A Experimental Setup

Here, we discuss the implementation details of the experiments in Section 5. Source code is included in the supplementary materials.

A.1 Batch Learning from Logged Compression Data

In our experiments, we found that initializing the compression model such that it outputs uniform-random mask probabilities, collecting a batch of 1000 tuples $(T = 1, \mathbf{x}, \hat{\mathbf{x}}, \mathbf{a})$ using this random compression model, and training the models D_ϕ, D_ψ, and f_θ to convergence on this data yielded a high-performing compression model f_θ. In other words, rather than alternating one step of data collection with one gradient step as in Algorithm 1, we used batch learning. The initial random compression model explored the structured output space of feature masks well enough to generate useful training data for our models, so we did not need to learn from on-policy data generated by partially-trained compression models. This approach illustrates how PICO can be practically deployed in real-world applications where other compression algorithms are already in use and have generated large amounts of offline data $\mathcal{D}$, and where online learning may be difficult to implement.

In the digit identification, car shopping and survey, and car racing experiments, we set the compression rate hyperparameter λ (see Section 4) to 0.5 during training. In the photo verification experiments, we set $\lambda = 0.25$ during training.

A.2 Measuring Bitrates

For the digit identification, car racing, and car shopping and survey experiments, we use the following procedure to measure compression rates. To estimate the prior distribution (introduced in Section 4), we fit a multivariate Gaussian distribution to the latent embeddings of the images in our training set. To measure the number of bits needed to encode a given latent embedding, we normalize the latent feature values to their z-scores, discretize the z-scores into bins of width 0.1, and sum the negated base-2 log-probabilities of the discretized values under the prior distribution. For the photo verification experiments, we use the base-2 KL-divergence between the latent posterior and prior in the NVAE model [35].

In the digit identification experiments in Figure 3, we sweep $\lambda \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 1\}$ and measure the resulting bitrates (the hyperparameter λ is defined in Section 4). In the car shopping experiments in Figure 4, we sweep $\lambda \in \{0, 0.375, 0.5, 0.625, 1\}$. In the car survey experiments in Figure 4, we sweep $\lambda \in \{0, 0.25, 0.5, 0.75, 1\}$. In the photo verification experiments in Figure 5, we sweep $\lambda \in \{0, 0.25, 0.375, 0.5, 0.625, 0.75, 1\}$. In the car racing experiments in Figure 6, we set $\lambda = 0.5$.

A.3 Network Architectures and Training

We use stochastic gradient descent – in particular, Adam [51] – to perform the optimization steps in Algorithm 1.

In the car racing and digit identification experiments, we use a feedforward network with 2 layers of 256 units to represent the discriminators; to represent the compression model, the same architecture, but with 64 instead of 256 units. In the car shopping and survey experiments, we use the same architecture, but with 64 instead of 256 units, for the discriminators. In the photo verification experiment, we combine the convolutional network architecture from `https://github.com/yzwxx/vae-celebA/blob/master/model_vae.py` with 2 additional fully-connected layers of 256 units to represent the discriminators and the compression model.

A.4 Compressing Images using a Generative Model

Following up on the discussion in Section 4 about structuring the compression model $f_\theta(\hat{\mathbf{z}}|\mathbf{z})$, let $\hat{\mathbf{z}} = [\mathbf{z}_1\ \mathbf{z}_2]$ denote the decomposition of $\hat{\mathbf{z}}$ into masked features $\mathbf{z}_1$ and transmitted features $\mathbf{z}_2$. In the digit identification, photo verification, and car racing experiments, we set the masked features to follow the distribution $\mathbf{z}_1 \sim \mathcal{N}(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\Sigma}})$, where

$$\bar{\boldsymbol{\mu}} = \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\mathbf{z}_2 - \boldsymbol{\mu}_2),$$
$$\bar{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}.$$

We estimate $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ empirically, from the data used to train the generative model. In the car racing experiments, at each timestep t, we set the prior mean $\boldsymbol{\mu}$ to the transmitted feature values at the previous timestep $t-1$, and estimate the prior covariance $\boldsymbol{\Sigma}$ empirically from state transition data. In the car shopping and survey experiments, we sample the masked features from the StyleGAN2 prior – i.e., by feeding Gaussian noise input to the StyleGAN2 mapping network, and computing the intermediate latents $\mathbf{w}$.

To make exploration easier (see Section 4 for discussion), we reduce the dimensionality d of the mask output space by grouping together consecutive latent features. In the car racing experiments, we train a VAE with 32 latent features using prior methods [52], and reduce the dimensionality of the compression model's output space from 32 to $d = 8$ by creating 8 groups of 4 latent features each – where group 1 contains latent features 1-4, group 2 contains features 5-8, etc. – and masking groups instead of masking individual features. In the car shopping experiments, we use a StyleGAN2 model with 16 style layers – trained on the LSUN Car dataset using prior methods [34] – and reduce the dimensionality to $d = 8$ by dividing the 16 style layers into 8 groups. In the car survey experiments, we reduce to $d = 4$ using the same method. To encode images into the StyleGAN2 latent space, we use the optimization-based projection method described in Section 5 of [34]. In the photo verification experiments, we use the NVAE model for CelebA 64x64 described in Table 6 of [35]. We always sample the latents in the second and third scales from the prior. For the latents in the first scale, we reduce the dimensionality of the mask output space from $d = 5 \cdot 8^2$ to $d = 8$ by applying the same mask to all 5 groups, and dividing the 64 latents into groups of 8. In the digit identification experiments, we use a β-VAE with 10 latent features, which we do not group together as in the other experiments.

In the digit identification, car shopping and survey, and car racing experiments, we use the latent embedding $\mathbf{z}$ instead of the full image $\mathbf{x}$ as input to the discriminators – i.e., we set $D_\phi(\mathbf{a}, \mathbf{x}) \leftarrow D_\phi(\mathbf{a}, \mathbf{z})$ and $D_\psi(\boldsymbol{p}, \mathbf{x}) \leftarrow D_\psi(\boldsymbol{p}, \mathbf{z})$.

A.5 Positive Examples for Discriminator Training

In the digit identification experiments, we treat 63,000 labeled images from the MNIST training set as positive examples of user behavior without compression. In the photo verification experiments, we do the same with 202,397 examples from the labeled CelebA training set – in particular, the Eyeglasses and Hat labels. In the car shopping experiments, we automatically label the Ferrari, Bugatti, McLaren, Aston Martin, Lamborghini, Spyker, and Porsche categories as unaffordable, and the Wagon, Minivan, and Van categories as affordable, discard images belonging to any other categories, and treat 1,507 of the remaining labeled images as positive examples. In the car survey experiments, we collect positive examples by eliciting 1,507 binary labels of "dark-colored" vs. "light-colored" on Amazon Mechanical Turk.

A.6 Prompts for Amazon Mechanical Turk Participants

For the photo verification experiment in Figure 5 in which users check if eyes are covered:

> In this task, you will examine photos of people and check if their eyes are covered. Photos of people wearing eyeglasses or sunglasses should be classified as covered. Choose the appropriate label that best suits the image: 'Eyes are not covered' or 'Eyes are covered'.

For the photo verification experiment in Figure 5 in which users check if heads are covered:

> In this task, you will examine photos of people and check if their head is covered. Photos of people wearing hats or caps should be classified as covered. Choose the appropriate label that best suits the image: 'Head is not covered' or 'Head is covered'.

For the car shopping experiments in Figure 4:

> In this task, you will examine photos of cars and guess if they are affordable for someone with a budget of approximately $20,000. Choose the appropriate label that best suits the image: ‘Affordable’ or ’Not affordable’.

For the car survey experiments in Figure 4:

> In this task, you will examine photos of cars and determine if they are dark-colored (black, dark blue, dark red, etc.) or light-colored (white, silver, light red, yellow, etc.). Choose the appropriate label that best suits the image: ‘Dark-colored car’ or ’Light-colored car’.

For the handwritten digit identification experiments in Figure 3:

> Choose the appropriate label that best suits the image: 0, 1, 2, 3, 4, 5, 6, 7, 8, or 9

A.7 Subject Allocation

Amazon Mechanical Turk experiments. In the car shopping and survey tasks, we assigned 10 users to label each of the 100 held-out images, in order to reduce the variance introduced by our intentionally-vague prompts (see previous section). For the other AMT experiments, we only assigned one user to each image, since we found that behavior did not vary substantially across users.

Car racing video game experiment. We recruited 10 male and 2 female participants, with an average age of 25. Each participant was provided with the rules of the game and played 5 practice episodes to familiarize themselves with the controls. To generative positive and negative examples for training the PICO discriminator, we had a pilot user play 10 episodes without compression and 15 episodes with a compression model that outputs uniform-random mask probabilities. Each of the 12 participants played in both experimental conditions: with the non-adaptive compression baseline, and with the trained compression model from PICO. To avoid the confounding effect of users learning to play the game better over time, we counterbalanced the order of the two conditions. Each condition lasted 15 episodes, with 100 timesteps (10 seconds) per episode.

B Subjective Evaluations in Car Racing Experiment

After evaluating the non-adaptive compression baseline:

> It was quite hard to understand where the car/road were when the video got hazy
>
> by the end of the training, the delay felt less significant. I almost didn’t notice it. I had difficulty figuring out what the environment wanted me to do when it would bend the road far ahead of me but not near me.
>
> It was often hard to tell if the car was moving or not, and the road sometimes disappeared, which also made it hard to tell when steering was needed
>
> Often the task wasn’t too hard, but it was most challenging when the scene geometry would suddenly shift and I couldn’t anticipate how to react with my controls.

After evaluating PICO:

> It was a lot more predictable and the blur was very infrequent. The road did behave pretty unpredictably sometimes and I could not control
>
> 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.
>
> After model training much easier than before model training
>
> This time around the task was a lot easier – the fact that the scene geometry changed more naturally, and the fact that the effects of any delayed actions were predictable, made it easier to decide how to steer

Table 1: Subjective Evaluations in the Car Racing User Study

	p-value	Non-Adaptive	PICO
I was able to keep the car on the road	$< .0001$	3.45	**5.64**
I could anticipate the consequences of my steering actions	$< .01$	3.82	**5.36**
I could tell when the car was about to go off road	$< .01$	3.55	**5.36**
I could tell when I needed to steer to keep the car on the road	$< .01$	4.09	**5.73**
I was often able to determine the car's current position	$< .001$	4.00	**5.82**

Means reported for responses on a 7-point Likert scale, where 1 = Strongly Disagree, and 7 = Strongly Agree. p-values from a one-way repeated measures ANOVA with the use of PICO as a factor influencing responses.

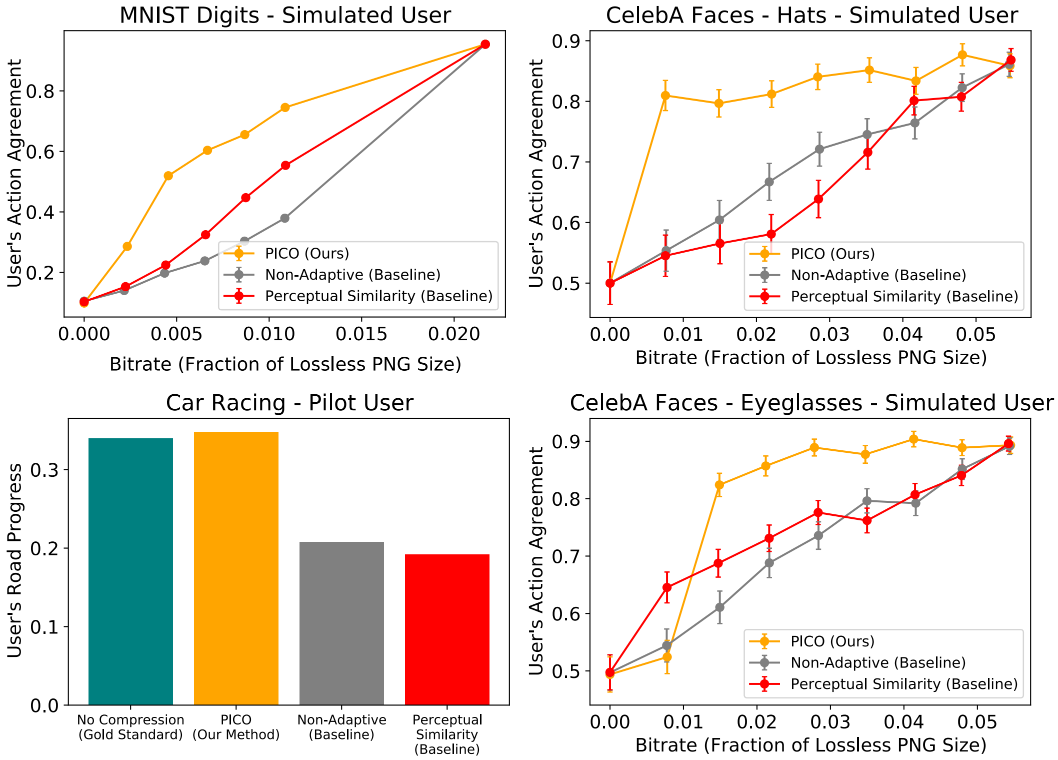

Figure 7: Experiments with simulated and pilot users show that PICO outperforms both baselines, and that the perceptual similarity baseline only performs better than the non-adaptive baseline on the digit identification task (top left).

C Simulation Experiments

To determine which baseline methods to compare with PICO in the user studies in Section 5, we ran preliminary experiments in which we simulated user behavior. In the digit identification and photo verification tasks, we simulated the user's policy by training a classifier on labeled data (see Appendix A.5 for a description of the labeled data in each domain). In the car shopping, car surveying, and car racing tasks, we did not have enough labeled data to train a policy that qualitatively matched real user behavior. Hence, in the car racing task, we conducted a small-scale experiment with a single pilot user; and in the car shopping and surveying tasks, we perform a qualitative analysis of compressed image samples.

Figure 7 shows that PICO outperformed both the non-adaptive and perceptual similarity baselines in all domains. Furthermore, the perceptual similarity baseline only performed better than the non-adaptive baseline in the digit identification task; hence, our decision to omit the perceptual similarity baseline from the other user studies in Section 5. Figure 8 shows that, while PICO learns to preserve the perceived price of the car in the shopping task, and to preserve the color of the car in the survey task, the perceptual similarity baseline does not preserve either of the two features.

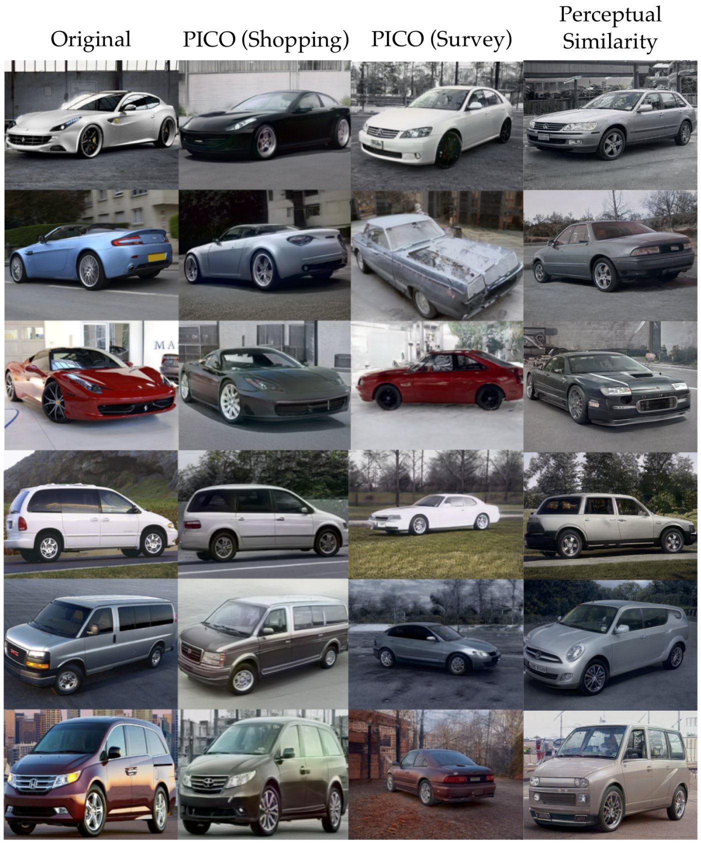

Figure 8: While PICO learns to preserve the perceived price of the car in the shopping task (second column), and to preserve the color of the car in the survey task (third column), the perceptual similarity baseline fails to preserve either of the two features (fourth column).

D Failure Cases

There are several ways in which PICO can fail to match the user's actions with and without compression. For example, the latent embedding $\mathbf{z}$ produced by the pre-trained generative model (see Section 4) may lack the necessary features for performing the downstream task: the yellow sports car in rows 7-8 of Figure 10 gets distorted when encoded into the StyleGAN2 latent space, even without any additional compression. Another failure mode is for the latent features to be entangled, causing the structured mask output space of the compression model (see Section 4) to be insufficiently expressive for learning an effective compression policy: many of the compressed faces in Figure 9 are visually distorted, most likely because the true prior distribution over latent embeddings is not modeled accurately by a Gaussian (see Appendix A.4).

E Examples of Compression at Different Bitrates

Figures 9 and 10 show that PICO tends to preserve task-relevant features like digit number, eyeglasses and hats, and the price and color of a car, more often than the non-adaptive baseline, and especially at lower bitrates. As the bitrate decreases, PICO discards task-irrelevant features before discarding task-relevant features. At extremely low bitrates (e.g., zero), PICO gracefully degrades to sampling a random image from the pre-trained generative model (see the right-most columns in Figures 9 and 10), instead of, e.g., transmitting a heavily-distorted image with visual artifacts that make it difficult for the user to even attempt to perform their task.

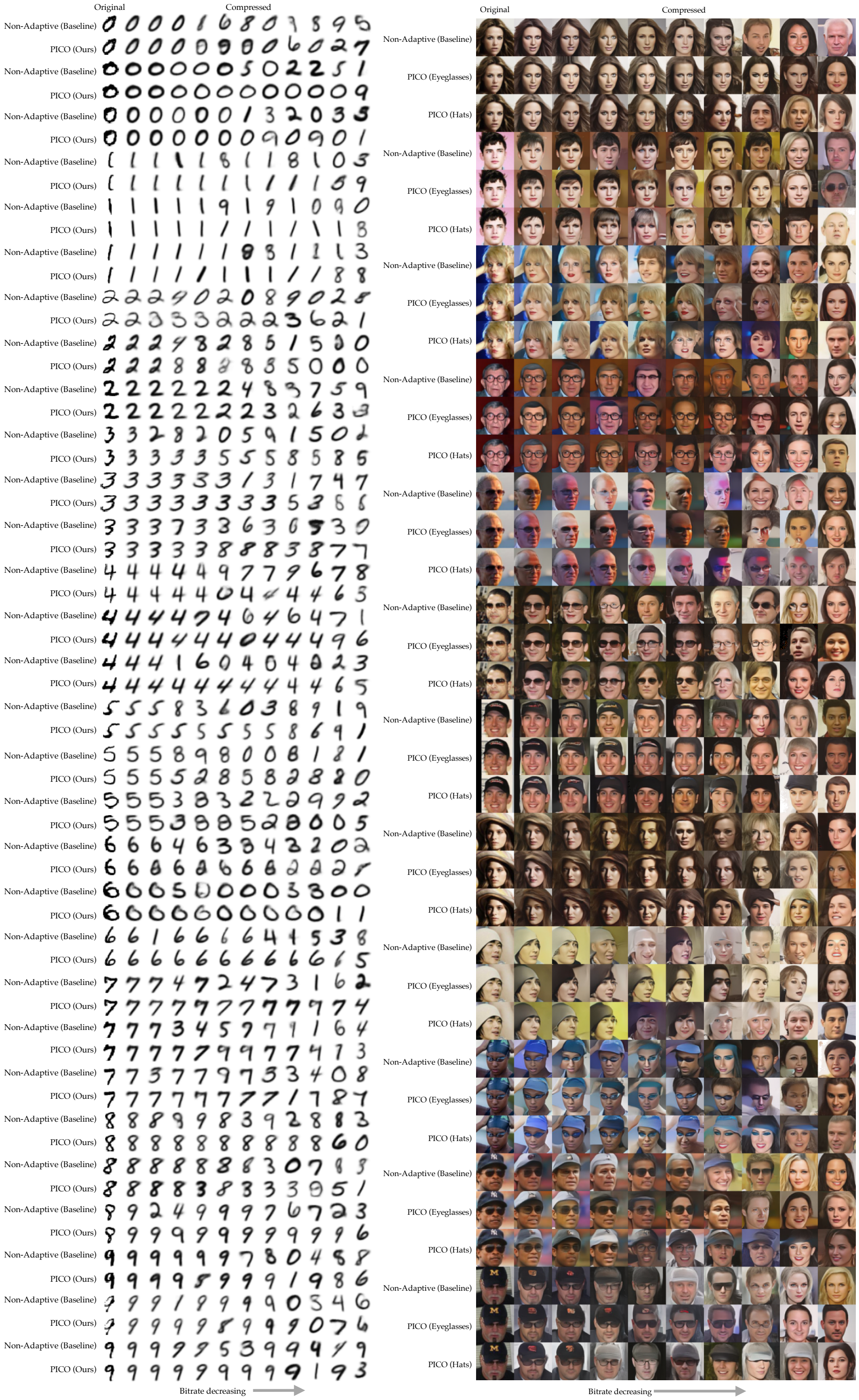

Figure 9: Additional samples from the non-adaptive compression baseline and PICO drawn at varying bitrates. Left: digit identification experiments from Section 5.1 and Figure 3. Right: photo verification experiments from Section 5.2 and Figure 5.

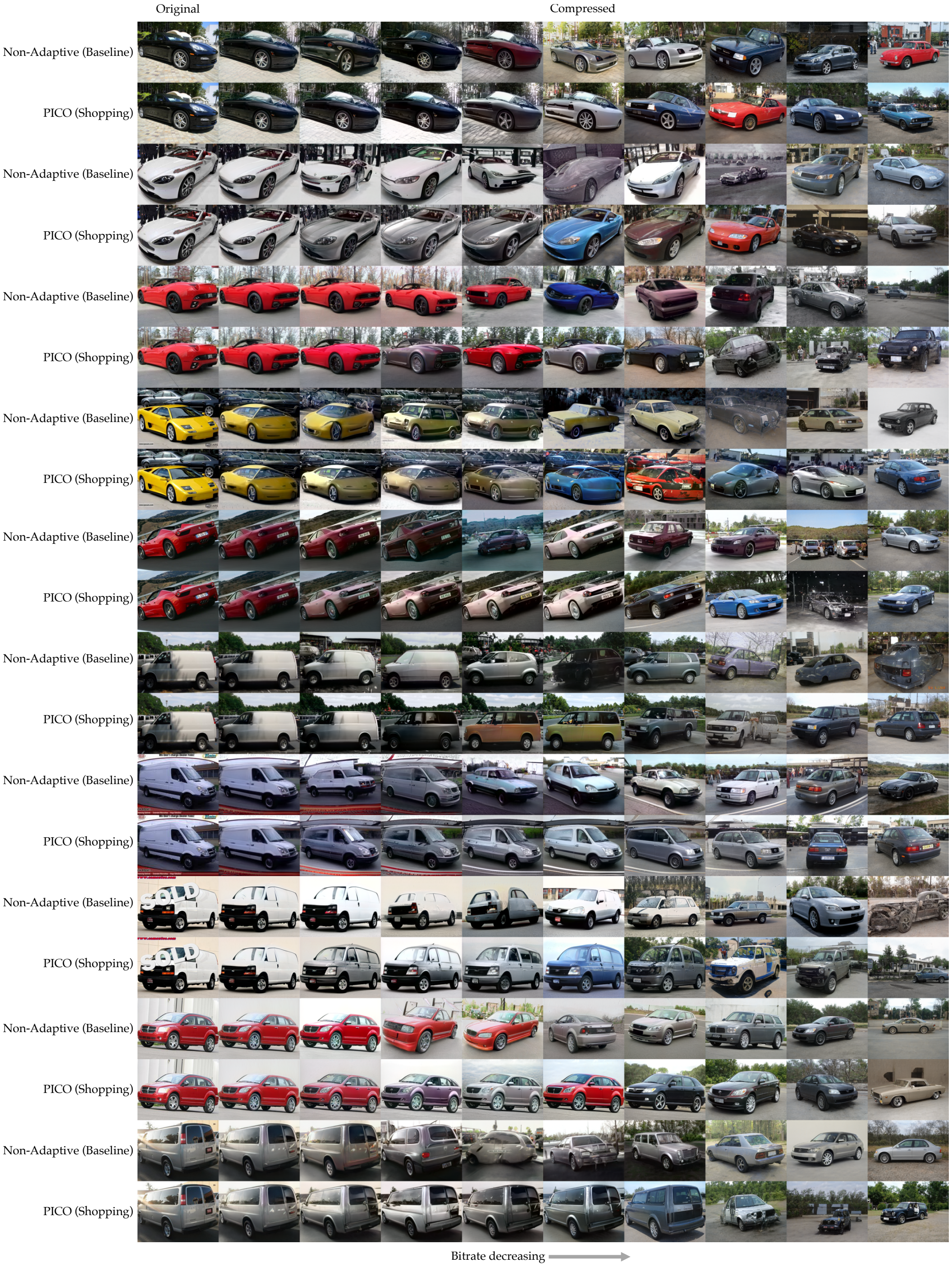

Figure 10: Additional samples from the non-adaptive compression baseline and PICO drawn at varying bitrates, for the car shopping experiments in Section 5.1 and Figure 4.