Multimodal Prediction and Personalization of Photo Edits with Deep Generative Models

Ardavan Saeedi
CSAIL, MIT

Matthew D. Hoffman
Google Brain

Stephen J. DiVerdi
Adobe Research

Asma Ghandeharioun
Media Lab, MIT

Matthew J. Johnson
Google Brain

Ryan P. Adams
Princeton and Google Brain

Abstract

Professional-grade software applications are powerful but complicated—expert users can achieve impressive results, but novices often struggle to complete even basic tasks. Photo editing is a prime example: after loading a photo, the user is confronted with an array of cryptic sliders like “clarity”, “temp”, and “highlights”. An automatically generated suggestion could help, but there is no single “correct” edit for a given image—different experts may make very different aesthetic decisions when faced with the same image, and a single expert may make different choices depending on the intended use of the image (or on a whim). We therefore want a system that can propose multiple diverse, high-quality edits while also learning from and adapting to a user’s aesthetic preferences. In this work, we develop a statistical model that meets these objectives. Our model builds on recent advances in neural network generative modeling and scalable inference, and uses hierarchical structure to learn editing patterns across many diverse users. Empirically, we find that our model outperforms other approaches on this challenging multimodal prediction task.

1 INTRODUCTION

Many office workers spend most of their working days using pro-oriented software applications. These applications are often powerful, but complicated. This complexity may overwhelm and confuse novice users, and even expert users may find some tasks time-consuming and repetitive. We want to use machine learning and statistical modeling to help users manage this complexity.

Fortunately, modern software applications collect large amounts of data from users with the aim of providing them with better guidance and more personalized experiences. A photo-editing application, for example, could use data about how users edit images to learn what kinds of adjustments are appropriate for what images, and could learn to tailor its suggestions to the aesthetic preferences of individual users. Such suggestions can help both experts and novices: experts can use them as a starting point, speeding up tedious parts of the editing process, and novices can quickly get results they could not have otherwise achieved.

Several models have been proposed for predicting and personalizing user interaction in different software applications. These existing models are limited in that they only propose a single prediction or are not readily personalized. Multimodal predictions[1] are important in cases where, given an input from the user, there could be multiple possible suggestions from the application. For instance, in photo editing/enhancement, a user might want to apply different kinds of edits to the same photo depending on the effect he or she wants to achieve. A model should therefore be able to recommend multiple enhancements that cover a diverse range of styles.

In this paper, we introduce a framework for multimodal prediction and personalization in software applications. We focus on photo-enhancement appli-

Proceedings of the 21^{st} International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. PMLR: Volume 84.

[1]We mean “multimodal” in the statistical sense (*i.e.,* coming from a distribution with multiple maxima), rather than in the human-computer-interaction sense (*i.e.,* having multiple modes of input or output).

(a) (b)

Figure 1: The main goals of our proposed models: (a) **Multimodal photo edits**: For a given photo, there may be multiple valid aesthetic choices that are quite different from one another. (b) **User categorization**: A synthetic example where different user clusters tend to prefer different slider values.

cations, though our framework is also applicable to other domains where multimodal prediction and personalization is valuable. Fig. 1 demonstrates our high-level goals: we want to learn to propose diverse, high-quality edits, and we want to be able to personalize those proposals based on users' historical behavior.

Our modeling and inference approach is based on the variational autoencoder (VAE) (Kingma and Welling, 2013). We propose an extension of the VAE which uses a hierarchical structure to learn styles across many diverse users. We further extend our model to provide personalized results, learning each user's personal style from their historical behavior. We introduce a novel encoder architecture that, for each user, analyzes each edit independently, and then combines the results in a symmetric, exchangeable way that extends to any number of user edits.

We apply our framework to three different datasets (collected from novice, semi-expert, and expert users) of image features and user edits from a photo-enhancement application and compare its performance qualitatively and quantitatively to various baselines. We demonstrate that our model outperforms other approaches.

2 BACKGROUND AND RELATED WORK

In this section, we first briefly review the VAE framework that our model is built upon; next, we provide an overview of the available models for predicting photo edits and summarize their pros and cons.

2.1 Variational Autoencoder (VAE)

The VAE, introduced by Kingma and Welling (2013), has been successfully applied to various models with continuous latent variables and a complicated likelihood function (*e.g.,* a neural network with nonlinear hidden layers). In these settings, posterior inference is typically intractable, and even approximate inference may be prohibitively expensive to run in the inner loop of a learning algorithm. The VAE allows this difficult inference to be amortized over many learning updates, making each learning update cheap even with complex likelihood models.

As an instance of such models, consider modeling a set of N i.i.d. observations $y = \{y_n\}_{n=1}^N$ with the following generative process: $z_n \overset{\text{iid}}{\sim} h$ and $y_n \sim f(g_\theta(z_n))$, where z_n is a latent variable generated from a prior $h(z)$ (*e.g.,* $\mathcal{N}(0, I)$) and the likelihood function $p_\theta(y_n|z_n) = f(y_n; g_\theta(z_n))$ is a simple distribution f whose parameters $g_\theta(z_n)$ can be a complicated function of z_n. For example, $p_\theta(y_n|z_n)$ might be $\mathcal{N}(y_n; \mu(z_n;\theta), \Sigma(z_n;\theta))$ where the mean and the covariance depend on z_n through a multi-layer perceptron (MLP) richly parameterized by weights and biases θ.

In the VAE framework, the posterior density $p_\theta(z|y)$ is approximated by a recognition network $q_\phi(z|y)$, which can take the form of a flexible conditional density model such as an MLP parameterized by ϕ. To learn the parameters of the likelihood function θ and the recognition network ϕ, the following lower bound on the marginal likelihood is maximized:

$$\mathcal{L}_{\text{VAE}}(\phi, \theta) \triangleq \mathbb{E}_{q_\phi(z|y)}[\log p_\theta(y|z)] - \text{KL}(q_\phi(z|y)||p(z)).$$

To compute a Monte Carlo estimate of the gradient of this objective with respect to ϕ, Kingma and Welling (2013) propose a reparameterization trick for sampling from $q_\phi(z|y)$ by first sampling from an auxiliary noise variable and then applying a differentiable map to the sampled noise. This yields a differentiable Monte Carlo estimate of the expectation with respect to ϕ. Given the gradients, the parameters are updated by stochastic gradient ascent.

When the prior on z has more structure, this amortized inference approach can be extended to exploit that structure (Johnson et al., 2016; Sønderby et al., 2016). In Section 3.2, we will develop a particularly natural structure-exploiting amortized variational approximation.

2.2 Related Work on the Prediction of Photo Edits

There are two main categories of models, parametric and nonparametric, that have been used for prediction of photo edits:

Parametric methods These methods approximate a parametric function by minimizing a squared (or a similar) loss. The loss is typically squared L_2 distance in Lab color space, which more closely approximates human perception than RGB space (Sharma and Bala, 2002). This loss is reasonable if the goal is to learn from a set of consistent, relatively conservative edits. But when applied to a dataset of more diverse edits, a model that minimizes squared error will tend to predict the *average* edit. At best, this will lead to conservative predictions; in the worst case, the average of several good edits may produce a bad result.

Bychkovsky et al. (2011) collect a dataset of 5000 photos enhanced by 5 different experts; they identify a set of features and learn to predict the user adjustments after training on the collected dataset. They apply a number of regression techniques such as LASSO and show their proposed adjustments can match the adjustments of one of the 5 experts. Their method only proposes a single adjustment and the personalization scheme that they suggest requires the user to edit a set of selected training photos. Yan et al. (2016) use a deep neural network to learn a mapping from an input photo to an enhanced one following a particular style; their results show that the proposed model is able to capture the nonlinear and complex nature of this mapping. More recently, Gharbi et al. (2017) propose a network architecture that is faster in processing every image compared to the model by Yan et al. (2016). Both of these models only propose a single style of adjustment.

Nonparametric methods The few available nonparametric methods are typically able to propose multiple edits or some uncertainty over the range of adjustments. Lee et al. (2015) propose a method that can generate a diverse set of edits for an input photograph. The authors have a curated set of exemplar images in various styles. They use an example-based style-transfer algorithm to transfer the style from an exemplar image to an input photograph. To choose the right exemplar image, they do a semantic similarity search using features that they have learned via a convolutional neural network (CNN). Although their approach can recommend multiple edits to a photo, their edits are destructive; that is, the recommended edits are directly applied to the photo and the user is not able to further customize those edits. Koyama et al. (2016) introduce a model for personalizing photo edits only based on the history of edits by a single user. The authors use a self-reinforcement procedure in which after every edit by a user they: 1) update the distance metric between the user's past photos, 2) update a feature vector representation of the user's photos, and 3) update an enhancement preference model based on the feature vectors and the user's enhancement history. This model requires data collection from a single user and does not benefit from other users' information.

2.3 Related Multimodal Prediction Models

Traditional neural networks using mean squared error (MSE) loss cannot naturally handle multimodal prediction problems, since MSE is minimized by predicting the average response. Neal (1992) introduces stochastic latent variables to the network and proposes training Sigmoid Belief Networks (SBN) with only binary stochastic variables. However, this model is difficult to train, and it can only make piecewise-constant predictions and is therefore not a natural fit to continuous-response prediction problems.

Bishop (1994) proposes mixture density networks (MDN), which are more suitable for continuous data. Instead of using stochastic units, the model directly outputs the parameters of a Gaussian mixture model. The complexity of MDNs' predictive distributions is limited by the number of mixture components. If the optimal predictive distribution cannot be well approximated by a relatively small number of Gaussians, then an MDN may not be an ideal choice.

Tang and Salakhutdinov (2013) add deterministic hidden variables to SBNs in order to model continuous distributions. The authors showed improvements over the SBN; nevertheless, training the stochastic units remained a challenge due to the difficulty of doing approximate inference on a large number of discrete variables. Dauphin and Grangier (2015) propose a new class of stochastic networks called linearizing belief networks (LBN). LBN combines deterministic units with stochastic binary units multiplicatively. The model uses deterministic linear units which act as multiplicative skip connections and allow the gradient to flow without diffusion. The empirical results show that this model can outperform standard SBNs.

3 MODELS

Given the limitations of the available methods for predicting photo edits (described in Section 2.2), our goal is to propose a framework in which we can: 1) recommend a set of diverse, parametric edits based on a labeled dataset of photos and their enhancements, 2) categorize the users based on their style and type of edits they apply, and finally 3) personalize the enhancements based on the user category. We focus on the photo-editing application in this paper, but the proposed framework is applicable to other domains where users must make a selection from a large, richly parameterized design space where there is no single right answer (for example, many audio processing algorithms have large numbers of user-tunable parameters).

Our framework is based on VAEs and follows a mixture-of-experts design (Murphy, 2012, Section 11.2.4). We first introduce a conditional VAE that can generate diverse set of enhancements to a given photo. Next, we extend the model to categorize the users based on their adjustment style. Our model can provide interpretable clusters of users with similar style. Furthermore, the model can provide personalized suggestions by first estimating a user's category and then suggesting likely enhancements conditioned on that category.

3.1 Multimodal Prediction with Conditional Gaussian Mixture VAE (CGM-VAE)

Given a photo, we are interested in predicting a set of edits. Each photo is represented by a feature vector x_n and its corresponding edits y_n are represented by a vector of slider values (*e.g.,* contrast, exposure, saturation, etc.). We assume that there are L clusters of possible edits for each image. To generate the sliders y_n for a given image x_n, we first sample a cluster assignment s_n and a set of latent features z_n from its corresponding mixture component $\mathcal{N}(\mu_{s_n}, \Sigma_{s_n})$. Next, conditioned on the image and z_n, we sample the slider values. The overall generative process for the slider values $\{y_n\}_{n=1}^N$ conditioned on the input images $\{x_n\}_{n=1}^N$ is

$$s_n|\pi \overset{\text{iid}}{\sim} \pi, \qquad z_n|s_n, \{\mu_\ell, \Sigma_\ell\}_{\ell=1}^L \sim \mathcal{N}(\mu_{s_n}, \Sigma_{s_n}),$$
$$y_n|x_n, z_n, \theta \sim \mathcal{N}(\mu(z_n, x_n; \theta), \Sigma(z_n, x_n; \theta)), \quad (1)$$

where $\mu(z_n, x_n; \theta)$ and $\Sigma(z_n, x_n; \theta)$ are flexible parametric functions, such as MLPs, of the input image features x_n concatenated with the latent features z_n. Summing over all possible values for the latent variables s_n and z_n, the marginal likelihood $p(y_n|x_n) = \sum_{s_n} \int_{z_n} p(y_n, s_n, z_n|x_n) dz_n$ yields complex, multimodal densities for the image edits y_n.

The posterior $p(s, z|x, y)$ is intractable. We approximate it with variational recognition models as

$$p_\theta(s, z|x, y) \approx q_{\phi_s}(s|x, y) q_{\phi_z}(z|x, y, s). \qquad (2)$$

Note that this variational distribution does not model s and z as independent. For $q_{\phi_s}(s|x, y)$, we use an MLP with a final softmax layer, and for $q_{\phi_z}(z|x, y, s)$, we use a Gaussian whose mean and covariance are the output of an MLP that takes s, x, and y as input. Fig. 2 (parts a and b) outlines the graphical model structures of the CGM-VAE and its variational distributions q_ϕ.

Given this generative model and variational family, to perform inference we maximize a variational lower bound on $\log p_\theta(y|x)$, writing the objective as

$$\begin{aligned}\mathcal{L}(\theta, \phi) \triangleq \; & \mathbb{E}_{q_\phi(s,z|x,y)}[\log p_\theta(y|z, x)] \\ & - \mathrm{KL}(q_\phi(s, z|x, y)||p_\theta(s, z)).\end{aligned}$$

By marginalizing over the latent cluster assignments s, the CGM-VAE objective can be optimized using stochastic gradient methods and the reparameterization trick. Marginalizing out the discrete latent variables is not computationally intensive since s and y are conditionally independent given z, $p_\theta(s, z)$ is cheap to compute relative to $p_\theta(y|x, z)$, and we use a relatively small number of clusters. However, with a very large discrete latent space, one could use alternate approaches such as the Gumbel-Max trick (Maddison et al., 2016) or REBAR (Tucker et al., 2017).

3.2 Categorization and Personalization

In order to categorize the users based on their adjustment style, we extend the basic CGM-VAE model to a hierarchical model that clusters users based on the edits they make. We call this new model personalized VAE or *P-VAE* for short. While the model in the previous section considered each image-edit pair x_n, y_n in isolation, we now organize the data according to U distinct users, using x_{un} to denote the nth image of user u and y_{un} to denote the corresponding slider values (see Fig. 2(c)). N_u denotes the number of photos edited by user u. As before, we assume a GMM with L components $\{\mu_\ell, \Sigma_\ell\}_{\ell=1}^L$ and mixing weights π, but here these clusters will model differences between users.

For each user u we sample a cluster index s_u to indicate the user's category, then for each photo $n \in \{1, \ldots, N_u\}$ we sample the latent attribute vector z_{un} from the corresponding mixture component:

$$s_u|\pi \overset{\text{iid}}{\sim} \pi, \quad z_{un}|s_u, \{(\mu_\ell, \Sigma_\ell)\}_{\ell=1}^L \overset{\text{iid}}{\sim} \mathcal{N}(\mu_{s_u}, \Sigma_{s_u}).$$

Finally, we use the latent features z_{un} to generate the vector of suggested slider values y_{un}. As before, we use a multivariate normal distribution with mean and variance generated from an MLP parameterized by θ:

$$y_{un}|x_{un}, z_{un}, \theta \overset{\text{iid}}{\sim} \mathcal{N}(\mu(z_{un}, x_{un}; \theta), \Sigma(z_{un}, x_{un}; \theta)).$$

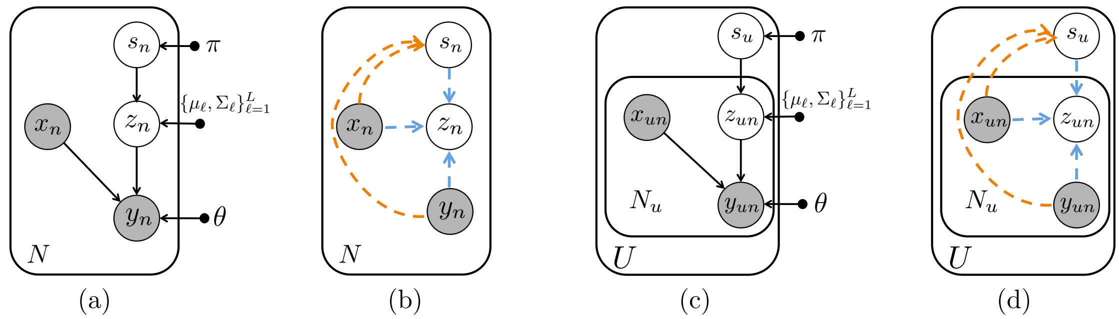

Figure 2: (a) The graphical model for CGM-VAE introduced in Section 3.1 (b) The dependency structure for the variational approximations $q_{\phi_s}(s|x, y)$ and $q_{\phi_z}(z|x, y, s)$ in CGM-VAE (c) The P-VAE model introduced in Section 3.2) for categorization and personalization. There are U users and each user u has N_u photos. (d) The dependency structure in the variational distributions for the P-VAE model. Note that the recognition network for s_u depends on all the images and their corresponding slider values of user u.

For inference in the P-VAE model, our goal is to maximize the following variational lower bound (VLB) for a dataset of U users:

$$\mathcal{L}(\theta, \phi) \triangleq \frac{1}{U} \sum_u \Big[\sum_{n=1}^{N_u} \mathbb{E}_{q_\phi(z,s|x,y)}[\log p_\theta(y_{un}|x_{un}, z_{un})] - \mathrm{KL}(q_\phi(z_{un}|x_{un}, y_{un}, s_u)||p_\theta(z_{un}|s_u))\Big] - \mathrm{KL}(q_\phi(s_u|\{x_{un}, y_{un}\}_{n=1}^{N_u})||p(s_u)).$$

In the following we define the variational factors and the recognition networks that we use.

Variational factors For the local variables z and s, we restrict $q(z|s)$ to be normal and we have $q(s)$ in the categorical form. As in the CGM-VAE, we marginalize over cluster assignments at the user level. Fig. 2 (parts c and d) outlines the graphical model structures of the P-VAE and its variational distributions q_ϕ.

For the variational factor of the latent mixture component index s_u, we write:

$$q(s_u|\{x_{un}, y_{un}\}_{n=1}^{N_u}; \phi) \propto \exp\left\{ \left\langle \log \pi + \textstyle\sum_{n=1}^{N_u} \log r(y_{un}, x_{un}; \phi),\ t_s(s_u) \right\rangle \right\},$$

where $t_s(s_u)$ denotes the one-hot vector encoding of s_u and $r(y_{un}, x_{un}; \phi)$ is the recognition network that belongs to some parametrized class of functions. That is, for each user image x_{un} and corresponding set of slider values y_{un}, the recognition network produces a potential over the user's latent mixture component s_u. These image-by-image guesses are then combined with each other and with the prior to produce the inferred variational factor on s_u.

This recognition network architecture is both natural and convenient. It is natural because a powerful enough r can set $r_k(y_{un}, x_{un}; \phi) \propto p_\theta(y_{un}|x_{un}, s_u = k)$, in which case $q_\phi(s_u|\{x_{un}, y_{un}\}_{n=1}^{N_u}) \equiv p(s_u|\{x_{un}, y_{un}\}_{n=1}^{N_u})$ and there is no approximation error. It is convenient because it analyzes image-edit pairs independently, and these evidence potentials are combined in a symmetric, exchangeable way that extends to any number of user images N_u.

4 EXPERIMENTS

We evaluate our models and several strong baselines on three datasets. We focus on the photo editing software Adobe Lightroom. The datasets that we use cover three different types of users that can be roughly described as 1) casual users who do not use the application regularly, 2) frequent users who have more familiarity with the application and use it more frequently 3) experts who have more experience in editing photos than the other two groups. We randomly split all three datasets into 10% test, 10% validation, and 80% train set. For more details on the datasets, baselines and hyperparameter settings, see the supplementary materials.

Datasets The casual users dataset consists of 345000 images along with the slider values that a user has applied to the image in Lightroom. There are 3200 users in this dataset. Due to privacy concerns, we only have access to the extracted features from a CNN applied to the images. Hence, each image in the dataset is represented by a 1024-dimensional vector. For the possible edits to the image, we only focus on 11 basic sliders in Lightroom. Many common editing tasks boil down to adjusting these sliders. The 11 basic sliders have different ranges of values, so we standardize them to all have a range between -1 and 1 when training the model.

The frequent users dataset contains 45000 images (in the form of CNN features) and their corresponding slider values. There are 230 users in this dataset.

Table 1: **Quantitative results:** *LL*: Predictive log-likelihood for our model CGM-VAE and the three baselines. We present the best model in terms of the VLB of the validation dataset. The optimal number of mixture components for all 3 datasets is 3 (choosing from $\{1, 3, 5, 10\}$). The predictive log-likelihood is calculated over the test sets from all three datasets. *JSD*: Jensen-Shannon divergence between normalized histograms of the true sliders and our model predictions over the test sets (lower is better). See Fig. 3 for an example of these histograms. *LAB*: LAB error between the images retouched by the experts and the images retouched by the model predictions. For each image we generate 3 proposals and compare that with the images generated by the top 3 active experts in the experts dataset.

Dataset	**Casual**		**Frequent**		**Expert**		
Eval. Metric	LL	JSD	LL	JSD	LL	JSD	LAB
MLP	-15.71 ± 0.21	0.26 ± 0.04	-2.72 ± 0.31	0.11 ± 0.02	-4.28 ± 0.12	0.22 ± 0.06	7.81 ± 0.26
LBN	-7.12 ± 0.15	0.14 ± 0.02	-3.7 ± 0.43	0.13 ± 0.02	-4.89 ± 0.24	0.17 ± 0.04	7.44 ± 0.29
MDN	-14.53 ± 0.25	0.31 ± 0.06	-1.67 ± 0.47	0.24 ± 0.08	-4.91 ± 0.07	0.28 ± 0.11	8.41 ± 0.27
CGM-VAE	$\mathbf{-6.39 \pm 0.11}$	$\mathbf{0.10 \pm 0.02}$	$\mathbf{-1.42 \pm 0.18}$	$\mathbf{0.08 \pm 0.02}$	$\mathbf{-2.6 \pm 0.15}$	$\mathbf{0.12 \pm 0.05}$	$\mathbf{6.72 \pm 0.27}$

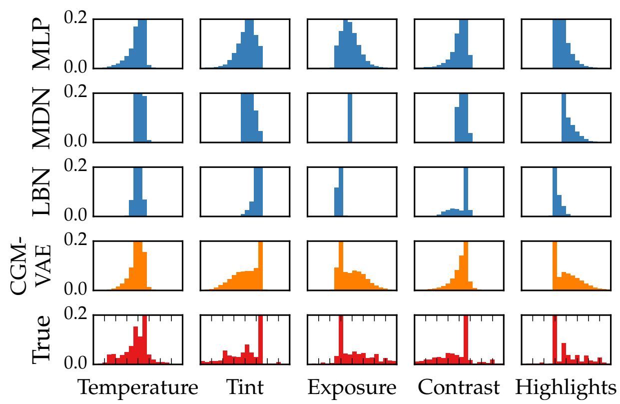

Figure 3: Marginal statistics for the prediction of the sliders in the casual users dataset (test set). Due to space limitations, We only display the top 5 most-used sliders in the dataset. LBN has limited success compared to CGM-VAE. MLP mostly concentrates around the mean edit. The quantitative comparison between different methods in terms of the distance between normalized histograms is provided in Table 1.

These users generally apply more changes to their photos compared to the users in the casual group.

Finally, the expert users dataset (Adobe-MIT5k, collected by Bychkovsky et al. 2011) contains 5000 images and edits applied to these images by 5 different experts, for a total of 25000 edits.

We augment this dataset by creating new images after applying random edits to the original images. To generate a random edit from a slider, we add uniform noise from a range of $\pm 10\%$ of the total range of that slider. Given the augmented set of images, we extract the "FC7" features of a VGG-16 (Simonyan and Zisserman, 2014) pretrained network and use the 4096-dimensional feature vector as a representation of each image in the dataset. After augmenting the dataset, we have 15000 images and 75000 edits in total. Similar to other datasets, we only focus on the basic sliders in Adobe Lightroom.

Baselines We compare our model for multimodal prediction with several models: a multilayer perceptron (MLP), mixture density network (MDN), and linearizing belief network (LBN). The MLP is trained to predict the mean and variance of a multivariate Gaussian distribution; this model will demonstrate the limitations of even a strong model that makes unimodal predictions. The MDN and LBN, which are specifically designed for multimodal prediction, are other baselines for predicting multimodal densities. Table 1 summarizes our quantitative results.

We use three different evaluation metrics to compare the models. The first metric is the predictive log-likelihood computed over a held-out test set of different datasets. Another metric is the Jensen-Shannon divergence (JSD) between normalized histograms of marginal statistics of the true sliders and the predicted sliders. Fig. 3 shows some histograms of these marginal statistics for the casual users.

Finally, we use the mean squared error in the CIE-LAB color space between the expert-retouched image and the model-proposed image. We use the CIE-LAB color space as it is more perpetually linear compared to RGB. We only calculate this error for the experts dataset (test set) since that is the only dataset with available retouched images. To compute this metric, we first apply the predicted sliders from the models to the original image and then convert the generated RGB image to a LAB image. For reference the difference between white and black in CIE-LAB is 100 and photos with no adjustments result in an error of 10.2 . Table 1, shows that our model outperforms the baselines across all these metrics.

Hyperparameters For the CGM-VAE model, we choose the dimension of the latent variable from {2, 20} and the number of mixture components from the set {1, 3, 5, 10}. Note that by setting the number of

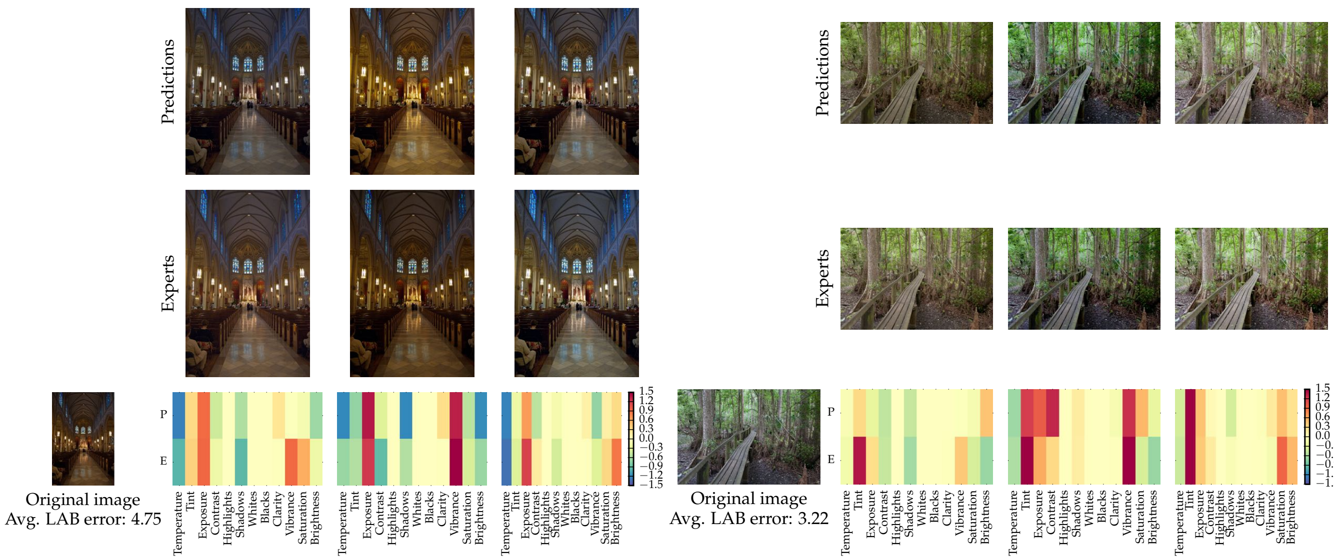

Figure 4: **Multimodal photo edits:** Sample slider predictions from the CGM-VAE model (denoted by P in the figure) compared to the edits of 3 most active experts in the expert users dataset (denoted by E). The images are selected from the test subset of the dataset; the 3 samples are selected from a set of 10 proposals from the CGM-VAE model such that they align with the experts. To show the difference between the model and experts, we apply their sliders to the original image. For images in larger scale and more examples, refer to the supplementary material.

mixture components to 1, CGM-VAE will reduce to a conditional VAE. For the remaining hyperparameters see the supplementary materials. We choose the best hyperparameter setting based on the VLB of the held-out dataset. In all three datasets, 3 mixture components provide the best VLB for the validation datasets.

Tasks In addition to computing the predictive log-likelihood and JSD over the held-out test sets for all three datasets, we consider the following two tasks:

1. Multimodal prediction: We predict different edits applied to the same image by the users in the experts dataset. Our goal is to show that CGM-VAE is able to capture different styles from the experts.
2. Categorizing the users and adapting the predictions based on users' categories: We show that the P-VAE model, by clustering the users, makes better predictions for each user. We also illustrate how inferred user clusters differ in terms of edits they apply to similar images.

4.1 Multimodal Predictions

To show that the model is capable of multimodal predictions, we propose different edits for a given image in the test subset of the experts dataset. To generate these edits, we sample from different cluster components of our CGM-VAE model trained on the experts dataset. For each image we generate 20 different samples and align these samples to the experts' sliders. From the 5 experts in the dataset, 3 propose a more diverse set of edits compared to the others; hence, we only align our results to those three to show that the model can reasonably capture a diverse set of styles.

For each image in the test set, we compare the predictions of MLP, LBN, MDN and the CGM-VAE with the edits from the 3 experts. In MLP (and also MDN), we draw 20 samples from the Gaussian (mixture) distribution with parameters generated from the MLP (MDN). For the LBN, since the network has stochastic units, we directly sample 20 times from the network. We align these samples to the experts' edits and find the LAB error between the expert-retouched image and the model-proposed image.

To report the results, we average across the 3 experts and across all the test images. The LAB error in Table 1 indicates that CGM-VAE model outperforms other baselines in terms of predicting expert edits. Some sample edit proposals and their corresponding LAB errors are provided in Fig. 4. This figure shows that the CGM-VAE model can propose a diverse set of edits that is reasonably close to those of experts. For further examples see the supplementary material.

4.2 Categorization and Personalization

Next, we demonstrate how the P-VAE model can leverage the knowledge from a user's previous edits and propose better future edits. For the users in the test sets of all three datasets, we use between 0 and 30 image-slider pairs to estimate the posterior of each user's

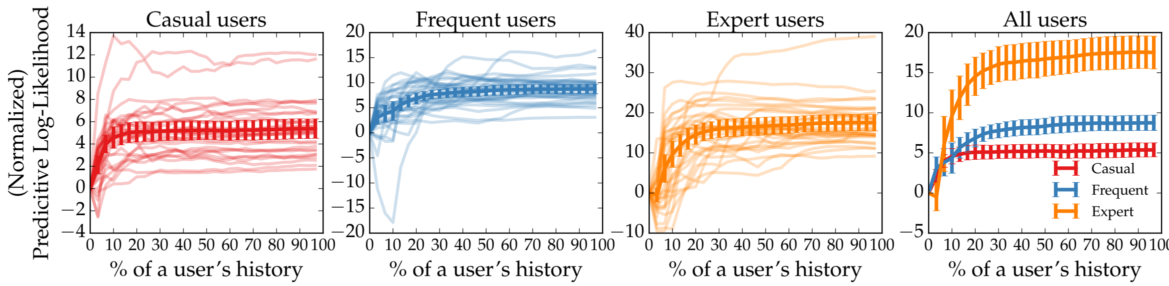

Figure 5: Predictive log-likelihood for users in the test set of different datasets. For each user in the test set, we compute the predictive log-likelihood of 20 images, given 0 to 30 images and their corresponding sliders from the same user. 30 sample trajectories and the overall average ± s.e. is shown for casual, frequent and expert users. The figure shows that knowing more about the user (up to around 10 images) can increase the predictive log-likelihood. The log-likelihood is normalized by subtracting off the predictive log-likelihood computed given zero images. Note the different y-axis in the plots. The rightmost plot is provided for comparing the average predictive log-likelihood across datasets.

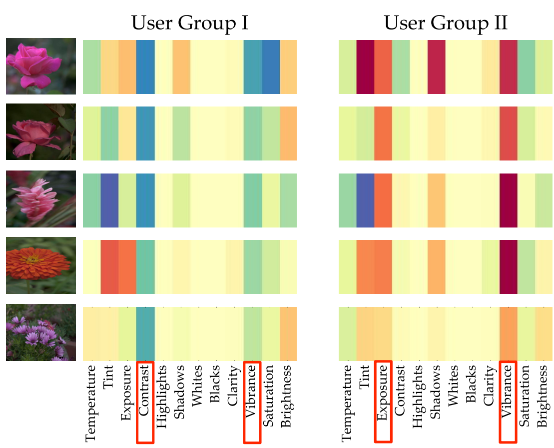

Figure 6: **User categorization:** An example of sample edits for two different user groups which the P-VAE model has identified (in the experts dataset). For similar flower photos, users in group I prefer to use low contrast and vibrance, whereas group II users tend to increase the exposure and vibrance from their default values. There is also group III users which do not show any specific preference for similar flower photos. For results on group III and more examples, see the supplementary materials.

cluster membership. We then evaluate the predictive log-likelihood for 20 other slider values conditioned on the images and the inferred cluster memberships.

Fig. 5 depicts how adding more image-slider combinations generally improves the predictive log-likelihood. The log-likelihood is normalized by subtracting off the predictive log-likelihood computed given zero images. The effect of adding more images is shown for 30 different sampled users; the overall average for the test dataset is also shown in the figure. To compare how various datasets benefit from this model, the average values from the 3 datasets are overlaid. According to Fig. 5, the frequent users benefit more than the casual users and the expert users benefit the most[2].

To illustrate how the trained P-VAE model proposes edits for different user groups, we use a set of similar images in the experts dataset and show the predicted slider values for those images. Fig. 6 shows how the inferred user groups edit a group of similar images (*i.e.*, flowers). This figure provides further evidence that the model is able to propose a diverse set of edits across different groups; moreover, it shows each user group may have a preference over which slider to use. For more examples see the supplementary material.

5 CONCLUSION

We proposed a framework for multimodal prediction of photo edits and extend the model to make personalized suggestions based on each user's previous edits. Our framework outperforms several strong baselines and demonstrates the benefit of having interpretable latent structure in VAEs. Although we only applied our framework to the data from photo editing applications, it can be applied to other domains where multimodal prediction, categorization and personalization are essential. Our proposed models could be extended further by assuming more complicated graphical model structure such as admixture models instead of the Gaussian mixture model that we used. Also, the categories learned by our model can be utilized to gain insights about the types of the users in the dataset.

Acknowledgements

We would like to thank Rahul Krishnan, and the anonymous reviewers for their insightful comments.

[2]To apply the P-VAE model to the experts dataset, we split the image-slider combinations from each of the 5 experts into groups of 50 image-sliders and pretend that each group belongs to a different user. This way we have more users to train the P-VAE model. However, this means the same expert may have some image-sliders in both train and test datasets. The significant advantage gained in the experts dataset might be due in part to this way of splitting the experts.

References

C. M. Bishop. Mixture density networks. 1994.

V. Bychkovsky, S. Paris, E. Chan, and F. Durand. Learning photographic global tonal adjustment with a database of input/output image pairs. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 97–104. IEEE, 2011.

Y. N. Dauphin and D. Grangier. Predicting distributions with linearizing belief networks. *arXiv preprint arXiv:1511.05622*, 2015.

M. Gharbi, J. Chen, J. T. Barron, S. W. Hasinoff, and F. Durand. Deep bilateral learning for real-time image enhancement. *ACM Transactions on Graphics (TOG)*, 36(4):118, 2017.

M. J. Johnson, D. Duvenaud, A. B. Wiltschko, S. R. Datta, and R. P. Adams. Composing graphical models with neural networks for structured representations and fast inference. In *Neural Information Processing Systems*, 2016.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Y. Koyama, D. Sakamoto, and T. Igarashi. Selph: Progressive learning and support of manual photo color enhancement. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 2520–2532. ACM, 2016.

J.-Y. Lee, K. Sunkavalli, Z. Lin, X. Shen, and I. S. Kweon. Automatic content-aware color and tone stylization. *arXiv preprint arXiv:1511.03748*, 2015.

C. J. Maddison, A. Mnih, and Y. W. Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

R. M. Neal. Connectionist learning of belief networks. *Artificial intelligence*, 56(1):71–113, 1992.

G. Sharma and R. Bala. *Digital color imaging handbook*. CRC press, 2002.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther. Ladder variational autoencoders. In *Advances in Neural Information Processing Systems*, pages 3738–3746, 2016.

Y. Tang and R. R. Salakhutdinov. Learning stochastic feedforward neural networks. In *Advances in Neural Information Processing Systems*, pages 530–538, 2013.

G. Tucker, A. Mnih, C. J. Maddison, and J. Sohl-Dickstein. Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. *arXiv preprint arXiv:1703.07370*, 2017.

Z. Yan, H. Zhang, B. Wang, S. Paris, and Y. Yu. Automatic photo adjustment using deep neural networks. *ACM Transactions on Graphics (TOG)*, 35(2):11, 2016.