a car catalogue and select cars based on perceived price – PICO learned to preserve overall shape and sportiness while randomizing paint jobs and backgrounds (Figure 4). In the third study, we asked participants to verify photos of faces by checking if heads or eyes were covered – PICO learned to preserve hats and eyeglasses while randomizing faces (Figure 5). In all three studies, PICO achieved up to 2-4x lower bitrates than task-independent baseline methods. To show that PICO can be used in sequential decision-making problems, we also ran a user study with 12 participants who played a car racing video game – at a fixed bitrate, PICO learned to preserve bends in the road substantially better than a task-independent baseline method, enabling users to drive more safely (Figure 6).

Related work. Prior work on learned lossy image compression focuses on overcoming various challenges in training neural networks on images [14], including amortized variable-rate compression [1, 4], end-to-end training with quantization [2, 5, 6], optimizing the rate-distortion trade-off [7, 8], optimizing perceptual quality [9, 10, 11], training hierarchical latent variable models [3], and

sequential compression of videos [15, 16]. While these methods aim to generate visually-pleasing reconstructions that are perceptually similar to their originals, PICO focuses on matching user actions with and without compression in specific downstream tasks. Hence, PICO can achieve substantially lower bitrates while preserving functional equivalence (see Section 4.1). Prior work has studied human-in-the-loop learning in related contexts, including reinforcement learning of text summarization policies from user feedback [17] and automatic data visualization for decision support systems [18]. The idea of learning a model of human behavior using generative adversarial networks [19] has also been explored in prior work on imitation learning [20]. PICO differs from [18, 20] in that it tackles image compression – an entirely different problem from decision support and imitation learning. In contrast to [17], which elicits user comparisons between different summaries of the same text, PICO can be used for sequential tasks like video games (see Section 4.3) where the user cannot be repeatedly queried with different compressed versions of the same image.

2 Pragmatic Compression

Generative models are typically used for sampling and representation learning, but they can also be used for compression [21, 22, 23, 24]. For example, variational autoencoders [25] are trained with a variational information bottleneck [26] that explicitly constrains the amount of information carried by their latent variables – hence, we can use a trained encoder to compress an image, and a trained decoder to reconstruct it from latent features [5, 27]. In contrast to compression methods that train such generative models to maximize the visual fidelity of the reconstruction, we formulate compression as a problem of *control*, including the downstream behavior of the user in the problem statement. First, the environment generates an image $\mathbf{x} \in \mathbb{R}^{w \times h \times c}$. Given the original image $\mathbf{x}$, the compression system generates a compressed image $\hat{\mathbf{x}} \in \mathbb{R}^{w \times h \times c}$ that can be represented using no more than n bits, where n is a hyperparameter. The user then observes the compressed image $\hat{\mathbf{x}}$ and samples an action $\mathbf{a} \sim \pi(\mathbf{a}|\hat{\mathbf{x}})$ from their unknown policy π. We assume the user's utility function $U(\mathbf{x}, \mathbf{a})$ is unknown, and that the user cannot directly specify their desired task. Our goal is to generate a compressed image $\hat{\mathbf{x}}$ that induces an action $\mathbf{a}$ that maximizes the unknown utility $U(\mathbf{x}, \mathbf{a})$, while minimizing the amount of human interaction required to train the compression system.

Let $f_\theta(\hat{\mathbf{x}}|\mathbf{x})$ denote a parametric model of our compression function, where θ are the model parameters (e.g., neural network weights). We cannot train the compression model $f_\theta(\hat{\mathbf{x}}|\mathbf{x})$ to directly optimize the user's utility $U(\mathbf{x}, \mathbf{a})$, since the utility function U and the user's policy π are unknown. Instead, we aim to maximize the behavioral equivalence of the compressed image $\hat{\mathbf{x}}$ and original $\mathbf{x}$. Let $T \in \{0, 1\}$ denote whether the user sees the original or the compressed image before taking an action: if $T = 0$, then set $\hat{\mathbf{x}} \leftarrow \mathbf{x}$; else if $T = 1$, sample $\hat{\mathbf{x}} \sim f_\theta(\hat{\mathbf{x}}|\mathbf{x})$. We would like to train the compression model to minimize the divergence of the user's policy evaluated on the compressed image $\pi(\mathbf{a}|\hat{\mathbf{x}})$ from the user's policy evaluated on the original $\pi(\mathbf{a}|\mathbf{x})$,

$$\begin{aligned}\mathcal{L}(\theta) &= \mathbb{E}_{\mathbf{x}}[D(\pi(\mathbf{a}|\mathbf{x}) \parallel \mathbb{E}_{\hat{\mathbf{x}} \sim f_\theta(\hat{\mathbf{x}}|\mathbf{x})}[\pi(\mathbf{a}|\hat{\mathbf{x}})|\mathbf{x}])] \\ &= \mathbb{E}_{\mathbf{x}}[D(p(\mathbf{a}|\mathbf{x}, T = 0) \parallel p(\mathbf{a}|\mathbf{x}, T = 1))], \qquad (1)\end{aligned}$$

where D is a divergence (e.g., the Jensen-Shannon divergence). We still cannot directly optimize this loss function, since we do not know the user's policy π. We address this issue by using conditional generative adversarial networks (GAN) [19, 28] to approximately minimize the loss.

To train the GAN discriminator, we need positive and negative examples of user behavior; in our case, examples of user behavior with and without compression. To collect these examples, we randomize whether the user sees the compressed image or the original before taking an action. Let $T \sim \text{Bernoulli}(0.5)$ represent this random assignment. When $T = 0$, the user sees the original $\mathbf{x}$ and takes action $\mathbf{a}$, and we record $(\mathbf{x}, \hat{\mathbf{x}}, \mathbf{a})$ as a positive example of user behavior. When $T = 1$, the user sees the compressed image $\hat{\mathbf{x}}$ and takes action $\mathbf{a}$, and we record $(\mathbf{x}, \hat{\mathbf{x}}, \mathbf{a})$ as a negative example. Let $\mathcal{D}$ denote the dataset of all recorded $(T, \mathbf{x}, \hat{\mathbf{x}}, \mathbf{a})$ tuples. We train a discriminator $D_\phi(\mathbf{x}, \mathbf{a})$ to predict the likelihood $p(T = 0|\mathbf{x}, \mathbf{a})$, using the standard binary cross-entropy loss and the training data $\mathcal{D}$. Note that the discriminator is conditioned on the original image $\mathbf{x}$ and the user action $\mathbf{a}$, but not the compressed image $\hat{\mathbf{x}}$ – this follows from our problem formulation in Equation 1, and ensures that the discriminator focuses on differences in the user's behavior rather than superficial differences in the images. Given this discriminator, we would like to train the compression model f_θ to generate compressed images $\hat{\mathbf{x}}$ that induce actions $\mathbf{a}$ that maximize $\log D_\phi(\mathbf{x}, \mathbf{a})$, since that would approximately minimize the loss in Equation 1 when D is the Jensen-Shannon divergence [19, 28].

Algorithm 1 Pragmatic Compression (PICO)

Initialize compression model f_θ
while true **do**
  $\mathbf{x} \sim p_{\text{env}}(\mathbf{x})$ ▷ environment generates original image
  $T \sim \text{Bernoulli}(0.5)$ ▷ randomly decide whether user sees compressed image or original
  if $T = 0$ **then** $\hat{\mathbf{x}} \leftarrow \mathbf{x}$ **else** $\hat{\mathbf{x}} \sim f_\theta(\hat{\mathbf{x}}|\mathbf{x})$
  $\mathbf{a} \sim \pi(\mathbf{a}|\hat{\mathbf{x}})$ ▷ user takes action using unknown policy
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{(T, \mathbf{x}, \hat{\mathbf{x}}, \mathbf{a})\}$
  $\phi \leftarrow \phi + \nabla_\phi \sum_{(T,\mathbf{x},\mathbf{a})\in\mathcal{D}} T \cdot \log(1 - D_\phi(\mathbf{x}, \mathbf{a})) + (1 - T) \cdot \log D_\phi(\mathbf{x}, \mathbf{a})$ ▷ update discriminator
  $\psi \leftarrow \psi - \nabla_\psi \sum_{(\mathbf{x},\hat{\mathbf{x}},\mathbf{a})\in\mathcal{D}} D_{\text{KL}}(D_\phi(\mathbf{x}, \mathbf{a}) \parallel D_\psi(\mathbf{x}, \hat{\mathbf{x}}))$ ▷ update action-agnostic discriminator
  $\theta \leftarrow \theta + \nabla_\theta \sum_{\mathbf{x}\in\mathcal{D}} \log D_\psi(\mathbf{x}, f_\theta(\mathbf{x}))$ ▷ update compression model

The discriminator D_ϕ addresses the problem of not knowing the user's policy π, but is impractical to use as a loss function for the compression model. Since $D_\phi(\mathbf{x}, \mathbf{a})$ takes the user's action $\mathbf{a}$ as input, and that action may change as we optimize the compression model and produce different compressed images $\hat{\mathbf{x}}$ than those in the training set, we can only use D_ϕ to evaluate the quality of the compressed images in the training set. As such, we would need to repeatedly query the user for new actions after every gradient update of the compression model, which would be prohibitively expensive. Instead, we train a separate, action-agnostic discriminator model to emulate the discriminator's output given only the original and compressed images as input, not the user's action, and then train the compression model to maximize the action-agnostic discriminator's outputs. In particular, we train an action-agnostic discriminator $D_\psi(\mathbf{x}, \hat{\mathbf{x}})$ to approximate the discriminator $D_\phi(\mathbf{x}, \mathbf{a})$ by optimizing

$$\ell(\psi) = \sum_{(\mathbf{x},\hat{\mathbf{x}},\mathbf{a})\in\mathcal{D}} D_{\text{KL}}(D_\phi(\mathbf{x}, \mathbf{a}) \parallel D_\psi(\mathbf{x}, \hat{\mathbf{x}})). \tag{2}$$

Note that, since $D_\psi(\mathbf{x}, \hat{\mathbf{x}})$ is trained to mimic $D_\phi(\mathbf{x}, \mathbf{a})$, D_ψ does not learn to discriminate based on superficial visual differences between $\mathbf{x}$ and $\hat{\mathbf{x}}$, but rather based on semantic differences that cause the user's actions to change in ways detected by D_ϕ. Given the trained action-agnostic discriminator D_ψ, we train the compression model using the standard GAN generator loss [19, 28],

$$\ell(\theta) = \sum_{\mathbf{x}\in\mathcal{D}} -\log D_\psi(\mathbf{x}, f_\theta(\mathbf{x})), \tag{3}$$

where $f_\theta(\mathbf{x})$ denotes $\mathbb{E}_{\hat{\mathbf{x}}\sim f_\theta(\hat{\mathbf{x}}|\mathbf{x})}[\hat{\mathbf{x}}|\mathbf{x}]$. Our complete pragmatic compression method is summarized in Algorithm 1. We randomly initialize the compression model f_θ. The environment samples an original image $\mathbf{x}$ from an unknown distribution p_{env}. To decide whether the user sees the original or compressed image, we sample a Bernoulli random variable T. After seeing the chosen image, the user samples an action $\mathbf{a}$ from their unknown policy π. To update the discriminator D_ϕ, we take a gradient step on the binary cross-entropy loss. To update the action-agnostic discriminator D_ψ, we take a gradient step on the KL-divergence loss in Equation 2. To update the compression model f_θ, we take a gradient step on the GAN generator loss in Equation 3. See Appendix A.3 for details.

3 Structured Compression using Generative Models

One approach to representing the compression model f_θ could be to structure it as a variational autoencoder (VAE) [25], and train the VAE end to end on the loss function in Equation 3 instead of the standard reconstruction error loss. This approach is fully general, but requires training a separate model for each desired bitrate (which is determined by the β coefficient in the VAE training objective), and can require extensive exploration of the pixel output space before it discovers an effective compression model. To simplify variable-rate compression and exploration in our experiments, we forgo end-to-end training, and first train a generative model on a batch of images without the human in the loop by optimizing a task-agnostic perceptual loss, yielding an encoder and decoder such that $\mathbf{z} = \text{enc}(\mathbf{x})$ and $\hat{\mathbf{x}} = \text{dec}(\mathbf{z})$. We then train our compression model $f_\theta(\hat{\mathbf{z}}|\mathbf{z})$ to compress the latent embedding generated by the encoder, instead of compressing the original pixels. We use a variety of different generative models in our experiments, including a β-VAE [29] for the handwritten digit identification experiments in Figure 3, a StyleGAN2 model [30] for the car shopping experiments in Figure 4, an NVAE model [31] for the photo verification experiments in Figure 5, and a VAE for the car racing experiments in Figure 6. See Appendix A.4 for details.

Generative models like the VAE and StyleGAN2 tend to learn disentangled features – hence, instead of training f_θ to map directly to the latent space $\mathbb{R}^d$, we structure f_θ to output a vector of probabilities that determines which latent features are transmitted exactly between $\mathbf{z}$ and $\hat{\mathbf{z}}$, and which other features are masked out and then reconstructed from the prior distribution. In particular, we structure $f_\theta : \mathbb{R}^d \mapsto [0,1]^d$ to output a vector of mask probabilities $\boldsymbol{p} \in [0,1]^d$ given the latent embedding $\mathbf{z} \in \mathbb{R}^d$. Then, given a hyperparameter $\lambda \in [0,1]$ that controls the compression rate, we transmit the $\lfloor \lambda d \rfloor$ latent features i with the lowest mask probabilities $\boldsymbol{p}_i$, and mask out the remaining $d - \lfloor \lambda d \rfloor$ features. We reconstruct the masked features by assuming that $\hat{\mathbf{z}}$ follows a multivariate normal distribution, and sampling the masked feature values from the conditional prior distribution given the transmitted feature values. See Appendix A.4 for details.

This design of the compression model f_θ simplifies variable-rate compression: at test time, we simply choose a value of λ that achieves the desired bitrate, without retraining the model. It also simplifies exploration: instead of exploring in pixel output space, we explore in the space of masks over latent features, which leverages the decoder to generate more realistic compressed images during the early stages of training. We can now also simplify the action-agnostic discriminator: instead of training $D_\psi(\mathbf{x}, \hat{\mathbf{x}})$, we train $D_\psi(\mathbf{x}, \boldsymbol{p})$, since the mask input $\boldsymbol{p} \in [0,1]^d$ has far fewer dimensions than the compressed image input $\hat{\mathbf{x}} \in \mathbb{R}^{w \times h \times c}$. In our experiments, we also leverage the low-dimensional mask output space to perform batch learning instead of online learning, which greatly simplifies our implementation of PICO with real users. See Appendix A.1 for additional discussion. While these simplifications enable us to provide a proof of concept for pragmatic compression in this paper, we acknowledge that end-to-end training of the compression model is a more general approach that does not involve learning a separate generative model – this is a promising direction for future work.

4 User Studies

In our experiments, we evaluate to what extent PICO can minimize the number of bits needed to transmit an image, while still preserving the image's usefulness to users performing downstream tasks. We conduct user studies on Amazon Mechanical Turk, in which we ask human participants to complete three tasks at varying bitrates: reading handwritten digits from the MNIST dataset [32], verifying attributes of faces from the CelebA dataset [33], and browsing a shopping catalogue of cars from the LSUN Car dataset [34]. To study PICO's performance on sequential decision-making problems, we also run an experiment with 12 participants who play the Car Racing video game from OpenAI Gym [35] under a constraint on the bitrate of the video feed. In all experiments, we trained our discriminators and compression model on 1000 negative examples and varying numbers of positive examples (see Appendices A.1 and A.5). Appendix A discusses the implementation details.

4.1 Trade-off between Minimizing Bitrate and Maximizing User Action Agreement

We claim that PICO can learn to transmit only the features that users need to perform their tasks. Our first set of user studies seeks to answer **Q1**: to what extent can PICO minimize the bitrate while trying to match the user's actions with and without compression? We would like to study this question in domains where we can actually measure the agreement between the user's actions with and without compression, by collecting action labels for the original images and comparing the user's actions upon seeing compressed versions of those images to the labels. As such, we run experiments on Amazon Mechanical Turk that focus on single-step decision-making settings where we can collect action labels for a fixed dataset of images: (a) identifying a handwritten digit, and (b) clicking on an item in a shopping catalogue. In (a), we instruct users to identify the number in the image within the range 0-9. In (b), to simulate the experience of browsing a catalogue on a budget, we instructed users to click on images of cars that they perceive to be worth less than $20,000.

In both domains, we evaluate PICO by varying the bitrate and, at each bitrate, measuring the agreement of user actions upon seeing a compressed image, with user actions upon seeing the original version of that image. As discussed in Section 3, PICO learns a compression model f_θ that, given a separate generative model, selects which latent features to transmit. Since the purpose of this experiment is to test whether PICO's task-dependent compression is beneficial, we compare to a task-independent baseline method that selects a uniform-random subset of features to transmit, but otherwise uses the same generative model as PICO – this enables us to conduct an apples-to-apples comparison that isolates the effect of training f_θ on user behavior data.

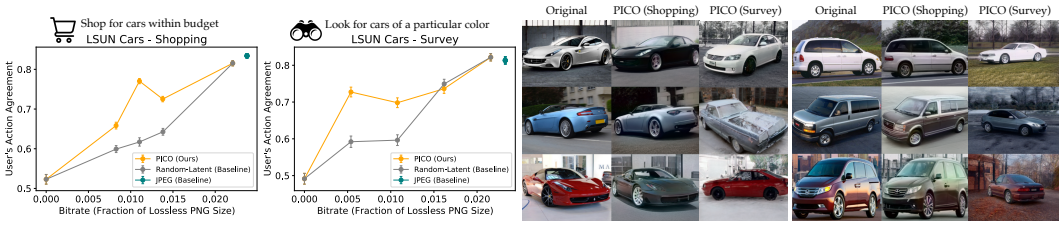

Figure 4: LSUN Car shopping experiment that addresses **Q1**, and survey experiment that addresses **Q2**. The plots show action agreement evaluated on 100 held-out images, with error bars representing standard error. The average lossless PNG file size is 247kB, and each image has dimensions 512x512x3. The shopping samples show that, when users are instructed to click on cars they perceive to be worth less than $20,000, PICO learns to preserve the overall shape and sportiness of the car, while randomizing paint jobs, backgrounds, and other details that are irrelevant to the users' perception of price. However, when users are instead instructed to determine whether the car is "dark-colored" or "light-colored" for a survey task, PICO learns to preserve the car's color while randomizing its pose. We intentionally show compressed image samples for a low bitrate (0.011) to highlight differences between the compression models learned for the two tasks.

To provide a point of comparison to widely-used compression methods, we also compare to a JPEG [12] baseline method, where the quality parameter is set to the lowest possible value of 1 in order to bring the JPEG bitrate as close as possible to the range of bitrates achieved by PICO and the random-latent baseline. Though JPEG is no longer a state-of-the-art method, it enables us to roughly calibrate the bitrates achieved by PICO and the random-latent baseline.

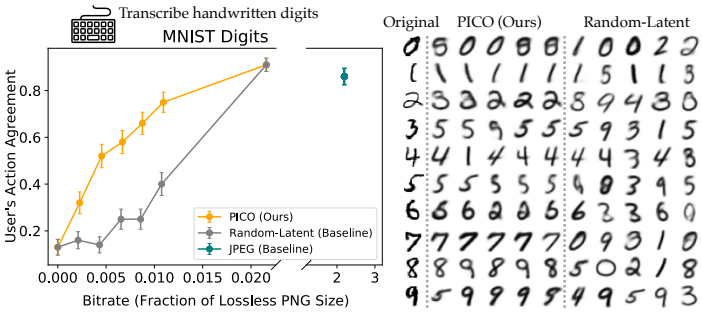

Figure 3: MNIST digit identification experiments that address **Q1**. When users are instructed to identify the digit number, PICO learns to preserve the digit number while randomizing handwriting style. The plots show user action agreement evaluated on 100 held-out images, with error bars representing standard error. The average lossless PNG file size is 0.3kB, and each image has dimensions 28x28x1. Each of the five columns in the two groups of compressed images represents a different sample from the stochastic compression model $f(\hat{\mathbf{x}}|\mathbf{x})$.

Figures 3 and 4 show that, at low bitrates, PICO achieves substantially higher user action agreement than the random-latent baseline (orange vs. gray). PICO also achieves much lower bitrates than the JPEG baseline (orange vs. teal), while maintaining comparable agreement on MNIST but lower agreement on LSUN Car. The samples in Figure 3 show that PICO learns to preserve digit numbers more often than the random-latent baseline, while randomizing handwriting style in order to satisfy the bitrate constraint. The samples in Figure 4 show that, for users performing the shopping task, PICO learns to preserve the overall shape and sportiness of the car, while randomizing paint jobs, backgrounds, and other details that are irrelevant to the user's perception of the price of the car. The dip in the orange curve in the shopping plot may be due to the fact that increasing the bitrate preserves more of the encoded latent features, which, when combined with features sampled from the prior, can be out-of-distribution inputs to the StyleGAN2 decoder [36, 37], potentially leading to degraded image quality for some examples (Appendix A.4 discusses our StyleGAN2 encoding method). Figures 7 and 8 in the appendix include more examples.

4.2 Adapting Compression to Different Downstream Tasks

The experiments in the previous section show that PICO can outperform a task-independent baseline method by transmitting only the features that users need to perform their tasks. Our second set of user studies investigates this mechanism further, by asking **Q2**: can PICO adapt the compression model to the specific needs of different downstream tasks in the same domain? To answer this question, we run two photo verification experiments: one in which users are instructed to check if the person's head is covered (e.g., by a hat) and click on one of two buttons labeled "covered" and "not covered",

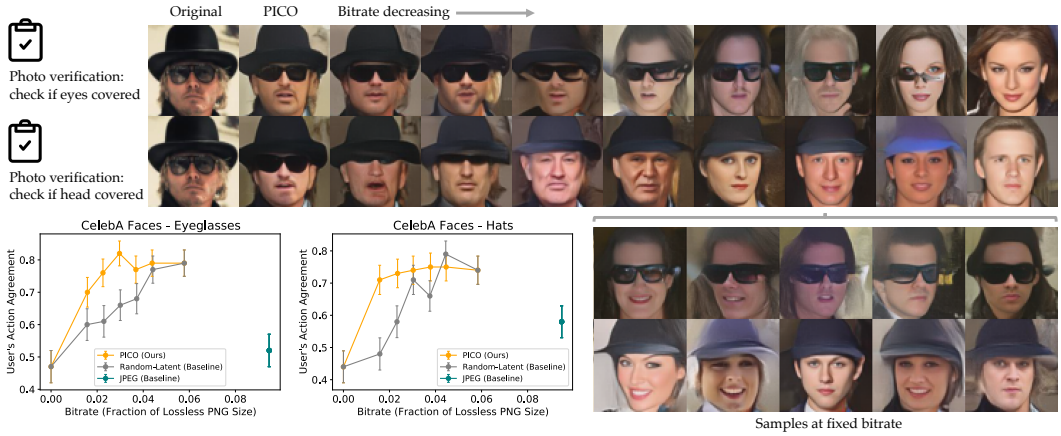

Figure 5: CelebA photo attribute verification experiments that address **Q2**. Depending on the instructions given to the user, PICO learns to either preserve hats or eyeglasses while randomizing faces and other task-irrelevant details. The plots show action agreement evaluated on 100 held-out images, with error bars representing standard error. The average lossless PNG file size is 7.7kB, and each image has dimensions 64x64x3.

and another in which users check if the person's eyes are covered (e.g., by eyeglasses). We also run an additional experiment in the LSUN Car domain from the previous section, in which we simulate a survey task that asks users to 'help a car dealership conduct market research' by determining whether an observed car has a "dark-colored" or "light-colored" paint job.

Figure 5 shows that, at low bitrates, PICO achieves substantially higher user action agreement than the random-latent baseline (orange vs. gray). PICO also achieves much lower bitrates and higher agreement than the JPEG baseline (orange vs. teal). In each experiment, PICO adapted the compression model to the user's particular task. When users checked eyes, PICO learned to preserve the presence of eyeglasses while randomizing hair color, faces, and other irrelevant details (see top row of samples). On the other hand, when users checked for head coverings like hats and helmets, PICO learned to preserve the presence of hats while randomizing eyes and other details (see second row of samples). The third and fourth rows of samples illustrate the fact that PICO learns a stochastic compression model $f_\theta(\hat{\mathbf{x}}|\mathbf{x})$ from which we can draw multiple compressed samples $\hat{\mathbf{x}}$ for a given original $\mathbf{x}$. The fact that all the samples in the third row have eyeglasses but differ in other attributes like pose angle, and those in the fourth row all have hats while some are smiling and some are not, shows that even though the compression model is stochastic, it produces stable attributes when they are needed for the downstream task. Figure 7 in the appendix includes more qualitative examples. In addition to these photo verification results, the samples in Figure 4 illustrate substantial differences in the compression models learned for the car shopping and survey tasks.

4.3 Compressing Observations for Sequential Decision-Making

Our third user study seeks to answer **Q3**: can PICO learn to compress image observations in the sequential decision-making setting? To answer this question, we run an experiment with 12 participants in which we ask users to play a 2D top-down car racing video game, while constraining the number of bits that can be used to transmit the image observation to the user at each timestep. Due to the difficulties of using action labels to measure action agreement outlined in Section 4.1, we instead measure the user's progress along the road in the game – specifically, the fraction of new road patches visited during an episode. In these experiments, we fix the bitrate to 85 bits per step, which is well below the 170 bits per step required to transmit the full set of features for the 64x64x3 images.

Figure 6 shows that, at a fixed bitrate, PICO enables the user to perform substantially better on the driving task than the random-latent compression baseline (orange vs. gray), and comparably to a positive control in which we do not compress the image observations at all (orange vs. teal). The first and second film strips show that, when we use the random-latent compression baseline, there is a substantial difference between the originals and the compressed images. For example, even at the first timestep, the compressed image shows the road to be less tilted than it actually is, so in the next frame we see that the user has mistakenly driven forward and ended up in the grass instead of

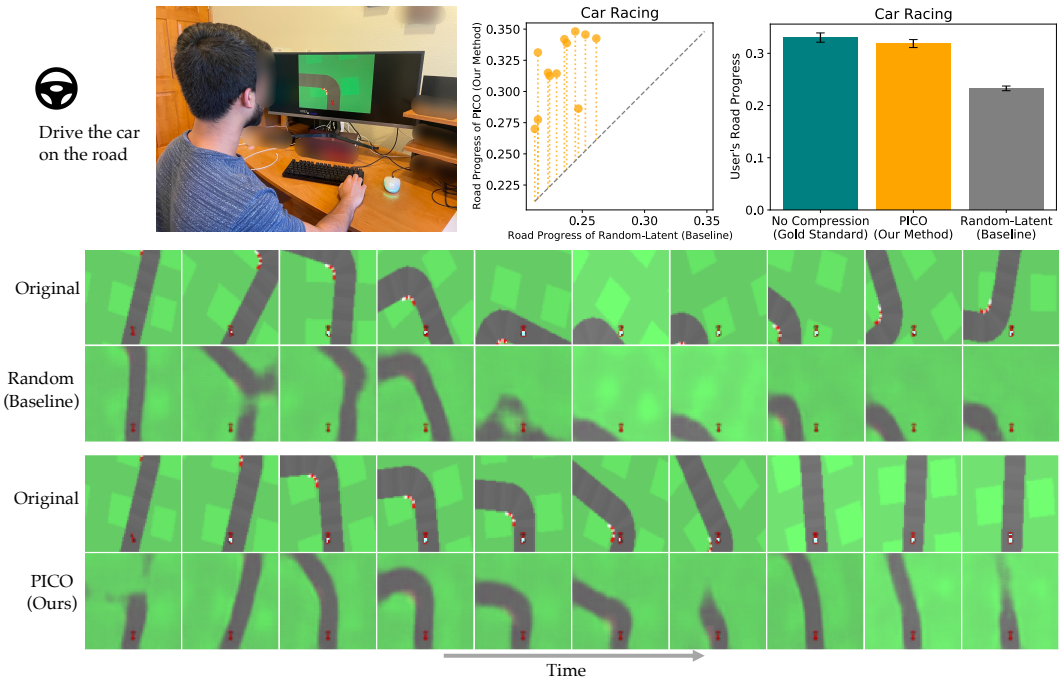

Figure 6: Car Racing game experiments that address **Q3**. The scatter plot shows that, for each of the 12 users (orange), road progress with PICO was substantially higher than with the random-latent compression baseline. The bar chart shows road progress averaged over all users, with error bars representing standard error.

turning right to stay on the road. In contrast, the third and fourth film strips show that PICO has learned to preserve the angle of the road, while discarding the details of the road much farther ahead in order to satisfy the bitrate constraint. We ran a one-way repeated measures ANOVA on the road progress metrics from the random-latent baseline and PICO conditions with the presence of PICO as a factor, and found that $f(1, 11) = 176.32, p < .0001$. The subjective evaluations in Table 1 in the appendix corroborate these results: users reported feeling higher situational awareness and ability to control the car with PICO compared to the random-latent baseline. After evaluating PICO, one user commented, "This environment was a lot easier. It felt more consistent. I felt like we had a mutual understanding of when I would turn and what it would show me to make me turn." Appendices A.7 and B discuss the subject allocation and results in more detail. Split-screen videos of compressed and uncompressed game play are included in the supplementary materials.

5 Discussion

Our experiments show that, for a variety of tasks with different kinds of images, pragmatic compression can reduce bitrates 2-4x compared to task-independent baseline methods, by optimizing reconstructions for behavioral equivalence. This paper is a proof of concept that, through human-in-the-loop learning, we can train models to communicate relevant information to users under network bandwidth constraints, without prior knowledge of the users' desired tasks. The main limitation of our experiments, as mentioned in Section 3, is that we train a separate generative model in order to simplify variable-rate compression and exploration. This can be problematic when the generative model does not include task-relevant features in its latent space – e.g., the yellow sports car in rows 7-8 of Figure 8 in the appendix gets distorted when encoded into the StyleGAN2 latent space, even without any additional compression. One direction for future work is to train the compression model end to end instead, so that it can learn latent features tailored to the user's tasks. End-to-end training could also enable PICO to be applied to problems other than compression, such as image captioning for visually-impaired users, or audio visualization for hearing-impaired users [38]. Pragmatic compression is applicable to a variety of emerging human-in-the-loop decision-making problems, including video compression for robotic space exploration [13], audio compression for hearing aids [39, 40], and spatial compression for virtual reality [41].

References

[1] George Toderici, Sean M O'Malley, Sung Jin Hwang, Damien Vincent, David Minnen, Shumeet Baluja, Michele Covell, and Rahul Sukthankar. Variable rate image compression with recurrent neural networks. *arXiv preprint arXiv:1511.06085*, 2015.

[2] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016.

[3] Karol Gregor, Frederic Besse, Danilo Jimenez Rezende, Ivo Danihelka, and Daan Wierstra. Towards conceptual compression. *arXiv preprint arXiv:1604.08772*, 2016.

[4] George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell. Full resolution image compression with recurrent neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[5] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with compressive autoencoders. *arXiv preprint arXiv:1703.00395*, 2017.

[6] Eirikur Agustsson, Fabian Mentzer, Michael Tschannen, Lukas Cavigelli, Radu Timofte, Luca Benini, and Luc Van Gool. Soft-to-hard vector quantization for end-to-end learning compressible representations. *arXiv preprint arXiv:1704.00648*, 2017.

[7] Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, Radu Timofte, and Luc Van Gool. Conditional probability models for deep image compression. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[8] Mu Li, Wangmeng Zuo, Shuhang Gu, Debin Zhao, and David Zhang. Learning convolutional networks for content-weighted image compression. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[9] Michael Tschannen, Eirikur Agustsson, and Mario Lucic. Deep generative models for distribution-preserving lossy compression. In *Neural Information Processing Systems*, 2018.

[10] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[11] Fabian Mentzer, George D Toderici, Michael Tschannen, and Eirikur Agustsson. High-fidelity generative image compression. *Neural Information Processing Systems*, 2020.

[12] Gregory K Wallace. The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics*, 1992.

[13] Terrence Fong, Jennifer Rochlis Zumbado, Nancy Currie, Andrew Mishkin, and David L Akin. Space telerobotics: unique challenges to human–robot collaboration in space. *Reviews of Human Factors and Ergonomics*, 2013.

[14] J Jiang. Image compression with neural networks – a survey. *Signal Processing: Image Communication*, 1999.

[15] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. DVC: An end-to-end deep video compression framework. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[16] Salvator Lombardo, Jun Han, Christopher Schroers, and Stephan Mandt. Deep generative video compression. In *Neural Information Processing Systems*, 2019.

[17] Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback. *arXiv preprint arXiv:2009.01325*, 2020.

[18] Sophie Hilgard, Nir Rosenfeld, Mahzarin R Banaji, Jack Cao, and David C Parkes. Learning representations by humans, for humans. *arXiv preprint arXiv:1905.12686*, 2019.

[19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Neural Information Processing Systems*, 2014.

[20] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Neural Information Processing Systems*, 2016.

[21] Claude E Shannon. A mathematical theory of communication. *The Bell system technical journal*, 1948.

[22] Brendan J Frey and Geoffrey E Hinton. Efficient stochastic source coding and an application to a bayesian network source model. *Computer Journal*, 1997.

[23] Guy E Blelloch. Introduction to data compression. 2001.

[24] David JC MacKay. *Information theory, inference and learning algorithms*. 2003.

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

[26] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

[27] James Townsend, Tom Bird, and David Barber. Practical lossless compression with latent variables using bits back coding. *arXiv preprint arXiv:1901.04866*, 2019.

[28] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[29] Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. *Neural Information Processing Systems*, 2018.

[30] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[31] Arash Vahdat and Jan Kautz. NVAE: A deep hierarchical variational autoencoder. *arXiv preprint arXiv:2007.03898*, 2020.

[32] Yann LeCun. The MNIST database of handwritten digits, 1998.

[33] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *International Conference on Computer Vision (ICCV)*, December 2015.

[34] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[35] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. *arXiv preprint arXiv:1606.01540*, 2016.

[36] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. *arXiv preprint arXiv:2008.00951*, 2020.

[37] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *arXiv preprint arXiv:2102.02766*, 2021.

[38] Yuchi Zhang, Willis Peng, Bastian Wandt, and Helge Rhodin. Audioviewer: Learning to visualize sound. *arXiv preprint arXiv:2012.13341*, 2020.

[39] Alex Armstrong, Chi Chung Lam, Shievanie Sabesan, and Nicholas A Lesica. The hearing aid dilemma: amplification, compression, and distortion of the neural code. *bioRxiv*, 2020.

[40] Nasim Alamdari, Edward Lobarinas, and Nasser Kehtarnavaz. Personalization of hearing aid compression by human-in-the-loop deep reinforcement learning. *IEEE Access*, 2020.

[41] Niels Christian Nilsson, Tabitha Peck, Gerd Bruder, Eri Hodgson, Stefania Serafin, Mary Whitton, Frank Steinicke, and Evan Suma Rosenberg. 15 years of research on redirected walking in immersive virtual environments. *IEEE Computer Graphics and Applications*, 2018.

[42] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[43] David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *arXiv preprint arXiv:1809.01999*, 2018.

[44] Michael Frank, Noah Goodman, Peter Lai, and Joshua Tenenbaum. Informative communication in word production and word learning. In *Proceedings of the annual meeting of the cognitive science society*, 2009.

[45] Herbert P Grice. Logic and conversation. In *Speech acts*. 1975.

[46] Dan Sperber and Deirdre Wilson. *Relevance: Communication and cognition*. 1986.

[47] Siddharth Reddy, Sergey Levine, and Anca D Dragan. Assisted perception: Optimizing observations to communicate state. *arXiv preprint arXiv:2008.02840*, 2020.