# CONDITIONAL IMAGE GENERATION BY CONDITIONING VARIATIONAL AUTO-ENCODERS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

We present a conditional variational auto-encoder (VAE) which, to avoid the substantial cost of training from scratch, uses an architecture and training objective capable of leveraging a foundation model in the form of a pretrained unconditional VAE. Training the conditional VAE then involves training an artifact to perform amortized inference over the unconditional VAE's latent variables given a conditioning input. We demonstrate our approach on the image completion task, and show that it outperforms state-of-the-art GAN-based approaches at faithfully representing the inherent uncertainty. We conclude by describing and demonstrating an application that requires an image completion model with the capabilities ours exhibits: the use of Bayesian optimal experimental design to guide a sensor.

## 1 INTRODUCTION

A major challenge with applying variational auto-encoders (VAEs) to high-dimensional data is the typically slow training times. For example, training a state-of-the-art VAE (Vahdat & Kautz, 2020; Child, 2020) on the $256 \times 256$ FFHQ dataset (Karras et al., 2019) takes on the order of 1 GPU-year, but a state-of-the-art generative adversarial network (GAN) (Lin et al., 2021; Karras et al., 2020) can be trained on the same dataset in a matter of GPU-weeks. One hypothesis for the cause of this disparity is that, whereas the "mass-covering" training objective for a VAE forces it to assign probability mass over the entirety of the data distribution, a GAN can "cut corners" by dropping modes (Arora & Zhang, 2017; Arora et al., 2017).

We focus on the problem of *conditional* generative modelling: given an input (e.g. a partially blanked-out image), we wish to map to a distribution over outputs (e.g. plausible completions of the image). Both conditional GANs (Zheng et al., 2019; Zhao et al., 2021) and conditional VAEs (Sohn et al., 2015; Ivanov et al., 2018) are applicable to this problem, with the same disparity in training times that we described for their unconditional counterparts. We present an approach based on the conditional VAE framework but, to mitigate the associated slow training times, we design the architecture so that we can incorporate pretrained unconditional VAEs. We show that re-using publicly available pretrained models in this way can lead to training times competitive with GANs.

Figure 1: **Left column:** Images with most pixels masked out. **Rest:** Completions from our method.

While requiring an existing pretrained model is a limitation, we note that: **(I)** The unconditional VAE need not have been (pre-)trained on the same dataset as the conditional model; we show unconditional models trained on ImageNet are suitable for later use with various photographic image datasets. **(II)** A single unconditional VAE can be used for later training of conditional VAEs on any desired conditional generation tasks (e.g. the same image model may be later used for image completion or image colourisation). **(III)** There is an increasing trend in the machine learning community towards sharing large, expensively trained models (Wolf et al., 2020), sometimes referred to as foundation models (Bommasani et al., 2021). Most of the unconditional VAEs in our experiments use publicly-available pretrained weights released by Child (2020). By presenting a use case for foundation models in image modelling, we hope to encourage even more sharing of pretrained weights in this domain.

We demonstrate our approach on several conditional generation tasks in the image domain but focus in particular on stochastic image completion: the problem of inferring the posterior distribution over images given the observation of a subset of pixel values. The visual quality of image completions produced by our method (see Fig. 1) is close to the state-of-the-art (Zhao et al., 2021), and we show results indicating that our coverage of the "true" posterior over image completions is superior to that of any of our baselines.

For some applications such as photo-editing the implicit distribution defined by GANs is likely to be good enough. We argue that our approach has substantial advantages when image completion is used as part of a larger pipeline. We demonstrate one such situation in Section 5: Bayesian optimal experimental design (BOED) for guiding a sensor or hard attention mechanism (Ma et al., 2018; Harvey et al., 2019; Rangrej & Clark, 2021). In this case, missing modes of the posterior over images is likely to lead to bad decisions. We show that our objective corresponds to the mass-covering KL divergence and therefore covers the posterior well, and outperforms GANs which typically learn a distribution with a fraction of the support of the data distribution (Arora & Zhang, 2017; Arora et al., 2017).

**Contributions** We develop a method to cheaply convert pretrained unconditional VAEs into conditional VAEs. We demonstrate that this allows conditional VAE training times competitive with GANs, while avoiding the mode-dropping behaviour associated with GANs. Finally, we showcase an application in Bayesian optimal experimental design that benefits from these capabilities.

## 2 Variational Auto-encoders

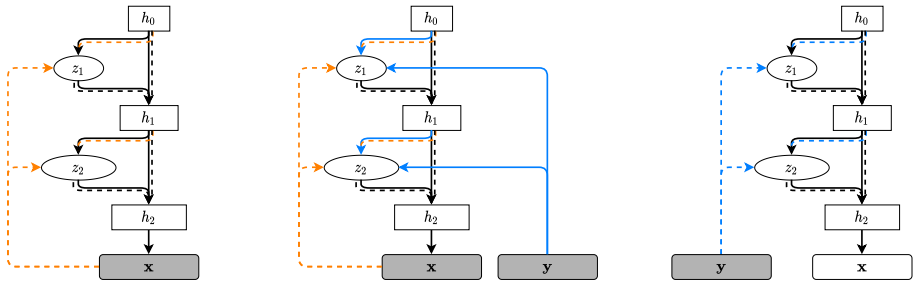
We describe VAEs in terms of three components. **(I)** A decoder with parameters $\theta \in \Theta$ maps from latent variables $z$ to a distribution over data $\mathbf{x}$, which we call $p_{\text{model}}(\mathbf{x}|z;\theta)$. **(II)** There is a prior over latent variables, $p_{\text{model}}(z;\theta)$. This may have learnable parameters, which we consider to be part of $\theta$. Together, the prior and decoder define a joint distribution, $p_{\text{model}}(z,\mathbf{x};\theta)$. Finally, **(III)** an encoder with parameters $\phi \in \Phi$ maps from data to an approximate posterior distribution over latent variables, $q(z|\mathbf{x};\phi) \approx p_{\text{model}}(z|\mathbf{x};\theta)$. Ideally, $\theta$ would be learned to maximise the log likelihood $\log p_{\text{model}}(\mathbf{x};\theta) = \log \int p_{\text{model}}(z,\mathbf{x};\theta)\mathrm{d}z$, averaged over training examples. Since this is intractable, $\theta$ and $\phi$ are instead trained jointly to maximise an average of the evidence lower-bound (ELBO) over each training example $\mathbf{x} \sim p_{\text{data}}(\cdot)$:

$$\mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[\text{ELBO}(\theta,\phi,\mathbf{x})\right] = \mathbb{E}_{p_{\text{data}}(\mathbf{x})}\mathbb{E}_{q(z|\mathbf{x};\phi)}\left[\log \frac{p_{\text{model}}(z;\theta)p_{\text{model}}(\mathbf{x}|z;\theta)}{q(z|\mathbf{x};\phi)}\right] \tag{1}$$

$$= -\mathcal{H}\left[p_{\text{data}}(\mathbf{x})\right] - \text{KL}\big(p_{\text{data}}(\mathbf{x})q(z|\mathbf{x};\phi) \;\big\|\; p_{\text{model}}(z,\mathbf{x};\theta)\big). \tag{2}$$

The data distribution's entropy, $\mathcal{H}\left[p_{\text{data}}(\mathbf{x})\right]$, is typically a finite constant, and this is guaranteed in our experiments where $\mathbf{x}$ is an image with discrete pixel values. Maximising the above objective will therefore drive $p_{\text{model}}(z,\mathbf{x};\theta)$ towards $p_{\text{data}}(\mathbf{x})q(z|\mathbf{x};\phi)$, and so the marginal $p_{\text{model}}(\mathbf{x};\theta)$ towards $p_{\text{data}}(\mathbf{x})$. The KL divergence shown leads to mass-covering behaviour from $p_{\text{model}}(z,\mathbf{x};\theta)$ (Bishop, 2006) so $p_{\text{model}}(\mathbf{x};\theta)$ should assign probability broadly over the data distribution $p_{\text{data}}(\mathbf{x})$. For notational simplicity in the rest of the paper, parameters $\theta$ and $\phi$ are not written when clear from the context.

Hierarchical VAEs (Gregor et al., 2015; Kingma et al., 2016; Sønderby et al., 2016; Klushyn et al., 2019) partition the latent variables $z$ in a way which has been found to improve the fidelity of




(a) Estimating ELBO. (b) Estimating $\mathcal{O}_{\text{for}}$. (c) Sampling $\mathbf{x} \sim p_{\text{cond}}(\cdot|\mathbf{y})$.

Figure 2: A hierarchical VAE architecture with $L = 2$. The blocks $h_0, \ldots, h_L$ represent the decoder's deterministic hidden state. Part (a) shows the computations involved in computing the ELBO for an unconditional VAE. The encoder is shown in orange, and the prior and decoder are shown in black. The dashed lines show the dependencies of the distribution from which $z$ is sampled. Part (b) shows the computations used to compute our training objective $\mathcal{O}_{\text{for}}$. This involves the partial encoder, shown in blue. Part (c) shows the computation graph used to sample image completions.

the learned $p_{\text{model}}(\mathbf{x})$, especially for the image domain (Vahdat & Kautz, 2020; Child, 2020). In particular, they define $z$ to consist of $L$ disjoint groups, $z_1, \ldots, z_L$. The prior for each $z_l$ can depend on the previous groups through the factorisation

$$p_{\text{model}}(z) = \prod_{l=1}^{L} p_{\text{model}}(z_l|z_{<l}). \tag{3}$$

where $z_{<l}$ is the null set for $l = 1$ and $\{z_1, \ldots, z_{l-1}\}$ otherwise. Fig. 2a shows the hierarchical VAE architecture we base this work on, in which the dependency of the prior for each $z_l$ on $z_{<l}$ is maintained via the decoder's (shown in black) hidden state $h_l$. The distribution produced by the encoder (shown in orange) for each $z_l$ also depends on the previous hidden state $h_{l-1}$ and so factorises as $q(z|\mathbf{x}) = \prod_{l=1}^{L} q(z_l|z_{<l}, \mathbf{x})$. We will parameterise $p_{\text{model}}(z_l|z_{<l})$ and $q(z_l|z_{<l}, \mathbf{x})$ as diagonal Gaussian distributions, as is common for hierarchical VAEs (Sønderby et al., 2016; Vahdat & Kautz, 2020; Child, 2020).

## 3 Amortized Inference in a Pretrained Artifact

To convert an unconditional VAE architecture to a conditional architecture, we introduce a *partial encoder* with parameters $\hat{\phi} \in \hat{\Phi}$. This is fed a conditioning input $\mathbf{y}$. For example, in the case of image completion, $\mathbf{y}$ could be an image with some pixels masked out. The partial encoder then defines an approximate posterior over the latent variables, $\hat{q}(z|\mathbf{y}; \hat{\phi})$.

The conditional generation task is to approximate $p_{\text{data}}(\mathbf{x}|\mathbf{y})$. Using the partial encoder, we define

$$p_{\text{cond}}(\mathbf{x}|\mathbf{y}; \theta, \hat{\phi}) := \int p_{\text{model}}(\mathbf{x}|z; \theta)\hat{q}(z|\mathbf{y}; \hat{\phi})\mathrm{d}z \tag{4}$$

with learnable parameters $\theta$ and $\hat{\phi}$. We can sample from $p_{\text{cond}}(\mathbf{x}|\mathbf{y})$ by sampling $z \sim \hat{q}(\cdot|\mathbf{y})$ and then $\mathbf{x} \sim p_{\text{model}}(\cdot|z)$ as shown in Fig. 2c. This defines a conditional VAE architecture which, unique amongst related work with high-dimensional $\mathbf{x}$ and $\mathbf{y}$ (Sohn et al., 2015; Zheng et al., 2019; Ivanov et al., 2018; Wan et al., 2021), has a decoder $p_{\text{model}}(\mathbf{x}|z; \theta)$ with no dependence on $\mathbf{y}$. This decoder can therefore use an architecture identical to that of an unconditional VAE and also, as we will show later, re-use unconditional VAE weights.

Before describing our method in further detail, we briefly introduce some notation. Let the distribution of paired data be $p_{\text{data}}(\mathbf{x}, \mathbf{y})$. Then recall that training an unconditional VAE matches two joint distributions: the distribution of samples from the generator, $p_{\text{model}}(z, \mathbf{x})$; and the distribution resulting from sampling data $\mathbf{x}$ and encoding it, $p_{\text{data}}(\mathbf{x})q(z|\mathbf{x})$. For notational convenience, we define the following extensions of these joint distributions to include $\mathbf{y}$:

$$p_{\text{model}}(z, \mathbf{x}, \mathbf{y}; \theta) = p_{\text{model}}(z; \theta)p_{\text{model}}(\mathbf{x}|z; \theta)p_{\text{data}}(\mathbf{y}|\mathbf{x}), \tag{5}$$

$$r(z, \mathbf{x}, \mathbf{y}; \phi) = p_{\text{data}}(\mathbf{x}, \mathbf{y})q(z|\mathbf{x}; \phi), \tag{6}$$

where $p_{\text{data}}(\mathbf{y}|\mathbf{x})$ is a (potentially intractable) conditional distribution under $p_{\text{data}}(\mathbf{x}, \mathbf{y})$. Note that $p_{\text{model}}(z, \mathbf{x}, \mathbf{y}; \theta)$ and $r(z, \mathbf{x}, \mathbf{y}; \phi)$ are exactly the two distributions matched by the unconditional VAE objective in Eq. (2) with an additional factor of $p_{\text{data}}(\mathbf{y}|\mathbf{x})$. Therefore, if the unconditional VAE represented by $\theta$ and $\phi$ is well trained, $p_{\text{model}}$ and $r$ will be close. From now on, we will use $p_{\text{model}}$ and $r$ to refer to any marginals and conditionals of the above joint distributions, with the specific marginal or conditional clear from context.

### 3.1 Training objective

Our training objective, previously used for training conditional VAEs (Sohn et al., 2015; Ivanov et al., 2018) and neural processes (Garnelo et al., 2018), is

$$\mathcal{O}_{\text{for}}(\theta, \phi, \hat{\phi}) = \mathbb{E}_{p_{\text{data}}(\mathbf{x},\mathbf{y})}\mathbb{E}_{q(z|\mathbf{x})}\left[\log \frac{p_{\text{model}}(\tilde{\mathbf{x}}|z)\hat{q}(z|\mathbf{y})}{q(z|\mathbf{x})}\right] \quad \leq \mathbb{E}_{p_{\text{data}}(\mathbf{x},\mathbf{y})}\left[\log p_{\text{cond}}(\tilde{\mathbf{x}}|\mathbf{y})\right] \tag{7}$$

where $\tilde{\mathbf{x}}$ is the part of $\mathbf{x}$ we wish to predict. In general we can set $\tilde{\mathbf{x}} := \mathbf{x}$ but for inpainting define $\tilde{\mathbf{x}}$ to consist of only the dimensions of $\mathbf{x}$ not observed in $\mathbf{y}$, abusing notation by ignoring the implication that $\tilde{\mathbf{x}}$ is formally a function of $\mathbf{x}$ and $\mathbf{y}$. The architectures we use have pixel-wise independent likelihoods so $p_{\text{model}}(\tilde{\mathbf{x}}|z)$ is tractable. Equation (7) lower-bounds $\log p_{\text{cond}}(\tilde{\mathbf{x}}|\mathbf{y})$ similarly to how an unconditional VAE's ELBO lower-bounds $\log p_{\text{model}}(\mathbf{x})$. The major difference is that the prior, $p_{\text{model}}(z)$, is replaced by $\hat{q}(z|\mathbf{y})$. This is reflected in Fig. 2b where each $z_l$ is conditioned on $\mathbf{y}$ via the partial encoder (blue). Similar to standard estimators for an unconditional hierarchical VAE's ELBO, reduced-variance estimates of Eq. (7) can be obtained by computing KL divergences between $q(z_l|z_{<l}, \mathbf{x})$ and $\hat{q}(z_l|z_{<l}, \mathbf{y})$ analytically for each layer $l$ (see Appendix F for details).

We are particularly interested in the properties of the learned partial encoder. Recall the joint distribution $r(z, \mathbf{x}, \mathbf{y}; \phi) = p_{\text{data}}(\mathbf{x}, \mathbf{y})q(z|\mathbf{x}; \phi)$. Then $r(z|\mathbf{y}; \phi)$ is the intractable posterior given by marginalising out $\mathbf{x}$ and conditioning on $\mathbf{y}$. We find that fitting $\hat{\phi}$ to maximise $\mathcal{O}_{\text{for}}(\theta, \phi, \hat{\phi})$ is equivalent to minimising the mass-covering KL divergence from $r(z|\mathbf{y}; \phi)$ to $\hat{q}(z|\mathbf{y}; \hat{\phi})$. We formalise this statement in the following theorem, which is proven in Appendix C.

**Theorem 3.1.** *For any set $\hat{\Phi}$ of permissible values of $\hat{\phi}$, and for any $\theta \in \Theta$ and $\phi \in \Phi$,*

$$\arg\max_{\hat{\phi}\in\hat{\Phi}} \mathcal{O}_{\text{for}}(\theta, \phi, \hat{\phi}) = \arg\min_{\hat{\phi}\in\hat{\Phi}} \mathbb{E}_{p_{\text{data}}(\mathbf{y})}\left[KL\big(r(z|\mathbf{y}; \phi) \;\|\; \hat{q}(z|\mathbf{y}; \hat{\phi})\big)\right]. \tag{8}$$

Due to the mass-covering properties of this "forward" KL divergence (Bishop, 2006), this theorem indicates that the learned $\hat{q}(z|\mathbf{y}; \hat{\phi})$ should have good coverage of all modes of $r(z|\mathbf{y}; \phi)$. Intuitively, the resulting diverse samples of latent variables $z \sim \hat{q}(\cdot|\mathbf{y}; \hat{\phi})$ should lead to diverse samples of $\tilde{\mathbf{x}} \sim p_{\text{model}}(\cdot|z)$ which cover the "true" posterior $p_{\text{data}}(\tilde{\mathbf{x}}|\mathbf{y})$. We formalise this in Appendix C by showing that maximising $\mathcal{O}_{\text{for}}$ also minimises an upper-bound on a KL divergence in $\tilde{\mathbf{x}}$-space.

### 3.2 Faster training with a pretrained VAE

To justify using weights trained as part of an unconditional VAE we present Theorem 3.2.

**Theorem 3.2.** *Assume that we have a sufficiently expressive encoder and decoder that there exist parameters $\theta^* \in \Theta$ and $\phi^* \in \Phi$ which make the unconditional VAE objective (Eq. (1)) equal to its upper bound of $-\mathcal{H}\left[p_{\text{data}}(\mathbf{x})\right]$. Assume also that the mutual information $I^*_{\tilde{\mathbf{x}},\mathbf{y}|z} := \mathbb{E}_{p_{\text{model}}(\tilde{\mathbf{x}},\mathbf{y},z;\theta^*)}\left[\log \frac{p_{\text{model}}(\tilde{\mathbf{x}},\mathbf{y}|z;\theta^*)}{p_{\text{model}}(\tilde{\mathbf{x}}|z;\theta^*)p_{\text{model}}(\mathbf{y}|z;\theta^*)}\right]$ is zero and $\tilde{\mathbf{x}}$ is defined such that there exists a mapping from $(\tilde{\mathbf{x}}, \mathbf{y})$ to $\mathbf{x}$. Then, given a sufficiently expressive partial encoder,*

$$\max_{\hat{\phi}} \mathcal{O}_{\text{for}}(\theta^*, \phi^*, \hat{\phi}) = \max_{\theta,\phi,\hat{\phi}} \mathcal{O}_{\text{for}}(\theta, \phi, \hat{\phi}). \tag{9}$$

See Appendix D for a proof and further explanation of the assumption that $I^*_{\tilde{\mathbf{x}},\mathbf{y}|z} = 0$. This implies that we can use values of $\theta$ and $\phi$ learned using the unconditional VAE objective. Then to train a conditional generative model we need only optimise $\hat{\phi}$. This leads to faster convergence, as well as faster training iterations since we only need to compute gradients for, and perform update steps on, the partial encoder's parameters $\hat{\phi}$. For all of our experiments in Section 4 we use pretrained models

Table 1: Image completion results. Best performance is shown in **bold**, and second best is underlined. In the last row, $t$ denotes the "temperature" parameter (Child, 2020).

| | CIFAR-10 | | | FFHQ-256 | | |
|---|---|---|---|---|---|---|
| Method | FID↓ | P-IDS↑ | LPIPS-GT↓ | FID↓ | P-IDS↑ | LPIPS-GT↓ |
| ANP | 30.03 | 5.86 | .0447 | 39.95 | 0.93 | .256 |
| CE | 21.92 | 4.77 | .0628 | 39.02 | 0.66 | .267 |
| RFR | 44.35 | 2.76 | .0883 | 72.50 | 0.46 | .271 |
| PIC | 14.73 | 5.95 | .0332 | 11.60 | 2.76 | .169 |
| CoModGAN | 9.65 | 11.59 | .0326 | **2.33** | **13.57** | .143 |
| IPA-R | 19.21 | 8.56 | .0330 | 8.82 | 4.56 | .142 |
| IPA (ours) | 10.50 | 13.24 | **.0262** | 3.93 | 7.79 | .123 |
| IPA$_{t=0.85}$ (ours) | **8.61** | **14.19** | .0263 | 3.29 | 8.50 | **.117** |

released by Child (2020), leveraging between 2 GPU-weeks and 1 GPU-year of unconditional VAE training for each dataset. We name our method IPA (Inference in a Pretrained Artifact).

Note that Theorem 3.2 applies only if the unconditional VAE parameters are learned on the same dataset as the conditional VAE is trained on; otherwise there will be a mismatch between the form of $p_{\text{data}}$ used in Eq. (1) to fit $\theta^*$ and $\phi^*$, and the form of $p_{\text{data}}$ implicit in the $\mathcal{O}_{\text{for}}$ objective. However we find empirically that we can use unconditional VAE parameters trained on ImageNet (Deng et al., 2009) with IPA on several other photographic image datasets.

## 4 Experiments

**Comparison to image completion baselines** We create an IPA image completion model based on the VD-VAE unconditional architecture (Child, 2020), and evaluate it for image completion on three datasets, CIFAR-10 (Krizhevsky et al., 2009), ImageNet-64 (Deng et al., 2009), and FFHQ-256 (Karras et al., 2019). We compare against four baselines: Co-Modulated Generative Adversarial Networks (CoModGAN) (Zhao et al., 2021); Pluralistic Image Completion (PIC) (Zheng et al., 2019); Context Encoders (CE) (Pathak et al., 2016); and Attentive Neural Processes (ANP) (Kim et al., 2019). We report some results for another two baselines: we show qualitative results for VQ-VAE (Peng et al., 2021), but not quantitative results because it takes too long (about a minute) to complete each test image. We report results for Recurrent Feature Reasoning for Image Inpainting (RFR) (Li et al., 2020a) with the caveat that it is slow to run on images with many missing pixels and so, although it used a similar computational budget to the other models, its training did not converge.

Given pretrained unconditional VAE parameters, IPA is faster to train than the best-performing baseline, CoModGAN. IPA takes 115 GPU-hours to train on CIFAR-10, and under 7 GPU-weeks on FFHQ-256. The CoModGAN models are trained for 270 GPU-hours and 8 GPU-weeks respectively. We provide more training details in Appendix G.

We report the FID (Heusel et al., 2017) and P-IDS (Zhao et al., 2021) metrics between a set of sampled completions from each method and a reference set. Broadly speaking, these measure the sample quality. To investigate how well the samples capture all modes of $p_{\text{data}}(\mathbf{x}|\mathbf{y})$, we also report the LPIPS-GT. We compute this using LPIPS (Zhang et al., 2018), a measure of distance between two images. Specifically, we compute the average over test pairs $(\mathbf{x}, \mathbf{y})$ of $\min_{k=1}^{K}(\text{LPIPS}(\mathbf{x}^{(k)}, \mathbf{x}))$, with each $\mathbf{x}^{(k)} \sim p_{\text{cond}}(\cdot|\mathbf{y})$. As $K \to \infty$, the LPIPS-GT should tend to zero if the ground truth completion is always within the support of $p_{\text{cond}}(\mathbf{x}|\mathbf{y})$. If not, the LPIPS-GT will remain high, penalising methods which miss modes of the posterior. We use $K = 100$. We report results on several other measures of diversity in Appendix J.

For the image completion tasks, we sample from $p_{\text{data}}(\mathbf{x}, \mathbf{y})$ by first sampling an image $\mathbf{x}$ from the dataset, and then sampling an image-sized binary mask $m$ from the freeform mask distribution used by Zhao et al. (2021), which is itself based on Yu et al. (2018). We then set $\mathbf{y} = \texttt{concatenate}(\mathbf{x} \odot m, m)$. Here, $\odot$ is a pixel-wise multiplication operation which removes information from the missing pixels. The concatenation is performed along the channel dimension and makes it possible to distinguish between unobserved pixels and zero-valued pixels.

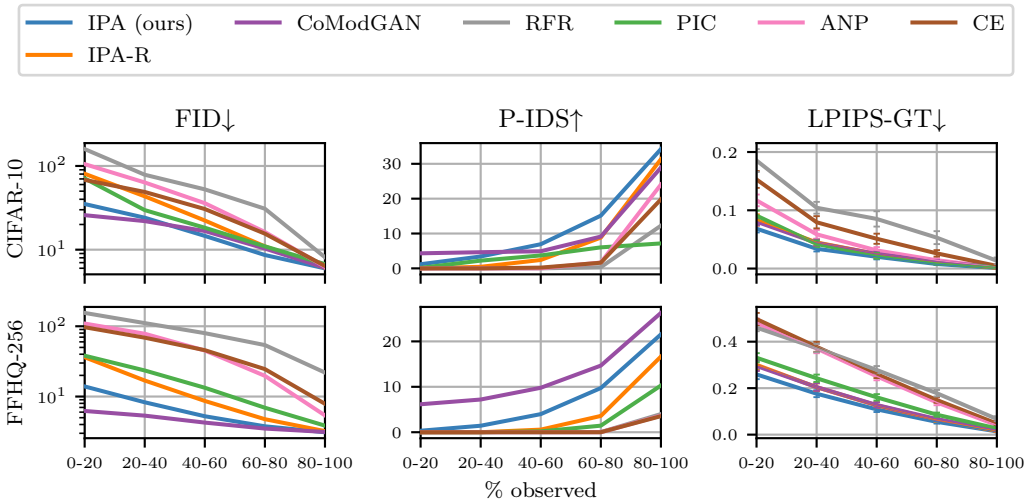



Figure 3: Our three test metrics evaluated for CIFAR-10 (top row) and FFHQ-256 (bottom row), and plotted as a function of the mask distribution. The error bars on LPIPS-GT show the standard error of our estimate for a single trained network.

For evaluation, since the number of observed pixels in freeform masks varies considerably, we follow Zhao et al. (2021) and partition the mask distribution by conditioning the procedure to return a mask with the proportion of pixels observed within some range (0-20%, 20-40%, and so on) and report metrics for each range separately in Fig. 3. To summarise the overall performance in Table 1, we sample masks from a uniformly-weighted mixture distribution over these five partitions. In terms of the LPIPS-GT scores in Table 1, IPA outperforms the best baselines by roughly 20%. Figure 3 shows that there is an improvement for any proportion of observed pixels. This suggests that IPA produces reliably diverse samples with good coverage of $p_{\text{data}}(\mathbf{x}|\mathbf{y})$. In contrast, we believe that the GAN-based approaches occasionally miss modes of $p_{\text{data}}(\mathbf{x}|\mathbf{y})$ and can therefore fail to capture the ground-truth. This hypothesis is supported by samples from CoModGAN we display in Appendix K. In terms of sample fidelity, as measured by both FID and P-IDS, IPA outperforms all baselines on CIFAR-10 when $> 40\%$ of the image is observed, and comes second to CoModGAN when $< 40\%$ is observed and on FFHQ-256.

**Edges-to-photos** We provide an additional demonstration of IPA on the Edges2Shoes and Edges2Handbags datasets (Isola et al., 2016), where the task is to generate an image conditioned on the output of an edge detector applied to that image. We downsample the datasets to $64 \times 64$ so that we can use unconditional VAEs pretrained on ImageNet (Deng et al., 2009) at this resolution by Child (2020). We show in Fig. 4 that IPA is useful for these tasks, and provide further discussion below. The images generated are diverse and photorealistic, as shown in Appendix L.

**Effectiveness of pretraining** We now seek to answer the question of how important the pretrained unconditional VAE weights are to IPA. To do so, we compare IPA with conditional VAEs which use the same architecture as IPA but are trained from scratch, and which we will refer to as "from-scratch" baselines. That is, $\theta$ and $\phi$ are randomly initialised and trained to maximise Eq. (7) (with $\tilde{\mathbf{x}} := \mathbf{x}$) along with $\hat{\phi}$. With an infinite training budget, the end-to-end training of the from-scratch baselines is likely to lead them to outperform any IPA models. Nevertheless it is apparent from Fig. 4 that, in the more realistic situation of a finite training budget, using IPA can be beneficial. This is the case even for training budgets of up to a few GPU-weeks on the relatively small CIFAR-10 dataset. In fact, even with only a couple of days of training, IPA on CIFAR-10 (with CIFAR-10 pretraining) achieves better FID and ELBO scores than the from-scratch baseline trained for several weeks. For ImageNet-64, IPA performs better after 4 hours than the from-scratch baseline does after 4 days.

For Edges2Handbags and Edges2Shoes, training with IPA for 2 days yields performance similar to or better than training with the from-scratch baseline for 1 week, as measured by the ELBO. This is despite IPA on these datasets using a trained ImageNet-64 model rather than a model pretrained

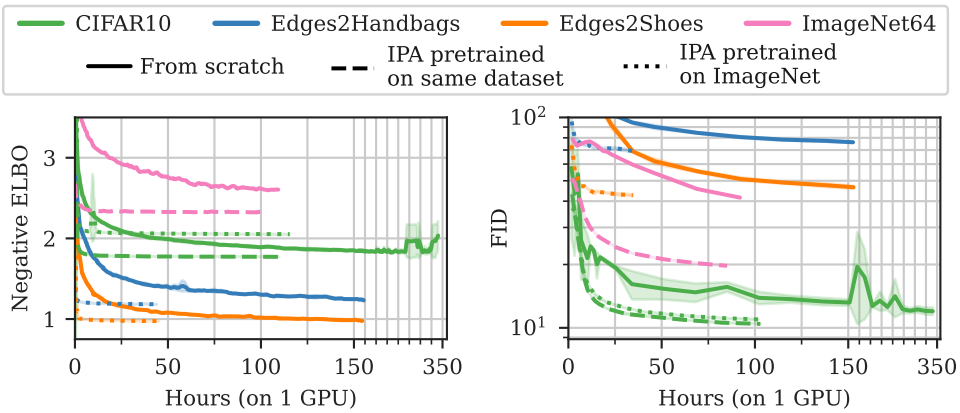



Figure 4: ELBO (Eq. (7) computed with $\tilde{\mathbf{x}} := \mathbf{x}$) and FID during training using IPA with pretraining on the same dataset, IPA with pretraining on ImageNet, and when trained from scratch. Error bars show standard deviations computed with 3 runs. IPA makes training faster and lower-variance.

on those specific datasets, supporting our suggestion that the dataset used for pretraining need not exactly match what IPA is then trained on. When measuring performance with the FID score, IPA looks even more appealing: wherever ELBOs are similar between IPA and the from-scratch baselines, IPA achieves a significantly better FID score. We see that IPA pretrained on ImageNet is less effective for CIFAR-10 than it is for the edges-to-photos datasets, but it nevertheless improves on the from-scratch baseline in terms of ELBO for the first 30 hours of training, and in terms of FID until the from-scratch baseline is trained for several weeks.

**An alternative training objective** In Table 1 and Fig. 3, we report results for IPA-R, a variation of IPA with a different training objective corresponding to a mode-seeking KL divergence. IPA almost always outperforms IPA-R, but we nonetheless provide a full description of IPA-R in Appendix E.

## 5 Application: Experimental Design for Medical Imaging

In this section, we demonstrate an application where stochastic image completion, and faithful representation of the posterior $p_{\text{data}}(\mathbf{x}|\mathbf{y})$, is necessary. In particular, we consider whether it is possible to automatically target a chest x-ray at areas most likely to reveal abnormalities. This could avoid the need to scan the entire chest and so bring benefits including reducing the patient's radiation exposure. We do not claim that our system, as it is now, is suitable for use in a clinical setting but believe this is a worthwhile avenue to explore. Specifically, we consider performing a series of x-ray scans, each targeted at only a small portion of the area of interest. We can select the coordinates $c_t = (x_t, y_t)$ of the location to scan at each step $t$, and this selection can be informed by what was observed in the previous scans. The task we consider is how to select $c_t$ to be maximally informative. In particular, assume we wish to infer a variable $v$ representing, e.g., whether the patient has a particular illness. Bayesian optimal experimental design (BOED) (Chaloner & Verdinelli, 1995) provides a framework to select a value of $c_t$ that is maximally informative about $v$. It involves taking a Bayesian perspective on the problem of estimating $v$. We have one posterior distribution over $v$ after taking scans at $c_1, \ldots, c_{t-1}$ and another (typically lower entropy) distribution after conditioning on a scan at $c_t$ as well. The *expected information gain*, or EIG, quantifies the utility of the choice of $c_t$ as the expected difference in entropy between these two distributions. Using BOED involves estimating the EIG and selecting the scan location, $c_t$, to minimise it.

We use an estimator for the EIG similar to that of Harvey et al. (2019). It requires two components: **(I)** A neural network trained to classify $v$ given a series of scans at locations $c_1, \ldots, c_t$. This outputs a classification distribution which we denote $g(v|f_{c_1,\ldots,c_t}(\mathbf{x}))$, where $f_{c_1,\ldots,c_t}$ is a function mapping from an image to the values of the pixels observed by scans at $c_1, \ldots, c_t$. We use this classification distribution as an approximation of the posterior over $v$, whose entropy we attempt to minimise by performing BOED. **(II)** A method for sampling image completions conditioned on some observed pixel values $f_{c_1,\ldots,c_{t-1}}(\mathbf{x})$. Harvey et al. (2019) used a "stochastic image completion" module which contributed significant complexity to their method. We entirely replace this with IPA.

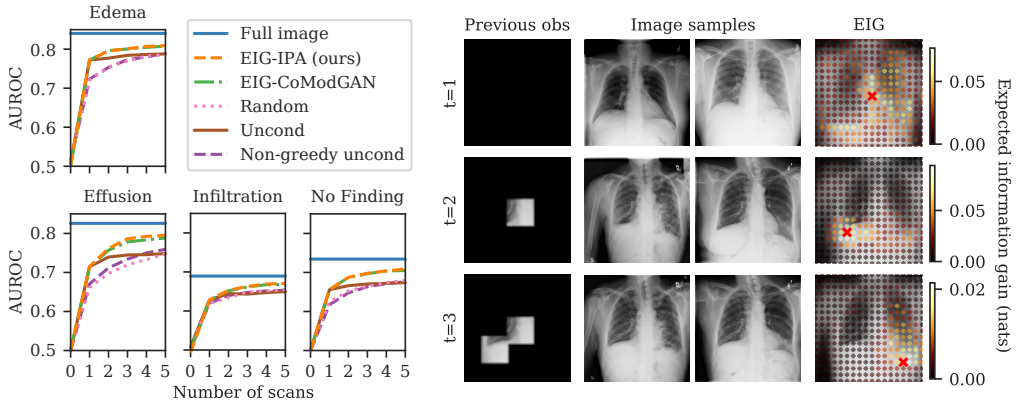



Figure 5: **Left:** Classification AUROC scores after $1, \ldots, 5$ scans chosen with each method. Scores for the "EIG-" methods more quickly approach the upper bound achieved by processing the full image. **Right:** Visualisation of BOED used to select three scan locations for diagnosing 'Effusion'. The left column shows the observations made prior to each time step. We then show samples from IPA (or the dataset when $t = 1$). The rightmost column shows the EIG overlaid on the pixel-space average of sampled images, with the optimal $c_t$ marked by a red cross.

Let the pixel values observed so far be $\mathbf{y}_{c_1,\ldots,c_{t-1}} = f_{c_1,\ldots,c_{t-1}}(\mathbf{x})$ for a latent image $\mathbf{x}$. Given these, we estimate the EIG of location $c_t$ as

$$\text{EIG}(c_t;\mathbf{y}_{c_1,\ldots,c_{t-1}}) \approx \overbrace{\mathcal{H}\left[\frac{1}{N}\sum_{n=1}^{N} g(\cdot|f_{c_1,\ldots,c_t}(\mathbf{x}^{(n)}))\right]}^{\text{entropy after } t-1 \text{ scans}} - \overbrace{\frac{1}{N}\sum_{n=1}^{N}\mathcal{H}\left[g(\cdot|f_{c_1,\ldots,c_t}(\mathbf{x}^{(n)}))\right]}^{\text{expected entropy after } t \text{ scans}}, \quad (10)$$

where $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}$ are sampled image completions from IPA given $\mathbf{y}_{c_1,\ldots,c_{t-1}}$. In Appendix H we report hyperparameters, provide further details of our EIG estimator, and compare it to the estimators used in related work. To select $c_t$, we simply estimate $\text{EIG}(c_t; \mathbf{y}_{c_1,\ldots,c_{t-1}})$ for many different values of $c_t$ and select the value which maximises it. This process of selecting $c_t$ and then taking a scan is repeated for each $t = 1, \ldots, T$.

We experiment on the NIH Chest X-ray 14 dataset (Wang et al., 2017) at $256 \times 256$ resolution. We simulate a scanner which returns a $64 \times 64$ pixel patch from this image, and the task is to diagnose the binary presence or absence of an illness. We run separate experiments diagnosing each of edema, effusion, infiltration and "no finding" (an additional label meaning there are no diagnosed illnesses). With appropriate data, this framework could be extended to also infer the severity of a given illness. We envisage BOED being used to select scan locations for an x-ray without necessarily performing an automated diagnosis. However, to quantify the informativeness of the chosen locations, Fig. 5 shows the results of using $g$ to perform a diagnosis, or classification, based on the chosen scan locations. Since the conditional distribution $g$ (used to estimate the EIG) depends on which illness we are classifying, the choice of scan locations is different in each case. We compare against a baseline where the image completion is performed by CoModGAN (our best-performing image completion baseline) rather than IPA, as well as numerous baselines which choose scan locations without image completion; see Appendix H for details.

Our method (denoted EIG-IPA) narrowly but consistently outperforms EIG-CoModGAN. We hypothesise that this is due to the aforementioned tendency of CoModGAN to sometimes collapse to a single mode of the posterior, and exhibit an example of this behaviour on the x-ray dataset in Appendix L. In the BOED context, such "overconfident" image completion could lead to salient scan locations being ignored. Nonetheless, both EIG-IPA and EIG-CoModGAN significantly outperform the other baselines, giving performance much closer to the upper bound of a CNN with access to the entire image. Another benefit of the "EIG-" approaches is that the choice of scan locations is highly interpretable; we can see why a particular location was chosen with visualisations similar to the right of Fig. 5. This shows the sampled images $\mathbf{x}^{(n)}$ and the estimated EIG for each $c_t$. In Appendix H, we show that we can further quantify the contribution of each $\mathbf{x}^{(n)}$ to the estimated EIG for each $c_t$.

## 6 Related Work

**Inference in pretrained VAEs** Several prior studies perform conditional generation using a previously trained unconditional VAE. Like us, Rezende et al. (2014); Nguyen et al. (2016); Wu et al. (2018) do so through inference in the VAE's latent space. However, they use non-amortized inference (Gibbs sampling, variational inference, and MCMC respectively), leading to slow sampling times for any new $\mathbf{y}$. Duan et al. (2019) learn variational distributions over $z$ for every possible value of $\mathbf{y}$, but this is not possible when $\mathbf{y}$ is high-dimensional or continuous-valued. Yeh et al. (2017) fit the latent variables of a GAN given observations, but this is neither amortized nor probabilistic.

**Conditional VAEs** Past research on conditional VAEs (Sohn et al., 2015; Zheng et al., 2019; Ivanov et al., 2018; Wan et al., 2021) has generally been unable to take advantage of pretrained weights as we have due to a difference in architectures: unlike almost all prior work, the IPA decoder does not receive $\mathbf{y}$ as input. The dependence between $\mathbf{y}$ and the decoder's output must therefore be expressed solely through the conditional distribution over the latent variables, $\hat{q}(z|\mathbf{y})$. This is a crucial difference because it means that the decoder can have exactly the same architecture as that of an unconditional VAE. This is key to letting us copy the pretrained weights of an unconditional VAE to speed up training. The exception to the above is Ma et al. (2018) who, like us, use a conditional VAE decoder with no dependence on $\mathbf{y}$. Their training objective and use case are different, however, and they do not consider using pretrained models or use an architecture which can scale to photorealistic images. Leveraging unconditional VAEs lets us drastically reduce the computational budget required to train a conditional VAE. We believe that this paper is the first to demonstrate photorealistic image completion with conditional VAEs at resolutions as high as $256 \times 256$. Another benefit of the decoder having no dependence on $\mathbf{y}$ is that it makes impossible the "posterior collapse" phenomenon discussed by Zheng et al. (2019), in which a conditional VAE's decoder learns to ignore $z$ and produce outputs conditioned solely on $\mathbf{y}$.

**Image completion** Early work on image completion, both before (Bertalmio et al., 2000; 2001; Ballester et al., 2001; Levin et al., 2003; Criminisi et al., 2003) and after (Köhler et al., 2014; Ren et al., 2015) deep learning became the dominant approach, aimed to deterministically fill in missing pixels in images. Even many methods incorporating generative adversarial networks (GANs), which were introduced by Goodfellow et al. (2014) as a tool to learn distributions, have been found to result in little or no diversity in the completions produced for a given input (Song et al., 2018; Yu et al., 2018; 2019; Pathak et al., 2016; Iizuka et al., 2017). However, some recent methods have managed to obtain diverse completions using the GAN framework (Zhao et al., 2020; 2021; Liu et al., 2021). Another approach is to generate a distribution over low-resolution images using VAEs (Zheng et al., 2019; Peng et al., 2021) or transformers (Zheng et al., 2021; Wan et al., 2021), and then use a GAN for upsampling. In contrast, we use a VAE to model image completions at the full resolution. As well as ensuring diverse coverage of the posterior, using such a likelihood-based model enables applications such as out-of-distribution detection for inputs $\mathbf{y}$, which we demonstrate in Appendix I. Another related approach is that of Song et al. (2020), who present a stochastic differential equation-based image model. This can be conditioned on subsets of image pixels to perform image completion, but sampling is slow.

## 7 Discussion and Conclusion

We have presented IPA, a method to adapt an unconditional VAE into a conditional model. Image completions generated with IPA are close to the state-of-the-art in terms of visual fidelity, and improve on all baselines in terms of their coverage of the posterior as measured by LPIPS-GT. This high-fidelity coverage of the posterior makes IPA ideal for use in Bayesian optimal experimental design, as demonstrated. In addition, IPA has all the benefits of a likelihood-based method, such as the potential to perform out-of-distribution detection. Future work may investigate further improving the image quality by, for example, using a partial encoder with more expressive distributions. Preliminary experiments revealed that normalizing flows could help the partial encoder better match the posterior, but with little impact on the resulting FID scores. Alternative directions include investigating different types of conditional generation, such as conditioning on previous frames to make a video model.

## REFERENCES


Sanjeev Arora and Yi Zhang. Do gans actually learn the distribution? an empirical study. *arXiv preprint arXiv:1706.08224*, 2017.

Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and equilibrium in generative adversarial nets (gans). In *International Conference on Machine Learning*, pp. 224–232. PMLR, 2017.

Coloma Ballester, Marcelo Bertalmio, Vicent Caselles, Guillermo Sapiro, and Joan Verdera. Filling-in by joint interpolation of vector fields and gray levels. *IEEE transactions on image processing*, 10(8):1200–1211, 2001.

Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp. 417–424, 2000.

Marcelo Bertalmio, Andrea L Bertozzi, and Guillermo Sapiro. Navier-stokes, fluid dynamics, and image and video inpainting. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pp. I–I. IEEE, 2001.

Lukas Biewald. Experiment tracking with weights and biases, 2020. URL https://www.wandb.com/. Software available from wandb.com.

Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical Science*, pp. 273–304, 1995.

Rewon Child. Very deep vaes generalize autoregressive models and can outperform them on images. *arXiv preprint arXiv:2011.10650*, 2020.

Antonio Criminisi, Patrick Perez, and Kentaro Toyama. Object removal by exemplar-based inpainting. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 2, pp. II–II. IEEE, 2003.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Yu Duan, Canwen Xu, Jiaxin Pei, Jialong Han, and Chenliang Li. Pre-train and plug-in: Flexible conditional text generation with variational auto-encoders. *arXiv preprint arXiv:1911.03882*, 2019.

Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J Rezende, SM Eslami, and Yee Whye Teh. Neural processes. *arXiv preprint arXiv:1807.01622*, 2018.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation. In *International Conference on Machine Learning*, pp. 1462–1471. PMLR, 2015.

William Harvey, Michael Teng, and Frank Wood. Near-optimal glimpse sequences for improved hard attention neural network training. *arXiv preprint arXiv:1906.05462*, 2019.

Jakob D Havtorn, Jes Frellsen, Søren Hauberg, and Lars Maaløe. Hierarchical vaes know what they don't know. *arXiv preprint arXiv:2102.08248*, 2021.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *arXiv preprint arXiv:1706.08500*, 2017.

Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (ToG)*, 36(4):1–14, 2017.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. arxiv (2016). *arXiv preprint arXiv:1611.07004*, 2016.

Oleg Ivanov, Michael Figurnov, and Dmitry Vetrov. Variational autoencoder with arbitrary conditioning. *arXiv preprint arXiv:1806.02382*, 2018.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401–4410, 2019.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8110–8119, 2020.

Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. Attentive neural processes. *arXiv preprint arXiv:1901.05761*, 2019.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Diederik P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improving variational inference with inverse autoregressive flow. *arXiv preprint arXiv:1606.04934*, 2016.

Alexej Klushyn, Nutan Chen, Richard Kurle, Botond Cseke, and Patrick van der Smagt. Learning hierarchical priors in vaes. *arXiv preprint arXiv:1905.04982*, 2019.

Rolf Köhler, Christian Schuler, Bernhard Schölkopf, and Stefan Harmeling. Mask-specific inpainting with deep neural networks. In *German conference on pattern recognition*, pp. 523–534. Springer, 2014.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Anat Levin, Assaf Zomet, and Yair Weiss. Learning how to inpaint from global image statistics. In *ICCV*, volume 1, pp. 305–312, 2003.

Jingyuan Li, Ning Wang, Lefei Zhang, Bo Du, and Dacheng Tao. Recurrent feature reasoning for image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7760–7768, 2020a.

Ke Li, Shichong Peng, Tianhao Zhang, and Jitendra Malik. Multimodal image synthesis with conditional implicit maximum likelihood estimation. *International Journal of Computer Vision*, 128: 2607–2628, 2020b.

Ji Lin, Richard Zhang, Frieder Ganz, Song Han, and Jun-Yan Zhu. Anycost gans for interactive image synthesis and editing. *arXiv preprint arXiv:2103.03243*, 2021.

Hongyu Liu, Ziyu Wan, Wei Huang, Yibing Song, Xintong Han, and Jing Liao. Pd-gan: Probabilistic diverse gan for image inpainting. *arXiv preprint arXiv:2105.02201*, 2021.

Chao Ma, Sebastian Tschiatschek, Konstantina Palla, José Miguel Hernández-Lobato, Sebastian Nowozin, and Cheng Zhang. Eddi: Efficient dynamic discovery of high-value information with partial vae. *arXiv preprint arXiv:1809.11142*, 2018.

Tom Minka et al. Divergence measures and message passing. Technical report, Microsoft Research, 2005.

Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? *arXiv preprint arXiv:1810.09136*, 2018.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

Anh Nguyen, Jason Yosinski, Yoshua Bengio, Alexey Dosovitskiy, and Jeff Clune. Plug & play generative networks: Conditional iterative generation of images in latent space.(2016). *arXiv preprint cs.CV/1612.00005*, 2016.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2536–2544, 2016.

Jialun Peng, Dong Liu, Songcen Xu, and Houqiang Li. Generating diverse structure for image inpainting with hierarchical vq-vae. *arXiv preprint arXiv:2103.10022*, 2021.

Samrudhdhi Bharatkumar Rangrej and James J. Clark. Achieving explainability in a visual hard attention model through content prediction, 2021. URL https://openreview.net/forum?id=pQq3oLH9UmL.

Jie Ren, Peter J Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark A DePristo, Joshua V Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection. *arXiv preprint arXiv:1906.02845*, 2019.

Jimmy SJ Ren, Li Xu, Qiong Yan, and Wenxiu Sun. Shepard convolutional neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*, pp. 901–909, 2015.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pp. 1278–1286. PMLR, 2014.

Mark Sabini and Gili Rusak. Painting outside the box: Image outpainting with gans. *arXiv preprint arXiv:1808.08483*, 2018.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems 29*, pp. 2234–2242. Curran Associates, Inc., 2016.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28: 3483–3491, 2015.

Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. *arXiv preprint arXiv:1602.02282*, 2016.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Yuhang Song, Chao Yang, Yeji Shen, Peng Wang, Qin Huang, and C-C Jay Kuo. Spg-net: Segmentation prediction and guidance network for image inpainting. *arXiv preprint arXiv:1805.03356*, 2018.

Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *arXiv preprint arXiv:2007.03898*, 2020.

Ziyu Wan, Jingbo Zhang, Dongdong Chen, and Jing Liao. High-fidelity pluralistic image completion with transformers. *arXiv preprint arXiv:2103.14031*, 2021.

Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2097–2106, 2017.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Ga Wu, Justin Domke, and Scott Sanner. Conditional inference in pre-trained variational autoencoders via cross-coding. *arXiv preprint arXiv:1805.07785*, 2018.

Zhisheng Xiao, Qing Yan, and Yali Amit. Likelihood regret: An out-of-distribution detection score for variational auto-encoder. *arXiv preprint arXiv:2003.02977*, 2020.

Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5485–5493, 2017.

Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5505–5514, 2018.

Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4471–4480, 2019.

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

Lei Zhao, Qihang Mo, Sihuan Lin, Zhizhong Wang, Zhiwen Zuo, Haibo Chen, Wei Xing, and Dongming Lu. Uctgan: Diverse image inpainting based on unsupervised cross-space translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5741–5750, 2020.

Shengyu Zhao, Jonathan Cui, Yilun Sheng, Yue Dong, Xiao Liang, Eric I Chang, and Yan Xu. Large scale image completion via co-modulated generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2021.

Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Pluralistic image completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1438–1447, 2019.

Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Tfill: Image completion via a transformer-based architecture. *arXiv preprint arXiv:2104.00845*, 2021.

Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. *arXiv preprint arXiv:1711.11586*, 2017.

Table 2: Definitions for symbols used.

| Symbol | Definition |
|---|---|
| $z$ | VAE's latent variables. |
| $\mathbf{x}$ | Data we learn a generative model of. |
| $\mathbf{y}$ | Data on which our generative model is conditioned. |
| $\tilde{\mathbf{x}}$ | The part of $\mathbf{x}$ which the conditional VAE models, defined as all pixels not in $\mathbf{y}$ for image completion, or simply $\tilde{\mathbf{x}} := \mathbf{x}$ for other conditional generation tasks. |
| $\theta$ | Parameters for VAE's prior and decoder. |
| $\phi$ | Parameters for VAE's encoder. |
| $p_{\text{data}}(\mathbf{x}, \mathbf{y})$ | Distribution from which we assume training/test instances are i.i.d. samples. |
| $p_{\text{model}}(z)$ | Learned prior of VAE. Implicitly parameterised by $\theta$ when not otherwise specified. |
| $p_{\text{model}}(\mathbf{x}\|z)$ | Distribution output by VAE's decoder given $z$. Implicitly parameterised by $\theta$ when not otherwise specified. |
| $q(z\|\mathbf{x})$ | Distribution output by VAE's encoder given $\mathbf{x}$. Implicitly parameterised by $\phi$ when not otherwise specified. |
| $p_{\text{model}}(z, \mathbf{x}, \mathbf{y})$ | Joint distribution defined as $p_{\text{model}}(z)p_{\text{model}}(\mathbf{x}\|z)p_{\text{data}}(\mathbf{y}\|\mathbf{x})$. |
| $r(z, \mathbf{x}, \mathbf{y})$ | Joint distribution defined as $p_{\text{data}}(\mathbf{x}, \mathbf{y})q(z\|\mathbf{x})$. |
| $\hat{\phi}$ | Partial encoder parameters. |
| $\hat{q}(z\|\mathbf{y})$ | Distribution output by partial encoder. Implicitly parameterised by $\hat{\phi}$. |
| $p_{\text{cond}}(\mathbf{x}\|\mathbf{y})$ | Distribution from which we sample image completions, $\int p_{\text{model}}(\mathbf{x}\|z)\hat{q}(z\|\mathbf{y})\mathrm{d}z$. |
| $p^*(z, \mathbf{x}, \mathbf{y})$ | Distribution with parameters $\theta^*$ and $\phi^*$ that are optimal as defined in Appendix D. Defined equivalently as $p_{\text{model}}(z, \mathbf{x}, \mathbf{y}; \theta^*)$ or $p_{\text{data}}(\mathbf{x}, \mathbf{y})q(z\|\mathbf{x}; \phi^*)$. |
| Symbols for Bayesian optimal experimental design: | |
| $c_t$ | The location scanned at time $t$. |
| $v$ | Variable we wish to infer. |
| $\mathbf{y}_{c_1,\ldots,c_t}$ | Image in which all pixel values are missing except for those observed by scans at locations $c_1, \ldots, c_t$. |
| $f_{c_1,\ldots,c_t}$ | Function mapping from an image $\mathbf{x}$ to $\mathbf{y}_{c_1,\ldots,c_t}$. |
| $g(v\|\mathbf{y})$ | Classification distribution for $v$ given partially observed image $\mathbf{y}$. |

# A NOTATION

We provide Table 2 for reference as a concise list of definitions for each symbol used. More thorough explanations are provided where symbols are introduced in the text

# B EXPANDED DERIVATIONS

## B.1 EXPANDED DERIVATION OF EQ. (2)

Here we expand on the steps to show that the expectation of the ELBO in Eq. (1) is equivalent to the lower bound on the negative entropy of $p_{\text{data}}(\mathbf{x})$ in Eq. (2).

$$\mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[\text{ELBO}(\theta, \phi, \mathbf{x})\right] = \mathbb{E}_{p_{\text{data}}(\mathbf{x})}\mathbb{E}_{q(z|\mathbf{x};\phi)}\left[\log \frac{p_{\text{model}}(z;\theta)p_{\text{model}}(\mathbf{x}|z;\theta)}{q(z|\mathbf{x};\phi)}\right] \tag{11}$$

$$= \mathbb{E}_{p_{\text{data}}(\mathbf{x})q(z|\mathbf{x};\phi)}\left[\log \frac{p_{\text{model}}(z, \mathbf{x};\theta)}{q(z|\mathbf{x};\phi)}\right] \tag{12}$$

$$= \mathbb{E}_{p_{\text{data}}(\mathbf{x})q(z|\mathbf{x};\phi)}\left[\log p_{\text{data}}(\mathbf{x}) + \log \frac{p_{\text{model}}(z, \mathbf{x};\theta)}{p_{\text{data}}(\mathbf{x})q(z|\mathbf{x};\phi)}\right] \tag{13}$$

$$= \mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[\log p_{\text{data}}(\mathbf{x})\right] + \mathbb{E}_{p_{\text{data}}(\mathbf{x})q(z|\mathbf{x};\phi)}\left[\log \frac{p_{\text{model}}(z, \mathbf{x};\theta)}{p_{\text{data}}(\mathbf{x})q(z|\mathbf{x};\phi)}\right] \tag{14}$$

$$= -\mathcal{H}\left[p_{\text{data}}(\mathbf{x})\right] - \text{KL}\big(p_{\text{data}}(\mathbf{x})q(z|\mathbf{x};\phi) \,\big\|\, p_{\text{model}}(z, \mathbf{x};\theta)\big) \tag{15}$$

as written in Eq. (2).

### B.2 EXPANDED DERIVATION OF EQ. (7)

Here we prove Eq. (7), which states that IPA's training objective is a lower bound on $\mathbb{E}_{p_{\text{data}}(\mathbf{x},\mathbf{y})}\left[\log p_{\text{cond}}(\mathbf{x}|\mathbf{y})\right]$.

$$\mathcal{O}_{\text{for}}(\theta,\phi,\hat{\phi}) = \mathbb{E}_{p_{\text{data}}(\mathbf{x},\mathbf{y})}\mathbb{E}_{q(z|\mathbf{x})}\left[\log \frac{p_{\text{model}}(\tilde{\mathbf{x}}|z)\hat{q}(z|\mathbf{y})}{q(z|\mathbf{x})}\right] \tag{16}$$

$$\leq \mathbb{E}_{p_{\text{data}}(\mathbf{x},\mathbf{y})}\left[\log \mathbb{E}_{q(z|\mathbf{x})}\left[\frac{p_{\text{model}}(\tilde{\mathbf{x}}|z)\hat{q}(z|\mathbf{y})}{q(z|\mathbf{x})}\right]\right] \tag{17}$$

$$= \mathbb{E}_{p_{\text{data}}(\mathbf{x},\mathbf{y})}\left[\log \mathbb{E}_{\hat{q}(z|\mathbf{y})}\left[p_{\text{model}}(\tilde{\mathbf{x}}|z)\right]\right] \tag{18}$$

$$= \mathbb{E}_{p_{\text{data}}(\mathbf{x},\mathbf{y})}\left[\log p_{\text{cond}}(\tilde{\mathbf{x}}|\mathbf{y})\right], \tag{19}$$

where the final step follows from the definition of $p_{\text{cond}}$ in Eq. (4).

## C PROOF AND DISCUSSION OF THEOREM 3.1

### C.1 PROOF

**Statement.** *For any set $\hat{\Phi}$ of permissible values of $\hat{\phi}$, and for any $\theta \in \Theta$ and $\phi \in \Phi$,*

$$\arg\max_{\hat{\phi}\in\hat{\Phi}} \mathcal{O}_{\text{for}}(\theta,\phi,\hat{\phi}) = \arg\min_{\hat{\phi}\in\hat{\Phi}} \mathbb{E}_{p_{\text{data}}(\mathbf{y})}\left[KL\big(r(z|\mathbf{y}) \,\big\|\, \hat{q}(z|\mathbf{y})\big)\right]. \tag{8}$$

**Proof.** We can decompose $\mathcal{O}_{\text{for}}$ as follows. Starting by multiplying both sides of the fraction by the intractable conditional distribution $r(z|\mathbf{y})$,

$$\mathcal{O}_{\text{for}}(\theta,\phi,\hat{\phi}) = \mathbb{E}_{r(z,\mathbf{x},\mathbf{y})}\left[\log \frac{p_{\text{model}}(\tilde{\mathbf{x}}|z)\hat{q}(z|\mathbf{y})r(z|\mathbf{y})}{q(z|\mathbf{x})r(z|\mathbf{y})}\right] \tag{20}$$

$$= \mathbb{E}_{r(z,\mathbf{x},\mathbf{y})}\left[\log \frac{p_{\text{model}}(\tilde{\mathbf{x}}|z)r(z|\mathbf{y})}{q(z|\mathbf{x})}\right] - \mathbb{E}_{r(z,\mathbf{y})}\left[\log \frac{r(z|\mathbf{y})}{\hat{q}(z|\mathbf{y})}\right] \tag{21}$$

$$= C(\theta,\phi) - \mathbb{E}_{p_{\text{data}}(\mathbf{y})}\left[\text{KL}\big(r(z|\mathbf{y}) \,\big\|\, \hat{q}(z|\mathbf{y})\big)\right] \tag{22}$$

where $C(\theta,\phi)$ is a grouping of terms which do not depend on the partial encoder's parameters $\hat{\phi}$. Any values of $\hat{\phi}$ which maximise $\mathcal{O}_{\text{for}}(\theta,\phi,\hat{\phi})$ will therefore also minimise $\mathbb{E}_{p_{\text{data}}(\mathbf{y})}\left[\text{KL}\big(r(z|\mathbf{y}) \,\big\|\, \hat{q}(z|\mathbf{y})\big)\right]$ for the same values of $\theta$ and $\phi$, which proves the theorem.

### C.2 BOUND ON $\tilde{\mathbf{x}}$-SPACE DIVERGENCE

The mass-covering KL divergence in Theorem 3.1 indicates that samples $z \sim \hat{q}(\cdot|\mathbf{y})$ will be diverse (when $r(z|\mathbf{y})$ is diverse/uncertain). One would expect that samples of $\tilde{\mathbf{x}}$ conditioned on these diverse samples of $z$ will also be diverse. Here we make a more formal argument by showing that minimising $\mathcal{O}_{\text{for}}$ corresponds to miminising an upper-bound on an expected mass-covering KL divergence in $\tilde{\mathbf{x}}$-space. Starting from the result in Appendix B.2,

$$\mathcal{O}_{\text{for}}(\theta,\phi,\hat{\phi}) \leq \mathbb{E}_{p_{\text{data}}(\mathbf{x},\mathbf{y})}\left[\log p_{\text{cond}}(\tilde{\mathbf{x}}|\mathbf{y})\right] \tag{23}$$

$$= \mathbb{E}_{p_{\text{data}}(\mathbf{x},\mathbf{y})}\left[\log p_{\text{data}}(\tilde{\mathbf{x}}|\mathbf{y})\right] - \mathbb{E}_{p_{\text{data}}(\mathbf{y})}\left[\text{KL}(p_{\text{data}}(\tilde{\mathbf{x}}|\mathbf{y}) \,\|\, p_{\text{cond}}(\tilde{\mathbf{x}}|\mathbf{y}))\right]. \tag{24}$$

Rearranging this gives

$$\mathbb{E}_{p_{\text{data}}(\mathbf{y})}\left[\text{KL}(p_{\text{data}}(\tilde{\mathbf{x}}|\mathbf{y}) \,\|\, p_{\text{cond}}(\tilde{\mathbf{x}}|\mathbf{y}))\right] \leq \mathbb{E}_{p_{\text{data}}(\mathbf{x},\mathbf{y})}\left[\log p_{\text{data}}(\tilde{\mathbf{x}}|\mathbf{y})\right] - \mathcal{O}_{\text{for}}(\theta,\phi,\hat{\phi}). \tag{25}$$

Maximising $\mathcal{O}_{\text{for}}$ therefore minimises an upper-bound on an expected mass-covering KL divergence in $\tilde{\mathbf{x}}$-space, providing a further suggestion that $\tilde{\mathbf{x}} \sim p_{\text{cond}}(\cdot|\mathbf{y})$ will be diverse.

# D PROOF AND DISCUSSION OF THEOREM 3.2

## D.1 PROOF

**Statement.** *Assume that we have a sufficiently expressive encoder and decoder that there exist parameters $\theta^* \in \Theta$ and $\phi^* \in \Phi$ which make the unconditional VAE objective (Eq. (1)) equal to its upper bound of $-\mathcal{H}\left[p_{\text{data}}(\mathbf{x})\right]$. Assume also that the mutual information $I^*_{\tilde{\mathbf{x}},\mathbf{y}|z} := \mathbb{E}_{p_{\text{model}}(\tilde{\mathbf{x}},\mathbf{y},z;\theta^*)}\left[\log \frac{p_{\text{model}}(\tilde{\mathbf{x}},\mathbf{y}|z;\theta^*)}{p_{\text{model}}(\tilde{\mathbf{x}}|z;\theta^*)p_{\text{model}}(\mathbf{y}|z;\theta^*)}\right] = \mathbb{E}_{p_{\text{model}}(\tilde{\mathbf{x}},\mathbf{y},z;\theta^*)}\left[\log \frac{p_{\text{model}}(\tilde{\mathbf{x}}|z,\mathbf{y};\theta^*)}{p_{\text{model}}(\tilde{\mathbf{x}}|z;\theta^*)}\right]$ is zero (meaning that $\tilde{\mathbf{x}}$ and $\mathbf{y}$ are conditionally independent given $z$) and $\tilde{\mathbf{x}}$ is defined such that there exists a mapping from $(\tilde{\mathbf{x}}, \mathbf{y})$ to $\mathbf{x}$. Then, given a sufficiently expressive partial encoder $\hat{q}(z|\mathbf{y};\hat{\phi})$,*

$$\max_{\hat{\phi}} \mathcal{O}_{\text{for}}(\theta^*, \phi^*, \hat{\phi}) = \max_{\theta,\phi,\hat{\phi}} \mathcal{O}_{\text{for}}(\theta, \phi, \hat{\phi}). \tag{9}$$

**Proof.** We have used, and will continue to use, $p_{\text{model}}(\cdot)$ and $q(\cdot)$ to refer to joint distributions parameterised by $\theta$ and $\phi$ respectively. Since $\theta^*$ and $\phi^*$ are defined as the maximisers of the unconditional ELBO in Eq. (2), and given our assumption of sufficient expressivity, we have $p_{\text{model}}(z, \mathbf{x}; \theta^*) = p_{\text{data}}(\mathbf{x})q(z|\mathbf{x};\phi^*)$. We can therefore introduce notation for the equal joint distributions: $p^*(z, \mathbf{x}, \mathbf{y}) = p_{\text{model}}(z, \mathbf{x}, \mathbf{y}; \theta^*) = p_{\text{data}}(\mathbf{x}, \mathbf{y})q(z|\mathbf{x};\phi^*)$. We note that we can also write the joint distribution $p^*(z, \mathbf{x}, \tilde{\mathbf{x}}, \mathbf{y}) = p_{\text{data}}(\mathbf{x}, \mathbf{y})q(z|\mathbf{x};\phi^*)p(\tilde{\mathbf{x}}|\mathbf{x}, \mathbf{y})$, where $p(\tilde{\mathbf{x}}|\mathbf{x}, \mathbf{y}) = \delta_{g(\mathbf{x},\mathbf{y})}(\tilde{\mathbf{x}})$ is the Dirac distribution mapping $\mathbf{x}$ and $\mathbf{y}$ deterministically to $\tilde{\mathbf{x}} = g(\mathbf{x}, \mathbf{y})$. Recalling that we define $\tilde{\mathbf{x}}$ as either equal to $\mathbf{x}$ or as the dimensions of $\mathbf{x}$ not observed in $\mathbf{y}$, such a deterministic $g$ will always exist.

We consider the decomposition of $\mathcal{O}_{\text{for}}$ shown in Eq. (22) and remind the reader that, given a sufficiently expressive partial encoder, the maximization over $\hat{\phi}$ will always make the KL divergence zero. That is,

$$\max_{\hat{\phi}} \mathcal{O}_{\text{for}}(\theta, \phi, \hat{\phi}) = C(\theta, \phi) = \mathbb{E}_{r(z,\mathbf{x},\mathbf{y})}\left[\log \frac{p_{\text{model}}(\tilde{\mathbf{x}}|z)r(z|\mathbf{y})}{q(z|\mathbf{x})}\right] \tag{26}$$

for any $\theta$ and $\phi$. Using $\theta^*$ and $\phi^*$, and therefore using $p^*$ in place of $p_{\text{model}}$, $r$ and $q$, we can expand the left-hand side of Eq. (9).

$$\begin{aligned}
\max_{\hat{\phi}} \mathcal{O}_{\text{for}}(\theta^*, \phi^*, \hat{\phi}) &= \mathbb{E}_{p^*(z,\mathbf{x},\mathbf{y})}\left[\log \frac{p^*(\tilde{\mathbf{x}}|z)p^*(z|\mathbf{y})}{p^*(z|\mathbf{x})}\right] && (27)\\
&= \mathbb{E}_{p^*(z,\mathbf{x},\mathbf{y})}\left[\log \frac{p^*(\tilde{\mathbf{x}}|z,\mathbf{y})p^*(z|\mathbf{y})}{p^*(z|\mathbf{x})}\right] - \mathbb{E}_{p^*(z,\tilde{\mathbf{x}},\mathbf{y})}\left[\log \frac{p^*(\tilde{\mathbf{x}}|z,\mathbf{y})}{p^*(\tilde{\mathbf{x}}|z)}\right] && (28)\\
&= \mathbb{E}_{p^*(z,\mathbf{x},\mathbf{y})}\left[\log \frac{p^*(\tilde{\mathbf{x}}|z,\mathbf{y})p^*(z|\mathbf{y})}{p^*(z|\mathbf{x})}\right] - I^*_{\tilde{\mathbf{x}},\mathbf{y}|z} && (29)\\
&= \mathbb{E}_{p^*(z,\mathbf{x},\mathbf{y})}\left[\log \frac{p^*(z,\tilde{\mathbf{x}}|\mathbf{y})}{p^*(z|\mathbf{x})}\right] - I^*_{\tilde{\mathbf{x}},\mathbf{y}|z} && (30)\\
&= \mathbb{E}_{p^*(z,\mathbf{x},\mathbf{y})}\left[\log \frac{p^*(z|\tilde{\mathbf{x}},\mathbf{y})p^*(\tilde{\mathbf{x}}|\mathbf{y})}{p^*(z|\mathbf{x})}\right] - I^*_{\tilde{\mathbf{x}},\mathbf{y}|z}. && (31)
\end{aligned}$$

The factorisation $p^*(z, \mathbf{x}, \tilde{\mathbf{x}}, \mathbf{y}) = p_{\text{data}}(\mathbf{x}, \mathbf{y})q(z|\mathbf{x};\phi^*)\delta(\tilde{\mathbf{x}}|\mathbf{x}, \mathbf{y})$ implies that we can express $p^*(z|\tilde{\mathbf{x}}, \mathbf{y})$ with a marginalisation: $p^*(z|\tilde{\mathbf{x}}, \mathbf{y}) = \int p^*(z|\mathbf{x})p^*(\mathbf{x}|\tilde{\mathbf{x}}, \mathbf{y})\mathrm{d}\mathbf{x}$. Since we have assumed that there is a deterministic function $f$ mapping $(\tilde{\mathbf{x}}, \mathbf{y})$ to $\mathbf{x}$, $p^*(\mathbf{x}|\tilde{\mathbf{x}}, \mathbf{y})$ is a Dirac distribution and the marginalisation becomes $p^*(z|\tilde{\mathbf{x}}, \mathbf{y}) = \int p^*(z|\mathbf{x})\delta(\mathbf{x}|\tilde{\mathbf{x}}, \mathbf{y})\mathrm{d}\mathbf{x} = p^*(z|\mathbf{x} = f(\tilde{\mathbf{x}}, \mathbf{y}))$. We therefore proceed, rewriting $p^*(z|\tilde{\mathbf{x}}, \mathbf{y})$ as $p^*(z|\mathbf{x})$:

$$\begin{aligned}
\max_{\hat{\phi}} \mathcal{O}_{\text{for}}(\theta^*, \phi^*, \hat{\phi}) &= \mathbb{E}_{p^*(z,\mathbf{x},\mathbf{y})}\left[\log \frac{p^*(z|\mathbf{x})p^*(\tilde{\mathbf{x}}|\mathbf{y})}{p^*(z|\mathbf{x})}\right] - I^*_{\tilde{\mathbf{x}},\mathbf{y}|z} && (32)\\
&= \mathbb{E}_{p^*(\mathbf{x},\mathbf{y})}\left[\log p^*(\tilde{\mathbf{x}}|\mathbf{y})\right] - I^*_{\tilde{\mathbf{x}},\mathbf{y}|z} && (33)\\
&= \mathbb{E}_{p_{\text{data}}(\mathbf{x},\mathbf{y})}\left[\log p_{\text{data}}(\tilde{\mathbf{x}}|\mathbf{y})\right] - I^*_{\tilde{\mathbf{x}},\mathbf{y}|z}. && (34)
\end{aligned}$$

Now that we have this expansion of the left-hand side of Eq. (9), we can derive a related upper-bound on the right-hand side as follows. For any $\theta$, any $\phi$, and any $\hat{\phi}$, we have from Appendix B.2:

$$\begin{aligned}
\mathcal{O}_{\text{for}}(\theta, \phi, \hat{\phi}) &\leq \mathbb{E}_{p_{\text{data}}(\mathbf{x},\mathbf{y})} \left[\log p_{\text{cond}}(\tilde{\mathbf{x}}|\mathbf{y})\right] && (35)\\
&= \mathbb{E}_{p_{\text{data}}(\mathbf{x},\mathbf{y})} \left[\log p_{\text{data}}(\tilde{\mathbf{x}}|\mathbf{y})\right] - \mathbb{E}_{p_{\text{data}}(\mathbf{y})} \left[\text{KL}(p_{\text{data}}(\tilde{\mathbf{x}}|\mathbf{y}) \parallel p_{\text{cond}}(\tilde{\mathbf{x}}|\mathbf{y}))\right] && (36)\\
&\leq \mathbb{E}_{p_{\text{data}}(\mathbf{x},\mathbf{y})} \left[\log p_{\text{data}}(\tilde{\mathbf{x}}|\mathbf{y})\right]. && (37)
\end{aligned}$$

This bound must also hold for the maximum over $\theta$, $\phi$, and $\hat{\phi}$, and so is an upper-bound on the right-hand side of Eq. (9):

$$\max_{\theta,\phi,\hat{\phi}} \mathcal{O}_{\text{for}}(\theta, \phi, \hat{\phi}) \leq \mathbb{E}_{p_{\text{data}}(\mathbf{x},\mathbf{y})} \left[\log p_{\text{data}}(\tilde{\mathbf{x}}|\mathbf{y})\right]. \tag{38}$$

By relating Eq. (34) and Eq. (38), we have:

$$\max_{\theta,\phi,\hat{\phi}} \mathcal{O}_{\text{for}}(\theta, \phi, \hat{\phi}) \leq \max_{\hat{\phi}} \mathcal{O}_{\text{for}}(\theta^*, \phi^*, \hat{\phi}) + I^*_{\tilde{\mathbf{x}},\mathbf{y}|z}. \tag{39}$$

We then point out that we can also obtain an inequality in the other direction: since the right-hand side of Eq. (9) is a maximisation of the left-hand side over $\theta$ and $\phi$, it must be greater than or equal to the left-hand side:

$$\max_{\hat{\phi}} \mathcal{O}_{\text{for}}(\theta^*, \phi^*, \hat{\phi}) \leq \max_{\theta,\phi,\hat{\phi}} \mathcal{O}_{\text{for}}(\theta, \phi, \hat{\phi}). \tag{40}$$

Combining Eq. (39) and Eq. (40), we have:

$$0 \leq \max_{\theta,\phi,\hat{\phi}} \mathcal{O}_{\text{for}}(\theta, \phi, \hat{\phi}) - \max_{\hat{\phi}} \mathcal{O}_{\text{for}}(\theta^*, \phi^*, \hat{\phi}) \leq I^*_{\tilde{\mathbf{x}},\mathbf{y}|z}. \tag{41}$$

We now have bounds in both directions on the difference between the right- and left-hand sides. To conclude the proof, we repeat the assumption that $I^*_{\tilde{\mathbf{x}},\mathbf{y}|z} = 0$. The upper- and lower- bounds of Eq. (41) are therefore both zero, so we have the final result that $\max_{\hat{\phi}} \mathcal{O}_{\text{for}}(\theta^*, \phi^*, \hat{\phi}) = \max_{\theta,\phi,\hat{\phi}} \mathcal{O}_{\text{for}}(\theta, \phi, \hat{\phi})$.

### D.2 Discussion of the assumption that $I^*_{\tilde{\mathbf{x}},\mathbf{y}|z} = 0$

The above proof of Theorem 3.2 relies on the assumption that $I^*_{\tilde{\mathbf{x}},\mathbf{y}|z} = \mathbb{E}_{p^*(\tilde{\mathbf{x}},\mathbf{y},z)} \left[\log \frac{p^*(\tilde{\mathbf{x}}|z,\mathbf{y})}{p^*(\tilde{\mathbf{x}}|z)}\right] = 0$. That is, under the joint distribution $p^*(z, \mathbf{x}, \mathbf{y})$, $\tilde{\mathbf{x}}$ and $\mathbf{y}$ should be conditionally independent given $z$. We now argue that this is true for image completion, and "close to" satisfied for many other types of conditional generation.

**Image completion** In the VAE architecture we use, the likelihood $p_{\text{model}}(\mathbf{x}|z)$ is pixel-wise independent. In other words, if we write $\mathbf{x}$ as a set of pixels $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, we can factorise $p_{\text{model}}(\mathbf{x}|z)$ as $\prod_{i=1}^N p_{\text{model}}(\mathbf{x}_i|z)$. This means that the conditional mutual information between any two disjoint subsets of $\mathbf{x}$ (conditioned on $z$) will be zero. Given our definition of $\tilde{\mathbf{x}}$ for image completion as the the set of purely unobserved pixels, we have by construction that $\mathbf{x}$ and $\mathbf{y}$ are disjoint. Therefore, the assumption that $I^*_{\tilde{\mathbf{x}},\mathbf{y}|z} = 0$ is satisfied for image completion.

**When $I^*_{\tilde{\mathbf{x}},\mathbf{y}|z}$ is non-zero.** When $I^*_{\tilde{\mathbf{x}},\mathbf{y}|z}$ is non-zero, we can still use it as an upper-bound on the difference between the left- and right-hand side of Eq. (9) using Eq. (41). Moreover, we posit that $I^*_{\tilde{\mathbf{x}},\mathbf{y}|z}$ will typically be small when $\mathbf{y}$ represents high-level features (e.g. the edges in Edges2Photos or a class label) and we have a pixel-wise independent likelihood. This is because any high-level structure cannot be modelled by the pixel-wise independent likelihood $p^*(\tilde{\mathbf{x}}|z)$ and so must be captured by $z$. Intuitively, high-level features $\mathbf{y}$ are unlikely to be informative about any pixel-level variations if all image structure spanning multiple pixels ($z$) is already known. This is equivalent to saying that $I^*_{\tilde{\mathbf{x}},\mathbf{y}|z}$, the mutual information between $\tilde{\mathbf{x}}$ and $\mathbf{y}$ conditioned on $z$, is likely to be small.

## E AN ALTERNATIVE TRAINING OBJECTIVE

In this section, we describe an alternative objective previously used by Ma et al. (2018). We find experimentally that its performance is typically worse than $\mathcal{O}_{\text{for}}$, but it has the advantage of only requiring data $\mathbf{y} \sim p_{\text{data}}(\cdot)$ during training, and not $\mathbf{x}$. This may allow training in settings such as outpainting (Sabini & Rusak, 2018). The objective can be interpreted as a lower bound on $\log p_{\text{model}}(\mathbf{y})$:

$$\mathcal{O}_{\text{rev}}(\theta, \phi, \hat{\phi}) = \mathbb{E}_{p_{\text{data}}(\mathbf{y})} \mathbb{E}_{\hat{q}(z|\mathbf{y})} \left[ \log \frac{p_{\text{model}}(z, \mathbf{y})}{\hat{q}(z|\mathbf{y})} \right] \leq \mathbb{E}_{p_{\text{data}}(\mathbf{y})} \left[ \log p_{\text{model}}(\mathbf{y}) \right]. \tag{42}$$

Computing this objective requires a computation graph similar to that shown in Fig. 2c. Note one caveat with this objective is that using it involves computing and differentiating $p_{\text{model}}(z, \mathbf{y})$, which requires a tractable likelihood $p_{\text{data}}(\mathbf{y}|\mathbf{x})$ (based on the definition of $p_{\text{model}}$ in Table 2). This is possible in image completion, but not in the general case of conditional generation.

The following theorem says that learning $\hat{\phi}$ to maximise this objective is equivalent to minimising another KL divergence.

**Theorem E.1.** *For any set $\hat{\Phi}$ of permissible values of $\hat{\phi}$, and for any $\theta \in \Theta$ and $\phi \in \Phi$,*

$$\arg\max_{\hat{\phi} \in \hat{\Phi}} \mathcal{O}_{rev}(\theta, \phi, \hat{\phi}) = \arg\min_{\hat{\phi} \in \hat{\Phi}} \mathbb{E}_{p_{\text{data}}(\mathbf{y})} \left[ KL\big(\hat{q}(z|\mathbf{y}; \hat{\phi}) \,\big\|\, p_{\text{model}}(z|\mathbf{y}; \theta)\big) \right]. \tag{43}$$

**Proof.** We show this by decomposing the objective:

$$\begin{aligned}
\mathcal{O}_{\text{rev}}(\theta, \phi, \hat{\phi}) &= \mathbb{E}_{p_{\text{data}}(\mathbf{y}) \hat{q}(z|\mathbf{y})} \left[ \log \frac{p_{\text{model}}(z, \mathbf{y})}{\hat{q}(z|\mathbf{y})} \right] && (44) \\
&= \mathbb{E}_{p_{\text{data}}(\mathbf{y}) \hat{q}(z|\mathbf{y})} \left[ \log p_{\text{model}}(\mathbf{y}) - \log \frac{\hat{q}(z|\mathbf{y})}{p_{\text{model}}(z|\mathbf{y})} \right] && (45) \\
&= \mathbb{E}_{p_{\text{data}}(\mathbf{y})} \left[ \log p_{\text{model}}(\mathbf{y}) - \text{KL}\big(\hat{q}(z|\mathbf{y}) \,\big\|\, p_{\text{model}}(z|\mathbf{y})\big) \right] && (46) \\
&= D(\theta) - \mathbb{E}_{p_{\text{data}}(\mathbf{y})} \left[ \text{KL}\big(\hat{q}(z|\mathbf{y}) \,\big\|\, p_{\text{model}}(z|\mathbf{y})\big) \right] && (47)
\end{aligned}$$

where $D(\theta)$ is not dependent on $\hat{\phi}$. Therefore, for any $\theta$ and $\phi$, any values of $\hat{\phi} \in \hat{\Phi}$ which maximise $\mathcal{O}_{\text{rev}}(\theta, \phi, \hat{\phi})$ will also minimise $\mathbb{E}_{p_{\text{data}}(\mathbf{y})} \left[ \text{KL}\big(\hat{q}(z|\mathbf{y}) \,\big\|\, p_{\text{model}}(z|\mathbf{y})\big) \right]$, proving the theorem.

Theorem E.1 implies that maximising $\mathcal{O}_{\text{rev}}$ will minimise a "reverse" KL divergence, causing $\hat{q}(z|\mathbf{y})$ to exhibit mode-seeking behaviour. We denote the method of training with this objective IPA-R (inference in a pretrained artifact with the reverse KL). As with IPA, we use pretrained $\theta$ and $\phi$ when training IPA-R, which we justify in the following paragraph. With IPA-R, we also use the pretrained encoder parameters $\phi$ as an initialisation for $\hat{\phi}$, which makes training more stable.

To justify the use of a pretrained $\theta$ and $\phi$ with $\mathcal{O}_{\text{rev}}$, we show that the objective $\mathcal{O}_{\text{rev}}$ has a property analogous to that described by Theorem 3.2.

**Theorem E.2.** *Assume we have a sufficiently expressive encoder and decoder such that there exists parameters $\theta^* \in \Theta$ and $\phi^* \in \Phi$ which make the unconditional VAE objective (Eq.* (1)*) equal to its upper bound of $-\mathcal{H}\left[p_{\text{data}}(\mathbf{x})\right]$. Then, given a sufficiently expressive partial encoder $\hat{q}(z|\mathbf{y}; \hat{\phi})$,*

$$\max_{\hat{\phi}} \mathcal{O}_{rev}(\theta^*, \phi^*, \hat{\phi}) = \max_{\theta, \phi, \hat{\phi}} \mathcal{O}_{rev}(\theta, \phi, \hat{\phi}).$$

**Proof.** We will prove Theorem E.2 by showing that both the quantity on the left-hand side and the quantity on the right-hand side are equal to the negative of the entropy of $p_{\text{data}}(\mathbf{x})$. We begin with the left-hand side,

$$\max_{\hat{\phi}} \mathcal{O}_{\text{rev}}(\theta^*, \phi^*, \hat{\phi}) = \max_{\hat{\phi}} \mathbb{E}_{p_{\text{data}}(\mathbf{y}) \hat{q}(z|\mathbf{y})} \left[ \log \frac{p^*(z, \mathbf{y})}{\hat{q}(z|\mathbf{y})} \right]. \tag{48}$$

This is a variational lower-bound which is made tight by setting $\hat{\phi}$ such that $\hat{q}(z|\mathbf{y}) = p^*(z|\mathbf{y})$:

$$\max_{\hat{\phi}} \mathcal{O}_{\text{rev}}(\theta^*, \phi^*, \hat{\phi}) = \mathbb{E}_{p_{\text{data}}(\mathbf{y})} \left[\log p^*(\mathbf{y})\right] = -\mathcal{H}\left[p_{\text{data}}(\mathbf{y})\right]. \tag{49}$$

Now we consider the right-hand side.

$$\max_{\theta,\phi,\hat{\phi}} \mathcal{O}_{\text{rev}}(\theta, \phi, \hat{\phi}) = \max_{\theta,\phi,\hat{\phi}} \mathbb{E}_{p_{\text{data}}(\mathbf{y})\hat{q}(z|\mathbf{y})} \left[\log \frac{p_{\text{model}}(z, \mathbf{y})}{\hat{q}(z|\mathbf{y})}\right] \tag{50}$$

This is another variational lower-bound made tight by setting $\hat{\phi}$ such that $\hat{q}(z|\mathbf{y}) = p_{\text{model}}(z|\mathbf{y})$:

$$\max_{\theta,\phi,\hat{\phi}} \mathcal{O}_{\text{rev}}(\theta, \phi, \hat{\phi}) = \max_{\theta,\phi} \mathbb{E}_{p_{\text{data}}(\mathbf{y})} \left[\log p_{\text{model}}(\mathbf{y})\right] \tag{51}$$

$$= -\mathcal{H}\left[p_{\text{data}}(\mathbf{y})\right]. \tag{52}$$

The right-hand side is therefore equivalent to the left-hand-side as shown in Eq. (49), proving the theorem.

## F ESTIMATING KL DIVERGENCES IN PRACTICE

In this section we describe how we compute unbiased and low-variance estimates of $\mathcal{O}_{\text{for}}$ and $\mathcal{O}_{\text{rev}}$ in practice. For both, we use similar techniques to those commonly used in unconditional hierarchical VAEs. First note that each can be written as the sum of a likelihood and KL divergence term. For the unconditional VAE objective, $\mathcal{O}_{\text{for}}$ and $\mathcal{O}_{\text{rev}}$ respectively:

$$\mathbb{E}_{p_{\text{data}}(\mathbf{x})} \left[\text{ELBO}(\theta, \phi, \mathbf{x})\right] = \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \left[\mathbb{E}_{q(z|\mathbf{x})} \left[\log p_{\text{model}}(\mathbf{x}|z)\right] - \text{KL}\big(q(z|\mathbf{x}) \,\big\|\, p_{\text{model}}(z)\big)\right], \tag{53}$$

$$\mathcal{O}_{\text{for}}(\theta, \phi, \hat{\phi}) = \mathbb{E}_{p_{\text{data}}(\mathbf{x},\mathbf{y})} \left[\mathbb{E}_{q(z|\mathbf{x})} \left[\log p_{\text{model}}(\tilde{\mathbf{x}}|z)\right] - \text{KL}(q(z|\mathbf{x}) \,\|\, \hat{q}(z|\mathbf{y}))\right], \tag{54}$$

$$\mathcal{O}_{\text{rev}}(\theta, \phi, \hat{\phi}) = \mathbb{E}_{p_{\text{data}}(\mathbf{y})} \mathbb{E}_{\hat{q}(z|\mathbf{y})} \left[\log p_{\text{model}}(\mathbf{y}|z) - \text{KL}(\hat{q}(z|\mathbf{y}) \,\|\, p_{\text{model}}(z))\right]. \tag{55}$$

In a non-hierarchical VAE with Gaussian $p_{\text{model}}(z)$ and $q(z|\mathbf{x})$, it is common to compute the KL divergence term analytically in order to reduce the variance of the estimate. In hierarchical VAEs where both $p_{\text{model}}(z)$ and $q(z|\mathbf{x})$ consists of multiple Gaussian distributions with non-linear dependencies, it is not possible to compute the KL divergence analytically. However, we can still make use of analytic estimates of the KL divergence at each layer. For the unconditional VAE objective, an estimator can be derived as follows:

$$\text{KL}\big(q(z|\mathbf{x}) \,\big\|\, p_{\text{model}}(z)\big) = \mathbb{E}_{q(z|\mathbf{x})} \left[\log \frac{q(z|\mathbf{x})}{p_{\text{model}}(z)}\right] \tag{56}$$

$$= \mathbb{E}_{q(z|\mathbf{x})} \left[\log \frac{\prod_{l=1}^{L} q(z_l|z_{<l}, \mathbf{x})}{\prod_{l=1}^{L} p_{\text{model}}(z_l|z_{<l})}\right] \tag{57}$$

$$= \mathbb{E}_{q(z|\mathbf{x})} \left[\sum_{l=1}^{L} \log \frac{q(z_l|z_{<l}, \mathbf{x})}{p_{\text{model}}(z_l|z_{<l})}\right] \tag{58}$$

$$= \mathbb{E}_{q(z|\mathbf{x})} \left[\sum_{l=1}^{L} \log \frac{q(z_l|z_{<l}, \mathbf{x})}{p_{\text{model}}(z_l|z_{<l})}\right] \tag{59}$$

$$= \sum_{l=1}^{L} \mathbb{E}_{q(z_{\leq l}|\mathbf{x})} \left[\log \frac{q(z_l|z_{<l}, \mathbf{x})}{p_{\text{model}}(z_l|z_{<l})}\right] \tag{60}$$

$$= \sum_{l=1}^{L} \mathbb{E}_{q(z_{<l}|\mathbf{x})} \left[\text{KL}\big(q(z_l|z_{<l}, \mathbf{x}) \,\big\|\, p_{\text{model}}(z_l|z_{<l})\big)\right]. \tag{61}$$

This is used by Vahdat & Kautz (2020); Child (2020) to estimate the KL divergence term for an unconditional hierarchical VAE, by sampling $z \sim q(\cdot|\mathbf{x})$ and then computing the resulting KL

divergence for each layer $l$ conditioned on $z_{<l}$. Similar derivations suggest the following estimates for the KL divergences in $\mathcal{O}_{\text{for}}$ and $\mathcal{O}_{\text{rev}}$:

$$\mathrm{KL}\big(q(z|\mathbf{x}) \,\big\|\, \hat{q}(z|\mathbf{y})\big) = \sum_{l=1}^{L} \mathbb{E}_{q(z_{<l}|\mathbf{x})} \left[\mathrm{KL}\big(q(z_l|z_{<l}, \mathbf{x}) \,\big\|\, \hat{q}(z_l|z_{<l}, \mathbf{y})\big)\right], \tag{62}$$

$$\mathrm{KL}\big(\hat{q}(z|\mathbf{y}) \,\big\|\, p_{\text{model}}(z)\big) = \sum_{l=1}^{L} \mathbb{E}_{\hat{q}(z_{<l}|\mathbf{y})} \left[\mathrm{KL}\big(\hat{q}(z_l|z_{<l}, \mathbf{y}) \,\big\|\, p_{\text{model}}(z_l|z_{<l})\big)\right]. \tag{63}$$

We compute unbiased estimates of the KL divergence terms by simply sampling $z$ (from $q(\cdot|\mathbf{x})$ for $\mathcal{O}_{\text{for}}$ or $\hat{q}(\cdot|\mathbf{y})$ for $\mathcal{O}_{\text{rev}}$) and analytically computing the resulting KL divergence for each layer $l$ conditioned on $z_{<l}$.

# G Experimental details

## G.1 IPA/IPA-R

**Architectures** The IPA encoder and decoder architectures we used for CIFAR-10 and FFHQ-256 were the same as those used by Child (2020) for the same datasets, with 45 and 62 groups of latent variables respectively. The encoder and decoder had 39 million parameters for CIFAR-10 and 115 million for FFHQ-256. We used partial encoders with structure identical to the encoders, other than an additional input channel to accept the concatenated mask. The partial encoders contained 18 million and 65 million parameters respectively for CIFAR-10 and FFHQ-256. The architecture used for the edges-to-photos experiments was the same as that used for ImageNet-64 by Child (2020). The architecture used for Chest X-ray 14 was identical to that used for FFHQ-256. We use the same architecture for both IPA and IPA-R.

**Training** On CIFAR-10, IPA was trained for 550 000 iterations ($\sim$115 hours) with a batch size of 30 on a single 16GB GeForce RTX 2080 Ti. IPA-R was trained for 250 000 iterations ($\sim$83 hours) with a batch size of 16 on a 32GB V100 GPU. The learning rates used were selected with a sweep over three values and were $2 \times 10^{-4}$ for IPA and $2 \times 10^{-5}$ for IPA-R. On FFHQ-256, IPA was trained for 626 000 iterations ($\sim$283 hours) with a batch size of 12. IPA-R was trained for 416 000 iterations ($\sim$200 hours) with a batch size of 8. Training was parallelised over 4 32GB V100 GPUs. We used learning rates of $1.5 \times 10^{-4}$ for IPA and $5 \times 10^{-5}$ for IPA-R. As for CIFAR-10, these learning rates were selected with a search over three values. Other hyperparameters are the same as those used by Child (2020) for training VAEs on these datasets. We use the Weights & Biases (Biewald, 2020) experiment-tracking infrastructure. Computational resources were obtained primarily from a cloud provider.

On the edges-to-photos datasets, all networks were trained on a single 16GB GeForce RTX 2080 Ti with the largest batch size that could fit into memory (4 for IPA and 1 for the from-scratch baselines). Learning rates were chosen with sweeps over three values, and were $2 \times 10^{-4}$ for IPA on both edges-to-photos datasets and $5 \times 10^{-5}$ for the from-scratch baseline on both edges-to-photos datasets.

The from-scratch baseline on CIFAR-10 was trained on a GeForce RTX 2080 Ti with a batch size of 9 (the largest that fit onto memory) and learning rate $2 \times 10^{-4}$ (chosen with a sweep over three values). The variation of IPA on CIFAR-10 with ImageNet pretraining used a learning rate of $1.5 \times 10^{-4}$ (chosen with a sweep over three values) but otherwise identical to hyperparameters to the version with CIFAR-10 pretraining. We trained the unconditional VAE for ImageNet-32 ourselves (again on a GeForce RTX 2080 Ti) for 14 days, using a batch size of 15 and learning rate of $2 \times 10^{-4}$ (chosen with a sweep over 2 values).

IPA and the from-scratch baseline on ImageNet64 are trained on a single 16GB GeForce RTX 2080 Ti with a batch size of 4 and a learning rate of $5 \times 10^{-5}$, selected with a sweep over 3 values. The samples shown in Appendix L are taken after 102 hours of training for each.

### G.2 BASELINES

For all the baselines, we based our implementations on publicly available (official or unofficial) implementations. A link to these repositories is provided for each baseline. All the training procedures were modified to use the same distribution of partially observed images as for training IPA (see Section 4).

**CoModGAN** We used the official implementation of Zhao et al. (2021)[1] and used their reported hyperparameters for this baseline. The networks for CIFAR-10 and FFHQ-256 had 82,097,684 and 108,041,635 parameters respectively. They were trained for 25 million iterations. Training took 134 hours on 2 GPUs for CIFAR-10 and 333 hours on 4 GPUs for FFHQ-256. All the GPUs used were Tesla V100s.

**PIC** We adapted the official implementation of Zheng et al. (2019)[2] for the FFHQ-256 and CIFAR-10 datasets. We used the same architecture, hyperparameters and objective functions as their implementation. The networks for both CIFAR-10 and FFHQ-256 had 9,128,390 parameters. They were trained using Adam optimizer (Kingma & Ba, 2015) with a batch size of 20, a learning rate of $10^{-4}$ and $\beta_1, \beta_2$ parameters of 0.0 and 0.999 respectively. The model for FFHQ-256 was trained for 600 epochs and the model for CIFAR-10 was trained for 800 epochs. It took 430 hours to train each of these models on a Tesla V100 GPU.

The PIC architecture is defined for $256 \times 256$ images. Therefore, in order to test it on the CIFAR-10 dataset, we resized the $32 \times 32$ CIFAR-10 images to $256 \times 256$ (via bilinear interpolation) before feeding them into the PIC model. We then down-sampled the inpainted images back to the original size of $32 \times 32$ for evaluation.

**ANP** Our ANP network architecture was based on that of Kim et al. (2019)[3], differing only in that we used hidden and latent dimensions of 512. Both the CIFAR-10 and FFHQ-256 models had 11,563,526 parameters and were trained for 250 epochs with a batch size of 16 using Adam optimizer. Learning rates of $5 \times 10^{-5}$ and $10^{-4}$ was used for the CIFAR-10 and FFHQ-256 models respectively. Training took 33 hours for CIFAR-10 and 43 hours for FFHQ-256 on a Tesla V100 GPU.

In the original image inpainting experiments of (Kim et al., 2019) the images are $32 \times 32$. Since ANPs embed each pixel separately and their self-attention and cross-attention layers attend to every other pixel in the observed and target sets, it is computationally expensive to scale them to larger images. We therefore downsampled the FFHQ-256 images to $64 \times 64$, and upsampled the inpainted image back to $256 \times 256$ via bilinear interpolation. Additionally, at training time, we randomly dropped half of the observed pixels (a.k.a. context set in neural process literature) to reduce the computational cost. Finally, the target set at training time was half of the unobserved pixels. For CIFAR-10, on the other hand, no modfication was done to the image resolution or observation masks i.e. images were fed in at $32 \times 32$ resolution, the context sets were the set of all observed pixels in partial images $\mathbf{y}$, and the target sets consisted of all unobserved pixels.

**CE** We used the same architecture as reported by Pathak et al. (2016)[4]. The models for CIFAR-10 and FFHQ-256 had 34,094,528 and 71,321,536 parameters respectively. Both models were trained for 500 epochs with a batch size of 64. We used Adam optimizer with a learning rate of $2 \times 10^{-4}$, $\beta_1 = 0.5$, and $\beta_2 = 0.999$. It took 14 hours to train the CIFAR-10 model and 46 hours to train the FFHQ-256 model on a RTX 2080 Ti GPU.

[1] https://github.com/zsyzzsoft/co-mod-gan

[2] https://github.com/lyndonzheng/Pluralistic-Inpainting

[3] Our implementation of ANP is based on https://github.com/EmilienDupont/neural-processes and https://github.com/YannDubs/Neural-Process-Family

[4] Our implementation of CE is based on https://github.com/BoyuanJiang/context_encoder_pytorch, with some modifications according to https://github.com/pathak22/context-encoder to support larger image sizes and non-centered observation masks.

**RFR** We adapted the official implementation of Li et al. (2020a)[5]. Similar to PIC, all the input images to this model were resized to $256 \times 256$. Both models had 31,224,065 parameters and were trained with a batch size of 8 and learning rate of $10^{-4}$ and other hyperparameters the same as reported in (Li et al., 2020a). The iterative way of completing images in this baseline makes it slow to complete images with most of their pixels missing. Therefore, after around 450 hours of training on a Tesla V100 GPU, they were were trained for only 120 and 85 epochs, respectively for CIFAR-10 and FFHQ-256. This may partly explain the poor performance of RFR that we report.

**VQ-VAE** We adapted the official implementation of Peng et al. (2021)[6] to train the model on FFHQ-256 and CIFAR-10 with freeform masks. Similar to PIC and RFR, all the input images to this model were resized to $256 \times 256$. To train this model, we used the hyperparameters reported in (Peng et al., 2021). The model has three modules: a VAE, a structure generator and a texture generator. Each is trained separately:

- The VAEs for both datasets were trained for 1 million iterations on an RTX 2080 Ti GPU. It took less than 13 hours for both datasets.
- The structure generator for CIFAR-10 was trained for 1 million iterations and the FFHQ-256 one was trained for 1.5 million iterations. It took 128 hours on a Tesla V100 and 212 hours on an RTX 2080 Ti, for CIFAR-10 and FFHQ-256 respectively.
- The texture generators were trained for 2 million traces for both datasets. It took 171 hours on a Tesla V100 and 211 hours on an RTX 2080 Ti GPU, for CIFAR-10 and FFHQ-256 respectively.

This model is very slow to complete images at inference time. The authors reported that it takes 45 seconds to complete a $256 \times 256$ image, and in our experiments we observed that this time could be up to 3 minutes on an RTX 2080 Ti GPU. This makes evaluating the quantitative metrics we use (FID, P-IDS, and LPIPS-GT) prohibitively expensive. We do, however, report qualitative results in Appendix L.

### G.3 METRICS

**FID for completed images** We use the FID score (Heusel et al., 2017) to quantify the distance between $p_{\text{data}}(\mathbf{x})$ and $\int p_{\text{cond}}(\mathbf{x}|\mathbf{y})p_{\text{data}}(\mathbf{y})\mathrm{d}\mathbf{y}$, the distribution resulting from sampling a dataset image, masking out some pixels, and replacing them by performing inpainting. This should be zero if $p_{\text{cond}}(\mathbf{x}|\mathbf{y}) = p_{\text{data}}(\mathbf{x}|\mathbf{y})$. Since this metric does not explicitly consider multiple completions of the same observation, we view it as a measure of image quality more than diversity.

**P-IDS** This metric was proposed by Zhao et al. (2021), who show that is correlates well with human evaluations. It measures the linear separability of dataset and inpainted images in a pretrained Inception network's feature space. Higher values are better, indicating that the data is less separable.

**LPIPS diversity score** This metric for diversity was proposed by Zhu et al. (2017), involving calculating the LPIPS distance between pairs of images sampled from $p_{\text{cond}}(\cdot|\mathbf{y})$ for the same $\mathbf{y}$. We report it here for completeness but it is not a good measure of the "fit" between $p_{\text{cond}}(\mathbf{x}|\mathbf{y})$ and $p_{\text{data}}(\mathbf{x}|\mathbf{y})$ since it can always be maximised by making $p_{\text{cond}}(\mathbf{x}|\mathbf{y})$ as diverse as possible, without reference to the "true" posterior $p_{\text{data}}(\mathbf{x}|\mathbf{y})$.

**Faithfulness weighted variance** The faithfulness weighted variance (FWV) was proposed by Li et al. (2020b), measuring both how diverse generated images $\mathbf{x} \sim p_{\text{cond}}(\cdot|\mathbf{y})$ are and how well they fit the ground truth. To compute it we first draw $N$ pairs of $(\mathbf{x}_i, \mathbf{y}_i)$ from the test set. Then, for each $\mathbf{y}_i$ draw $K$ conditional samples from the model $\hat{\mathbf{x}}_{i,j} \sim p_{\text{cond}}(\cdot|\mathbf{y}_i)$ and let $\bar{\mathbf{x}}_i$ be the pixel-wise average of $\{\hat{\mathbf{x}}_{i,j}\}_{j=1}^{K}$. Finally,

$$\text{FWV} = \sum_{i=1}^{N}\sum_{j=1}^{K} w_{i,j} d_{\text{LPIPS}}(\hat{\mathbf{x}}_{i,j}, \bar{\mathbf{x}}_i), \text{ where } w_{i,j} = \exp\left(-\frac{d_{\text{LPIPS}}(\hat{\mathbf{x}}_{i,j}, \mathbf{x}_i)}{2\sigma^2}\right). \tag{64}$$

[5] https://github.com/jingyuanli001/RFR-Inpainting

[6] https://github.com/USTC-JialunPeng/Diverse-Structure-Inpainting

In this paper we normalize FWV and report $\frac{\text{FWV}}{N}$. We use $N = K = 100$ and $\sigma = 0.2$.

However, this metric was designed for super-resolution and it is not clear how suitable it is for other conditional generation tasks due to the use of a pixel-space average (which may be less meaningful when the conditioning signal is less strong, and so sampled images more diverse).

# H BAYESIAN OPTIMAL EXPERIMENTAL DESIGN

## H.1 EIG ESTIMATORS

In this section, we expand on how our estimator for the expected information gain differs from prior work (Harvey et al., 2019). Repeating some relevant definitions from the main text, $f_{c_1,\ldots,c_t}$ is a function which maps from a full image to a sequence of observations of the image at locations $c_1, \ldots, c_t$. When estimating the EIG at time step $t$, we will have already observed a sequence of observations denoted $\mathbf{y}_{c_1,\ldots,c_{t-1}}$, extracted from some latent image $\mathbf{x}$ as $f_{c_1,\ldots,c_{t-1}}(\mathbf{x})$. We use a CNN which outputs $g(v|f_{c_1,\ldots,c_t}(\mathbf{x}))$, an approximation of the posterior over the latent variable of interest given a sequence of observations. Following Harvey et al. (2019), we will from now on refer to this as the AVP-CNN (attentional variational posterior CNN). Both our method and that of Harvey et al. (2019) sample image completions, which we denote $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}$, conditioned on $\mathbf{y}_{c_1,\ldots,c_{t-1}}$. Harvey et al. (2019) do so by retrieving images which roughly match the observed pixel values from a database. Although completions from this are diverse, they can match the observed values poorly. We therefore replace this stage with an image completion network trained using IPA.

Given these components, Harvey et al. (2019) approximate the expected information gain with

$$\text{EIG}(c_t; \mathbf{y}_{c_1,\ldots,c_{t-1}}) \approx \overbrace{\mathcal{H}\left[g(\cdot|\mathbf{y}_{1,\ldots,c-1})\right]}^{\text{entropy after } t-1 \text{ scans}} - \overbrace{\frac{1}{N}\sum_{n=1}^{N} \mathcal{H}\left[g(\cdot|f_{c_1,\ldots,c_t}(\mathbf{x}^{(n)}))\right]}^{\text{expected entropy after } t \text{ scans}}. \tag{65}$$

Since the prior entropy term of Eq. (65) does not depend on $c_t$, it can be neglected when choosing $c_t$ with BOED. As only the expected posterior entropy then needs to be estimated and compared for various $c_t$, we will from now on refer to this estimator with the acronym EPE (for 'expected posterior entropy').

We use a different estimator for the prior entropy. It becomes equivalent as $N \to \infty$ if $g$ and the image completion method are perfect but, when this is not the case, produces an estimate of the prior entropy with a dependence on $c_t$. To repeat Eq. (10), this results in the following estimator for the EIG:

$$\text{EIG}(c_t; \mathbf{y}_{c_1,\ldots,c_{t-1}}) \approx \overbrace{\mathcal{H}\left[\frac{1}{N}\sum_{n=1}^{N} g(\cdot|f_{c_1,\ldots,c_t}(\mathbf{x}^{(n)}))\right]}^{\text{entropy after } t-1 \text{ scans}} - \overbrace{\frac{1}{N}\sum_{n=1}^{N} \mathcal{H}\left[g(\cdot|f_{c_1,\ldots,c_t}(\mathbf{x}^{(n)}))\right]}^{\text{expected entropy after } t \text{ scans}}. \tag{66}$$

While it is not immediately clear that this estimator is better than that in Eq. (65), one useful property is that, like the true expected information gain, it is guaranteed to be non-negative. This can be seen as follows. First, note that the prior entropy term is the entropy of an expectation (over $n \sim \text{Uniform}(1, N)$), and the expected posterior entropy term swaps the order of the entropy and expectation. Since the mapping from a distribution $g(\cdot)$ to its entropy $\mathcal{H}[g(\cdot)]$ is strictly concave, Jensen's inequality is applicable. Invoking Jensen's inequality shows that the prior entropy term must be greater than or equal to the expected posterior entropy, and so the estimate of the EIG will be non-negative. In Fig. 6, we compare the performance of this estimator (denoted EIG) and that in Eq. (65) (EPE). We find that the estimator denoted EIG leads to significantly better classification performance.

Another view of our EIG estimator is as a variant of the inception score (Salimans et al., 2016) for the sampled images $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}$. The standard inception score is computed using an image classifier trained on ImageNet (Salimans et al., 2016). Equation (10) is the inception score if this classifier is replaced with the AVP-CNN acting on observations of each image at locations $c_1, \ldots, c_t$.

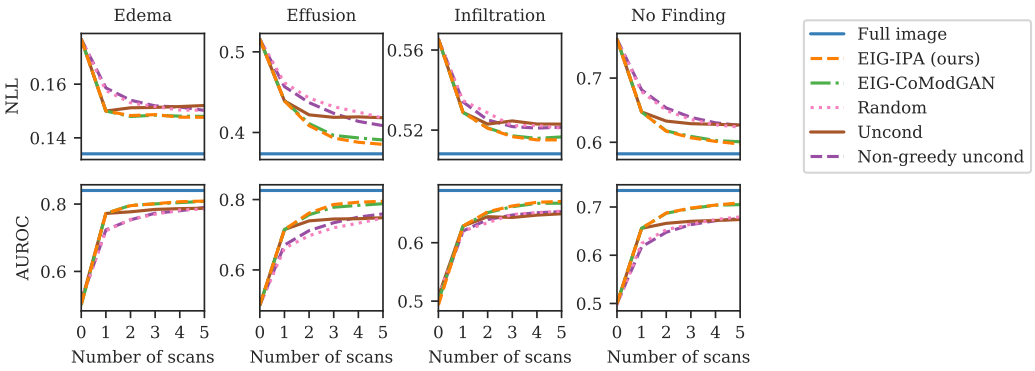



Figure 6: We extend on Fig. 5 by additionally reporting negative log-likelihoods (top row) for the classification tasks. We also include results from the 'EPE' estimator used by Harvey et al. (2019). We see that our BOED estimator provides significant improvements; resulting performance is always superior to the other baselines. This is only sometimes the case for the 'EPE' estimator.

with locations chosen to infer

| AUROC when predicting | Edema | Effusion | Infiltration | No Finding |
|---|---|---|---|---|
| Edema | 0.807 | 0.801 | 0.804 | 0.804 |
| Effusion | 0.783 | 0.791 | 0.781 | 0.781 |
| Infiltration | 0.674 | 0.671 | 0.675 | 0.675 |
| No Finding | 0.704 | 0.707 | 0.708 | 0.708 |

Figure 7: AUROC scores when performing one classification task with scan locations chosen for a different classification task. The intensity of the colour of each cell is proportional to the difference between its value and the greatest value in its row.

### H.2 BOED BASELINES

**Random** This baseline simply samples the scan location independently at each time step from a uniform distribution over all valid locations.

**Uncond** This baseline ablates our estimator for the EIG (Eq. (10)) by sampling images $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}$ from the training dataset (without conditioning on $\mathbf{y}_{c_1,\dots,c_{t-1}}$) instead of from IPA (conditioned on $\mathbf{y}_{c_1,\dots,c_{t-1}}$). That is, we use:

$$\mathrm{U}^{\text{uncond}}(c_t; c_1, \dots, c_{t-1}) = \mathcal{H}\left[\frac{1}{N}\sum_{n=1}^{N} g(\cdot|f_{c_1,\dots,c_t}(\mathbf{x}^{(n)}))\right] - \frac{1}{N}\sum_{n=1}^{N}\mathcal{H}\left[g(\cdot|f_{c_1,\dots,c_t}(\mathbf{x}^{(n)}))\right] \tag{67}$$

with $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)} \sim p(\mathbf{x})$. The resulting choice of scan location $c_t$ at each time step $t$ has no dependence on previous observations $\mathbf{y}_{c_1,\dots,c_{t-1}}$.

**Non-greedy uncond** When selecting scan locations that optimise the EIG, we select the scan location at each time step greedily to optimise the EIG from the next scan. To do otherwise (i.e. maximise the EIG from multiple next scans) is intractable because of the dependence of the EIG

at each $t$ on previous observations $\mathbf{y}_{c_1,\ldots,c_{t-1}}$. However, since the above 'Uncond' baseline selects each $c_t$ independently of what is observed at previous scans, we can remove this greedy property. We do so by selecting all $c_1, \ldots, c_T$ simultaneously to maximize

$$\mathrm{U}^{\text{non-greedy}}(c_1, \ldots, c_T) = \mathcal{H}\left[\frac{1}{N}\sum_{n=1}^{N} g(\cdot|f_\emptyset(\mathbf{x}^{(n)}))\right] - \frac{1}{N}\sum_{n=1}^{N}\mathcal{H}\left[g(\cdot|f_{c_1,\ldots,c_T}(\mathbf{x}^{(n)}))\right] \tag{68}$$

with $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)} \sim p(\mathbf{x})$. Since the number of possible values of $c_1, \ldots, c_T$ increases exponentially with $T$, it is not feasible to search over all of them. We therefore optimize this sequence by randomly sampling a set of such sequences, and choosing the best from this set.

### H.3 Additional BOED results

Figure 6 shows additional results from classification with sequences chosen by BOED and our baselines. In particular, we report negative log-likelihoods for the class labels as well as the AUROC scores. We see that the negative log-likelihood sometimes increases as more scans are taken, indicating that the classifier used is not perfectly trained. Interestingly, this only happens when glimpses are chosen with the EPE estimator of Harvey et al. (2019). This may be because the EPE estimator has a tendency to select locations which cause the classifier to be overconfident (and thus produce artificially low posterior entropies). With the EIG estimator, and the baselines, this issue is not observed.

To illustrate that the scan locations chosen by BOED are task-dependent, Fig. 7 shows the results from using scan locations chosen for one classification task (i.e. diagnosing a particular disease) to perform a different classification task. Each row shows the results for a particular task, with each column showing results when scan locations are chosen based on a particular task. As expected, the highest (or at least joint-highest) AUROC score in each row occurs on the diagonal, when the classification task and choice of scan locations are aligned. Away from the diagonal, the scores are only slightly lower, perhaps reflecting that there is a large overlap between the relevant areas for these tasks.

Figure 8 shows additional visualisations of the Bayesian experimental design process for different test images, adding to the one shown in the main paper. The 'information gains' shown for a particular $\mathbf{x}^*$ (sampled conditioned on observations $\mathbf{y}_{c_1,\ldots,c_{t-1}}$) are computed as the following KL divergence:

$$\mathrm{IG}(\mathbf{x}^*, c_t; \mathbf{y}_{c_1,\ldots,c_{t-1}}) \approx \mathrm{KL}\left(g(\cdot|f_{c_1,\ldots,c_t}(\mathbf{x}^*)) \;\middle\|\; \frac{1}{N}\sum_{n=1}^{N} g(\cdot|f_{c_1,\ldots,c_t}(\mathbf{x}^{(n)}))\right). \tag{69}$$

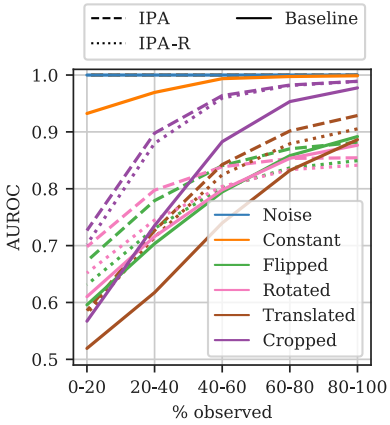
Crucially, averaging information gains computed with $\mathbf{x}^* = \mathbf{x}^{(1)}$ to $\mathbf{x}^{(N)}$ gives the estimate of the EIG presented in Eq. (10).

### H.4 BOED experimental details

The AVP-CNN, $g$, is trained to map from masked images, $\mathbf{y}$, to distributions over the class labels. Each image in the Chest X-ray 14 dataset has labels indicating the presence or absence of each of 14 pathologies. We train $g$ to produce 15 outputs: an estimated probability of the presence or absence of each of the 14 conditions individually; and an additional estimate of the probability that any (one or more) of these conditions is present. We train $g$ to estimate these using a cross-entropy loss. Masked images are sampled using almost the same mask distribution as for training the image completion networks (described in Section 4); the only difference is that patches now have 25% rather than 35% of the image width, to match the observations we use in the experiments with BOED. We use an AVP-CNN pretrained on ImageNet and then trained on Chest X-ray 14 for 32 000 iterations with a batch size of 32 and learning rate $1 \times 10^{-5}$.

To use IPA for the Chest X-ray dataset, we first trained an unconditional VAE. While IPA allows us to train all components from scratch, training this unconditional model allowed us to speed up later experimentation with IPA. The unconditional VAE was trained for 200 000 iterations (about 5 days) and the checkpoint with the best validation ELBO was used. The hyperparameters used were

Figure 8: Visualisations of BOED processes. On the top, the task is to detect 'Edema'. In the middle, we aim to detect 'Effusion' in a different patient. The task at the bottom is to detect 'Infiltration'. Four scan locations are selected for each. The left column shows the observations made before selecting each scan location. The next five columns show five of the $N = 10$ sampled image completions conditioned on these observations. Each is overlaid with the information gain predicted after scanning any location. The second column from the left shows the expected information gain at each location, given by averaging the information gains arising from each sampled image completion. A red cross marks the maximum. The final column shows the updated observations after scanning the location which maximises the expected information gain.




| ID | CIFAR-10 | | | | |
|---|---|---|---|---|---|
| OOD | Noise | Const. | Flip | Rotate | SVHN |
| IPA | **.995** | **.949** | .552 | .594 | **.753** |
| IPA-R | .987 | .921 | **.635** | **.619** | .686 |
| Baseline | .993 | .909 | .543 | .536 | .556 |

| ID | FFHQ-256 | | | | |
|---|---|---|---|---|---|
| OOD | Noise | Const. | Flip | Rotate | Crop |
| IPA | **1.00** | **1.00** | **.809** | **.808** | **.912** |
| IPA-R | **1.00** | **1.00** | .770 | .775 | .902 |
| Baseline | **1.00** | 0.98 | .769 | .771 | .823 |

Figure 9: OOD-detection on incomplete images. On the left, we show AUROC scores on FFHQ-256 computed separately for masks from varying distributions. On the right we report, for both CIFAR-10 and FFHQ-256, the average of these scores over all mask distributions.

the same as those of Child (2020) on FFHQ-256, but with a larger skip threshold of 15 000[7] and a smaller batch size (of 8 rather than 32). We trained IPA for 180 000 iterations (about 60 hours on 4 GPUs) using this unconditional model with a batch size of 8 and the original skip threshold of 100 used by Child (2020).

Selecting each scan location was done by evaluating Eq. (10) at every point in an evenly-spaced $17 \times 17$ grid over the image, and choosing the maximum. We evaluate the EIG with $N = 10$ sampled image completions. This is repeated to select the scan location for each $t = 1, \ldots, 5$ (with the sampled images conditioned on observations up to $t - 1$ at each stage). Where applicable, we use the same trained networks and hyperparameters for the baselines. For the 'non-greedy uncond' baseline, we select a scan location sequence by choosing the best of 289 sampled sequences. This number was chosen to match the number of locations in the $17 \times 17$ grid that the other methods search over at each time step. The results reported in Figs. 5 and 6 were computed on a randomly-sampled 5000 image subset of the Chest X-ray 14 test set.

## I Out-of-distribution detection

A major advantage of likelihood-based models is the possibility to use them to detect when an input is dissimilar to the training data (out-of-distribution, or OOD). Since a learned model is likely to perform poorly on such inputs, it is important for deployed systems to be able to recognise them and this has been the focus of a substantial body of work (Ren et al., 2019; Hendrycks & Gimpel, 2016; Xiao et al., 2020; Havtorn et al., 2021). The task we consider is detecting whether a partially observed image (e.g. the input to an image completion system) is OOD.

OOD-detection using probabilistic models is theoretically appealing and robust, since no assumptions need to be made about the OOD data (Xiao et al., 2020; Havtorn et al., 2021). Such OOD-detection metrics commonly have the following interface: they take as input a probabilistic model (e.g. a VAE) and a data point, and output a scalar value. The greater the value of this scalar, the more likely it is that the data point is within the distribution of the generative model. One intuitive metric is simply the log-likelihood of the data point under the model. This can work poorly in practice, however, and it has been found that learned models sometimes assign higher likelihood to out-of-distribution data than to in-distribution data (Nalisnick et al., 2018). Several alternative OOD-detection metrics have been proposed specifically for VAEs (Xiao et al., 2020; Havtorn et al., 2021). To use them, we note that we can use networks trained with either IPA or IPA-R to lower-bound the likelihood of a partially-observed image. Specifically, this lower-bound is the reverse KL objective in Eq. (42). After conducting preliminary experiments using the VD-VAE with several

[7] Child (2020) improve training stability by skipping updates with gradients larger than a set threshold. Using a larger threshold, and so skipping fewer updates, was found to be necessary to train their architecture on the Chest X-ray dataset.

OOD-detection metrics (Xiao et al., 2020; Havtorn et al., 2021), we found that improved performance could be obtained (for OOD-detection of both partially- and fully-observed images) using the "temperature gradient" metric described further in Appendix I.1 below. It is efficient to compute, requiring the VAE to be run just once and gradients computed.

As a baseline, we complete the partially observed image with our best performing baseline image completion method, CoModGAN, and perform OOD-detection on the result (using the "temperature gradient" metric on an unconditional VD-VAE). Figure 9 shows our results on CIFAR-10 and FFHQ-256. We test various types of OOD data: **Noise** samples each pixel value independently from a uniform distribution; **Constant** uniformly samples a single colour for the entire image; **Flip** vertically flips an in-distribution (ID) image; **Rotate** rotates an ID image by 90°; **Crop** upsamples an ID image by a factor of 1.5 and then crops to the original size. **SVHN** is the Street View House Number dataset (Netzer et al., 2011). With 80-100% of the image observed, we obtain better AUROCs than similar methods applied to the full image. In particular, on CIFAR-10 vs SVHN, we outperform both Xiao et al. (2020) and Havtorn et al. (2021), albeit with a larger VAE architecture. Using IPA for OOD-detection results in performance that degrades gracefully as less of the image is observed. There are therefore particularly large benefits over the baseline when only 0-20% or 20-40% of the image is observed.

Figure 10 shows a breakdown of the AUROC scores for each mask distribution on CIFAR-10, similar to that shown for FFHQ-256 in Fig. 9. Error bars are standard deviations computed with 3 runs, with the unconditional VD-VAE and IPA/IPA-R retrained each time.

### I.1 Out-of-distribution detection with temperature gradients

We conducted preliminary experiments with several techniques for OOD-detection with VAEs, applying each to the VD-VAE architecture. Using the likelihood-regret technique (Xiao et al., 2020) with the VD-VAE is computationally costly (as it involves optimizing neural network parameters), and we obtained results on CIFAR-10 considerably worse than those produced by the smaller VAE architecture with which it was introduced. We also implemented the log-likelihood ratio metric (Havtorn et al., 2021) for the VD-VAE, but did not find any hyperparameter configurations with which it was consistently better than random guessing across different tasks.

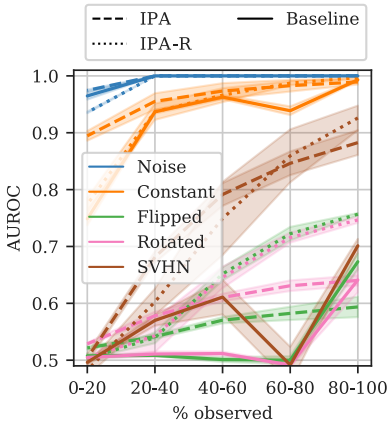



Figure 10: Breakdown of AUROC for each mask distribution on CIFAR-10.

We therefore introduce a metric which we found to work well with the VD-VAE. It is based on a common heuristic to make a VAE produce more regular samples: reducing the variance of the prior over its latent variables (Vahdat & Kautz, 2020). If $\mathbf{t}$ is a vector containing the temperature $\mathbf{t}_l$ for each layer $l$, then doing so corresponds to modifying the prior $p_{\text{model}}(z)$ so that the Gaussian distribution over each group of latent variables is scaled by the corresponding temperature:

$$p_{\text{model}}(z;\mathbf{t}) = \prod_{l=1}^{L} \frac{1}{C(\mathbf{t}_l, z_{<l})} p_{\text{model}}(z_l|z_{<l})^{\frac{1}{\mathbf{t}_l^2}} \tag{70}$$

where $C(\mathbf{t}_l, z_{<l})$ is a normalisation constant. In practice, since $p_{\text{model}}(z_l|z_{<l})$ is a Gaussian distribution, this scaling corresponds to simply multiplying the standard deviation by $\mathbf{t}_l$.

We can lower-bound the corresponding marginal likelihood $p_{\text{model}}(\mathbf{y};\mathbf{t})$ using the partial encoder similarly to the training objective in Eq. (42):

$$\mathbb{E}_{p_{\text{data}}(\mathbf{y})}\left[\log p_{\text{model}}(\mathbf{y};\mathbf{t})\right] \geq \mathbb{E}_{p_{\text{data}}(\mathbf{y})}\mathbb{E}_{\hat{q}(z|\mathbf{y})}\left[\log \frac{p_{\text{model}}(z;\mathbf{t})p_{\text{model}}(\mathbf{y}|z)}{\hat{q}(z|\mathbf{y})}\right]. \tag{71}$$

We hypothesise that OOD examples will usually either be excessively regular, and therefore have a higher marginal likelihood for $t < 1$, or excessively irregular, and therefore have a higher likelihood for $t > 1$. In-distribution (ID) examples, on the other hand, should have the highest marginal

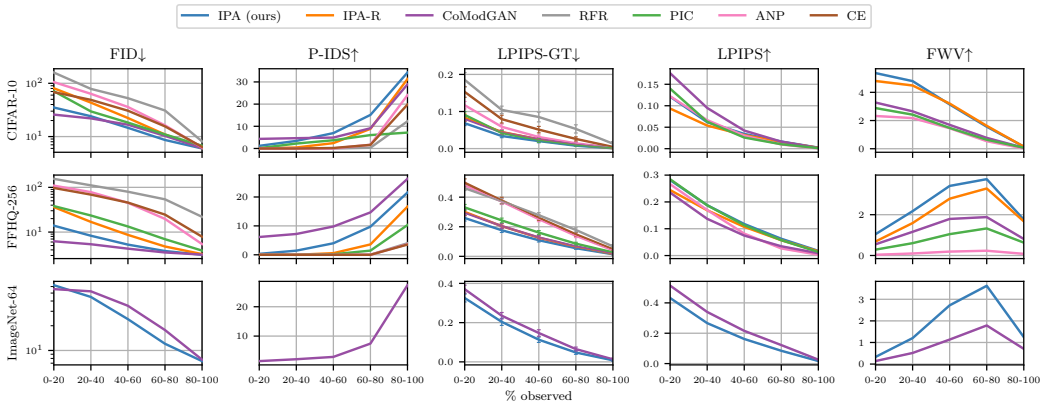



Figure 11: Expanded Fig. 3 including results on ImageNet-64 and two additional metrics for diversity. Preliminary results for CoModGAN on ImageNet-64 are computed after 9 million training iterations out of a planned 25 million. We will update this figure when more results are available.

likelihood for $t \approx 1$. A useful feature to distinguish between ID and OOD examples is therefore likely to be the gradient $\frac{\partial}{\partial \mathbf{t}} p(\mathbf{y}; \mathbf{t})|_{\mathbf{t}=\mathbf{1}}$, which we can approximate with the gradient w.r.t. $\mathbf{t}$ of the lower-bound above. We fit a multivariate Gaussian to 5000 samples of $\frac{\partial}{\partial \mathbf{t}} p(\mathbf{y}^{(i)}; \mathbf{t})|_{\mathbf{t}=\mathbf{1}}$ with $\mathbf{y}^{(i)}$ sampled from the training data with the training mask distribution. To obtain a score for how in-distribution a new example $\mathbf{y}$ is, we compute the probability density of $\frac{\partial}{\partial \mathbf{t}} p(\mathbf{y}; \mathbf{t})|_{\mathbf{t}=\mathbf{1}}$ under this multivariate Gaussian. The greater this score, the more likely it is that an example is in-distribution. We call this the "temperature-gradient" metric.

### I.2 OOD DETECTION DETAILS

We compute the AUROC scores presented in Fig. 9 using the 5000 image test set for CIFAR-10 and its OOD-transformations; a 5000 images subset of the SVHN test set; and the 7000 image test set for FFHQ-256. We use 5000 samples of $32 \times 32$ 'Noise' and 'Constant' images when comparing against CIFAR-10, and 200 samples at $256 \times 256$ resolution when comparing against FFHQ-256.

We also present results for the 'Translated' transformation in Fig. 9 for FFHQ-256. This transformation takes a crop with 90% of the image size from one of the four corners and then rescales it to the original size. The result is thus slightly translated so that e.g. the eyes of an FFHQ image are no longer aligned. It is not shown in our results table due to space constraints.

## J ADDITIONAL RESULTS

Figure 11 shows some additional results, including the LPIPS diversity score and FWV described in Appendix G.3, and results for ImageNet-64. We will update this upon computing the remaining metrics. Although we show results for CoModGAN on ImageNet-64, these are only preliminary as they are computed after 9 million training iterations out of a planned 25 million.

## K LACK OF SEMANTIC DIVERSITY IN COMODGAN

In this section we demonstrate the some failure cases of CoModGAN in which it fails to produce semantically diverse images. We suspect that this behaviour of CoModGAN contributes to its poor performance on the LPIPS-GT metric. Figure 12 shows a few examples of partial images for which CoModGAN struggles to generate semantically diverse images. Compare these results with IPA's results reported in Fig. 13, showing a wide range of completions for all the partial images.

## L IMAGE SAMPLES

**FFHQ-256** Figures 14 to 18 (on pages 32 to 36) show sampled completions from each method (IPA, IPA-R, and all baselines) for 5 FFHQ-256 images. For each image, completions based on four different masks are shown. For each mask, we plot five sampled completions from each stochastic method and the single completion from each deterministic method.

**CIFAR-10** Figures 19 to 23 (on pages 37 to 41) show sampled completions for CIFAR-10 images, formatted in the same way as Figures 14 to 18.

**ImageNet-64** We show samples from conditional VAEs trained on ImageNet-64 in Fig. 24.

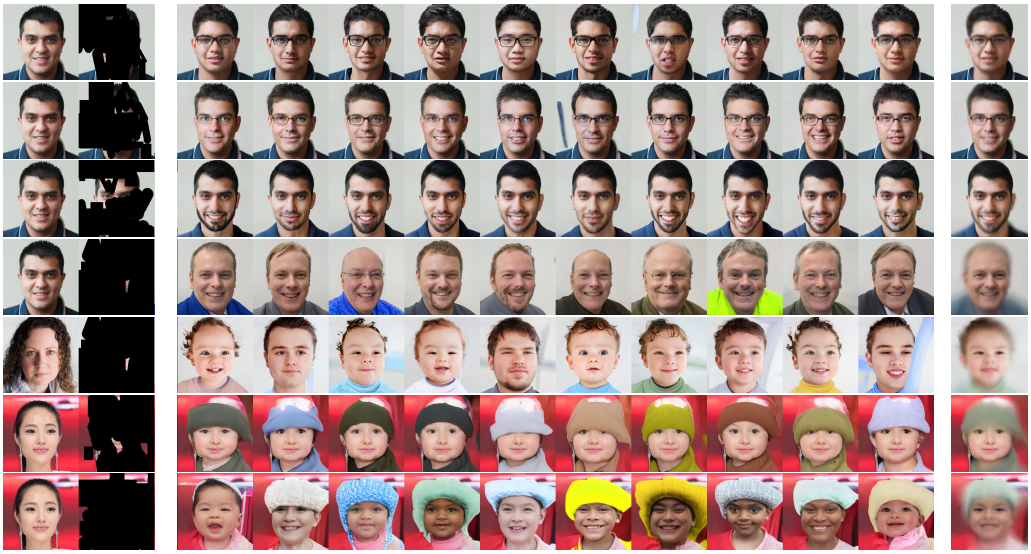
**Edges-to-photos** We show samples from conditional VAEs trained on the Edges2Handbags and Edges2Shoes datasets in Fig. 25.

**Chest X-ray 14** Figure 26 compares sampled completions of x-ray images from IPA and CoModGAN. Samples from IPA can be seen to be more reliably diverse, helping to explain the improved performance of IPA compared to CoModGAN.

**Discussion** It is apparent from the image samples, particularly when the number of observed pixels is small, that IPA-R has less sample diversity than IPA, which can be explained by the mode-seeking behaviour of the reverse-KL and subsequent under-estimation of posterior uncertainty (Minka et al., 2005). Finally, we note a flaw with some of the completions produced by IPA: a lack of bilateral symmetry in FFHQ-256 samples. Again, this is most apparent when few pixels are observed. This issue can be seen in unconditional samples from the underlying VAEs as well (see the samples reported by Child (2020)). Therefore, it is likely that any future advances in image modeling with VAEs could be integrated to improve this aspect of the results.

## M CODE

We have attached code for training IPA and IPA-R. We will later release code for using IPA with Bayesian experimental design and out-of-distribution detection. This will all be made publicly available, along with weights for networks trained with IPA/IPA-R and the weights for our baseline models.



(a) $\mathbf{x}, \mathbf{y}$ (b) Sampled completions (c) Mean

Figure 12: Examples of completions lacking semantic diversity in CoModGAN. Panel (a) shows the true image and the partial image on which the completions are conditioned. Panel (b) shows 10 completions sampled randomly from the CoModGAN model. Panel (c) shows the mean image computed from 100 sampled completions. In each row, almost all the completions have an attribute in common which does not exist in the true image. This common attribute is glasses in the first two rows and a beard in the third row. Most faces in the fourth row look older than the true image while in the remaining rows, completions are mostly babies. Finally, completions in the last two rows are mostly wearing a hat. These attributes are visible in the mean images as well.

(a) $\mathbf{x}, \mathbf{y}$ (b) Sampled completions (c) Mean

Figure 13: Examples completions from our IPA model, to compare with Fig. 12. Panel (a) shows the true image and the partial image on which the completions are conditioned. Panel (b) shows 10 completions sampled from the IPA model. Panel (c) shows the mean completion computed from 100 randomly sampled completions. Samples from IPA cover a wide variety of attributes.

Figure 14: Sampled completions from each method for an FFHQ-256 image. Panels (a), (d) and (h) all show the true image and the masked image on which the samples in each row are conditioned. Panels (b) and (c) show completions from IPA and IPA-R. (g) and (k) show the single deterministic completion produced by CE and RFR respectively. Each of the remaining panels show five completions for the stochastic baselines.

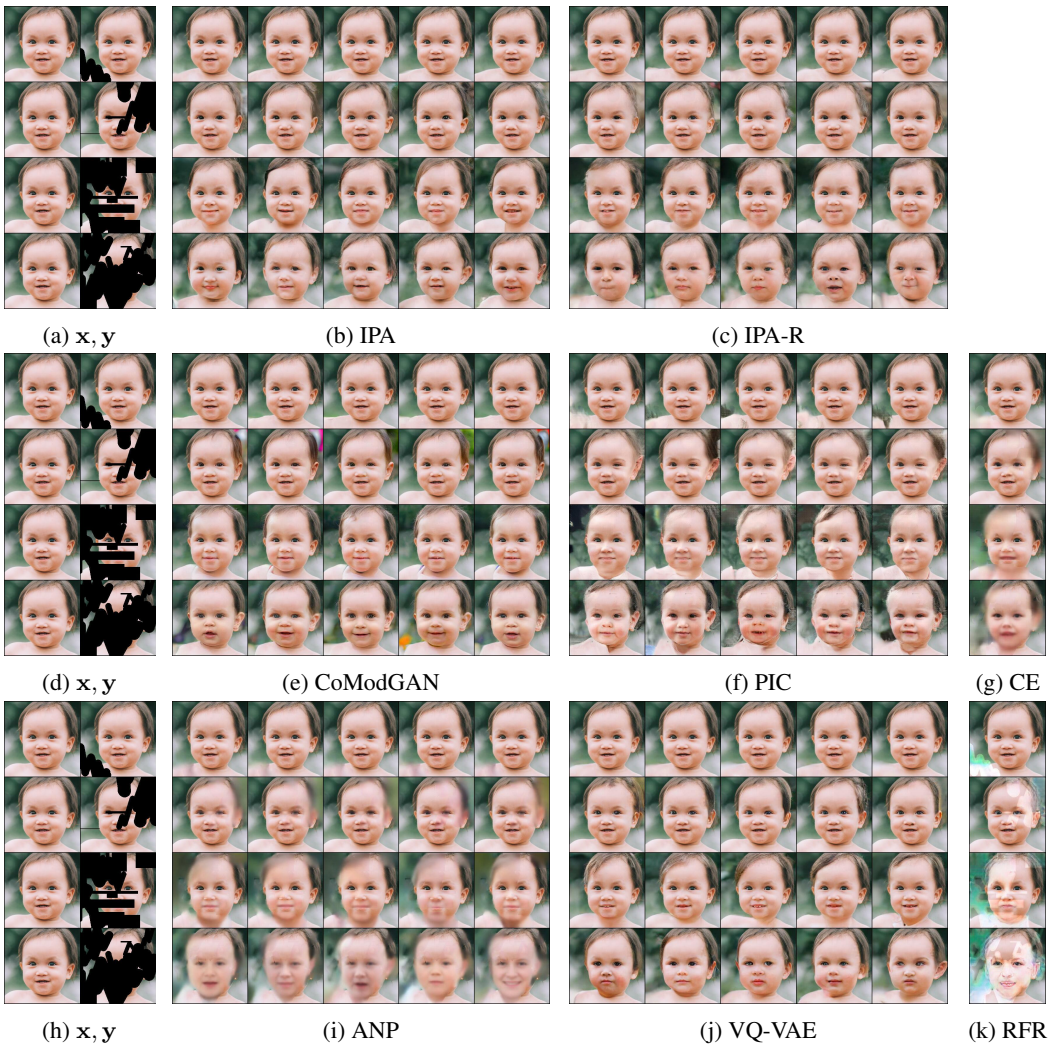



Figure 15: Sampled completions from each method for an FFHQ-256 image. Panels (a), (d) and (h) all show the true image and the masked image on which the samples in each row are conditioned. Panels (b) and (c) show completions from IPA and IPA-R. (g) and (k) show the single deterministic completion produced by CE and RFR respectively. Each of the remaining panels show five completions for the stochastic baselines.

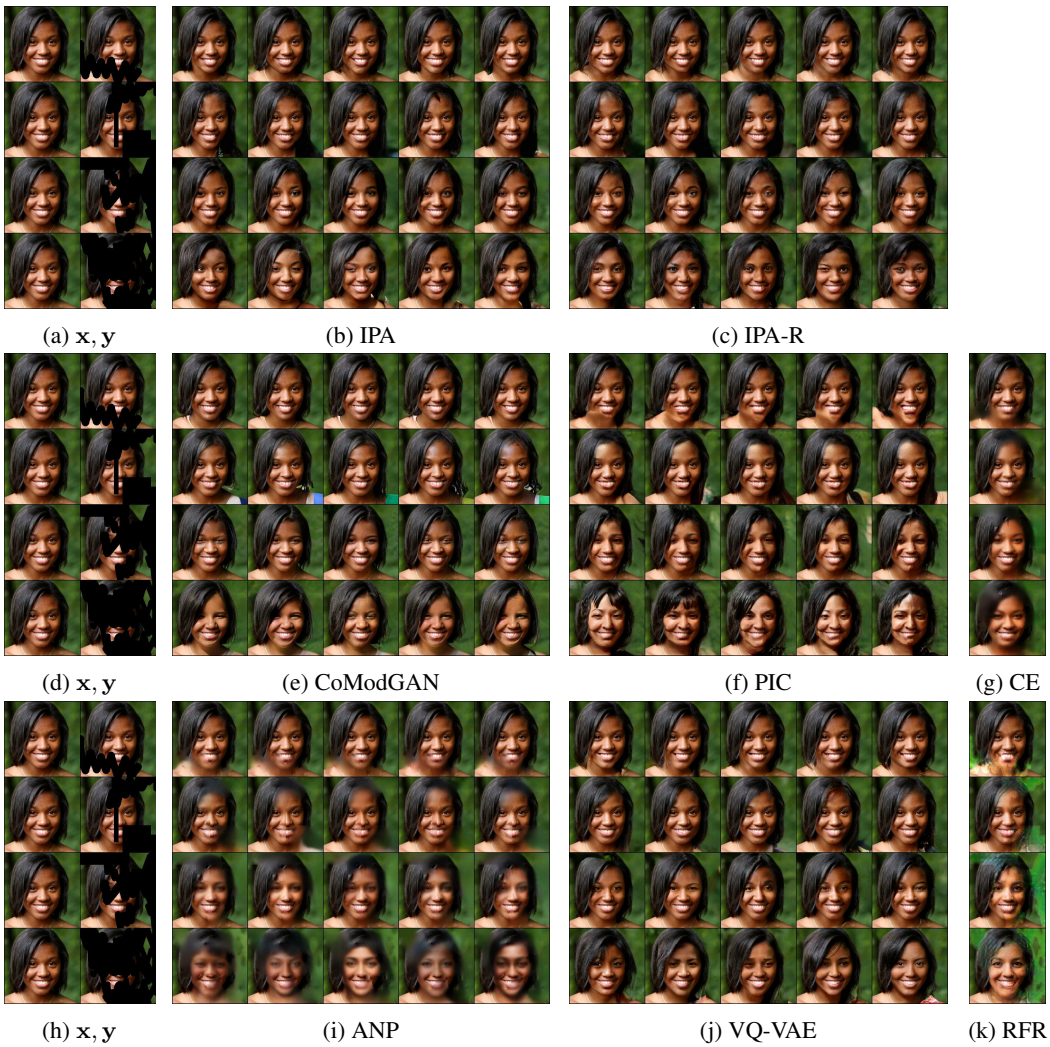
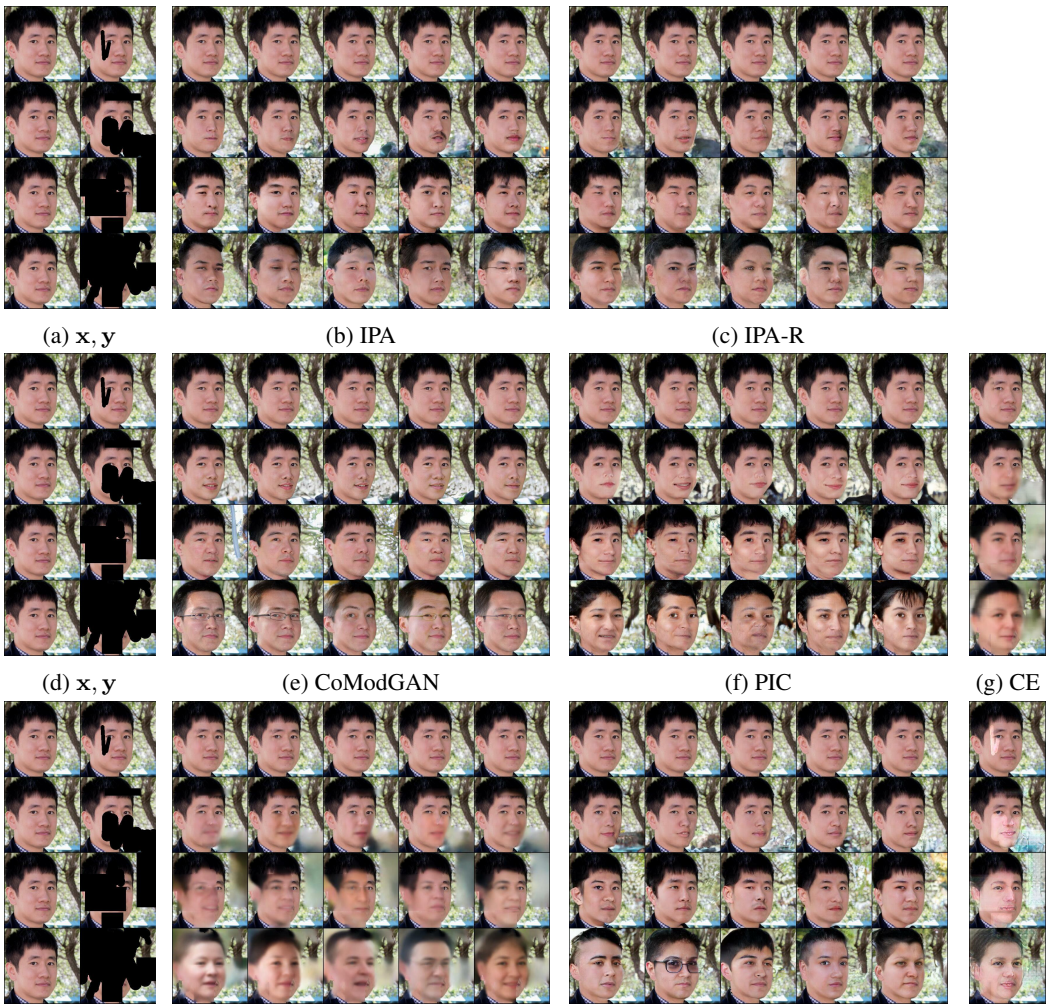



Figure 16: Sampled completions from each method for an FFHQ-256 image. Panels (a), (d) and (h) all show the true image and the masked image on which the samples in each row are conditioned. Panels (b) and (c) show completions from IPA and IPA-R. (g) and (k) show the single deterministic completion produced by CE and RFR respectively. Each of the remaining panels show five completions for the stochastic baselines.

Figure 17: Sampled completions from each method for an FFHQ-256 image. Panels (a), (d) and (h) all show the true image and the masked image on which the samples in each row are conditioned. Panels (b) and (c) show completions from IPA and IPA-R. (g) and (k) show the single deterministic completion produced by CE and RFR respectively. Each of the remaining panels show five completions for the stochastic baselines.



(a) $\mathbf{x}, \mathbf{y}$ (b) IPA (c) IPA-R

(d) $\mathbf{x}, \mathbf{y}$ (e) CoModGAN (f) PIC (g) CE

(h) $\mathbf{x}, \mathbf{y}$ (i) ANP (j) VQ-VAE (k) RFR

Figure 18: Sampled completions from each method for an FFHQ-256 image. Panels (a), (d) and (h) all show the true image and the masked image on which the samples in each row are conditioned. Panels (b) and (c) show completions from IPA and IPA-R. (g) and (k) show the single deterministic completion produced by CE and RFR respectively. Each of the remaining panels show five completions for the stochastic baselines.

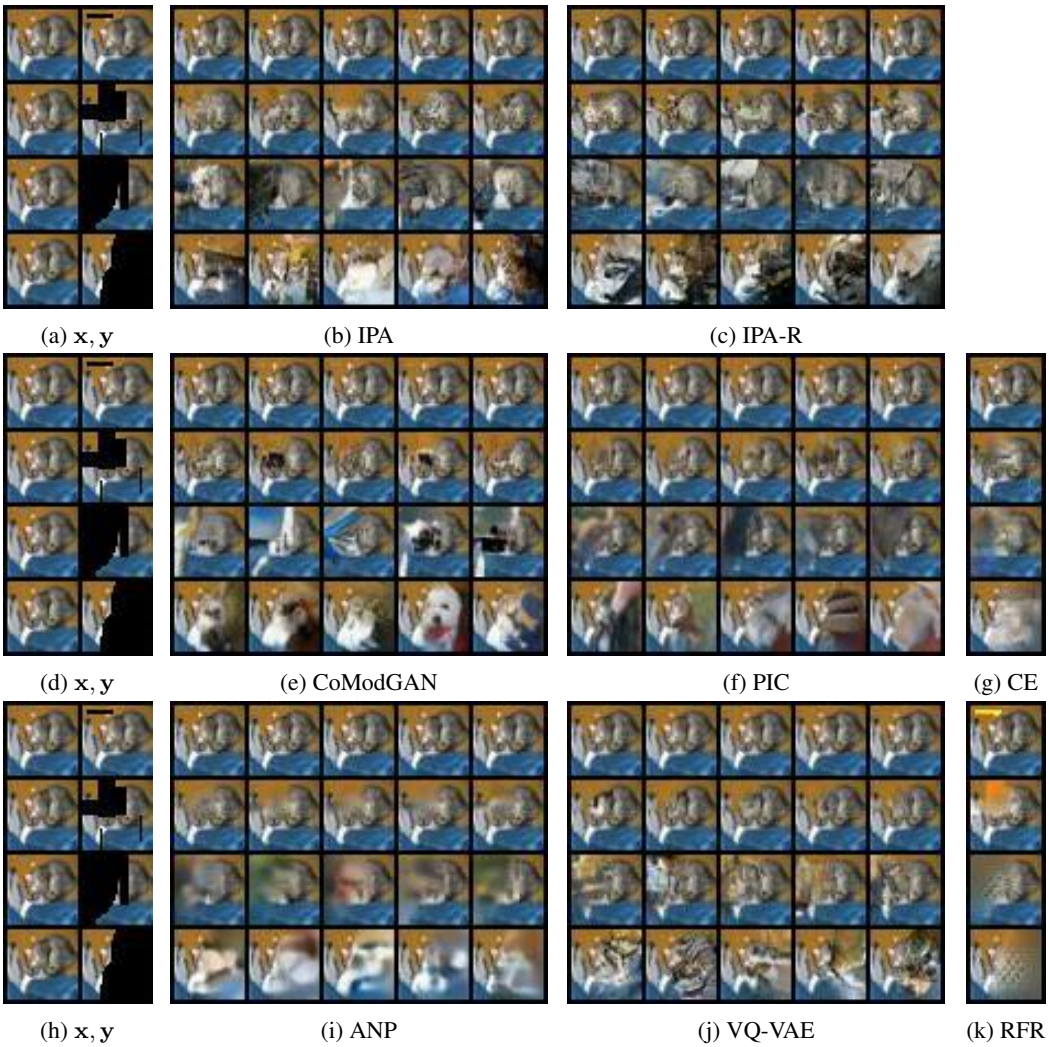



Figure 19: Sampled completions from each method for a CIFAR-10 image. Panels (a), (d) and (h) all show the true image and the masked image on which the samples in each row are conditioned. Panels (b) and (c) show completions from IPA and IPA-R. (g) and (k) show the single deterministic completion produced by CE and RFR respectively. Each of the remaining panels show five completions for the stochastic baselines.

Figure 20: Sampled completions from each method for a CIFAR-10 image. Panels (a), (d) and (h) all show the true image and the masked image on which the samples in each row are conditioned. Panels (b) and (c) show completions from IPA and IPA-R. (g) and (k) show the single deterministic completion produced by CE and RFR respectively. Each of the remaining panels show five completions for the stochastic baselines.

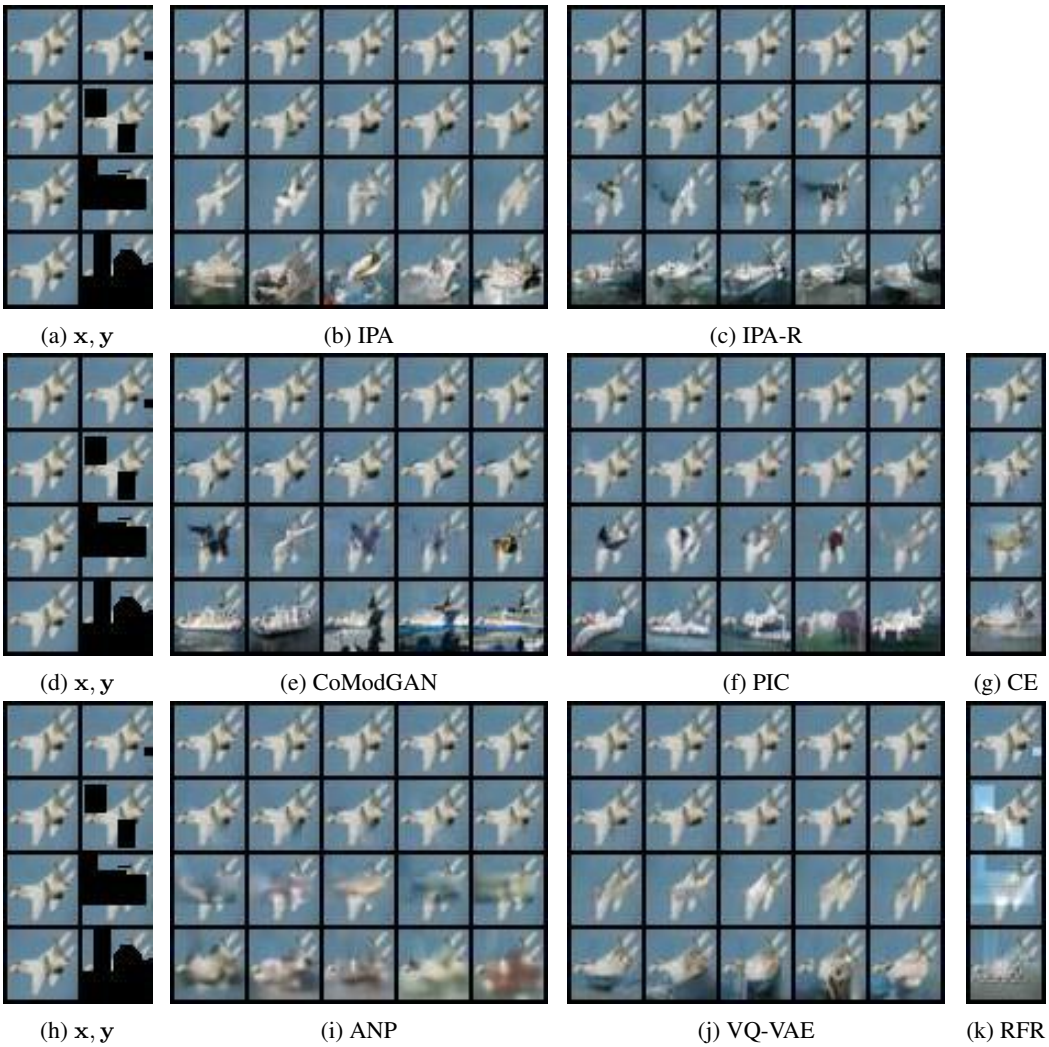



Figure 21: Sampled completions from each method for a CIFAR-10 image. Panels (a), (d) and (h) all show the true image and the masked image on which the samples in each row are conditioned. Panels (b) and (c) show completions from IPA and IPA-R. (g) and (k) show the single deterministic completion produced by CE and RFR respectively. Each of the remaining panels show five completions for the stochastic baselines.

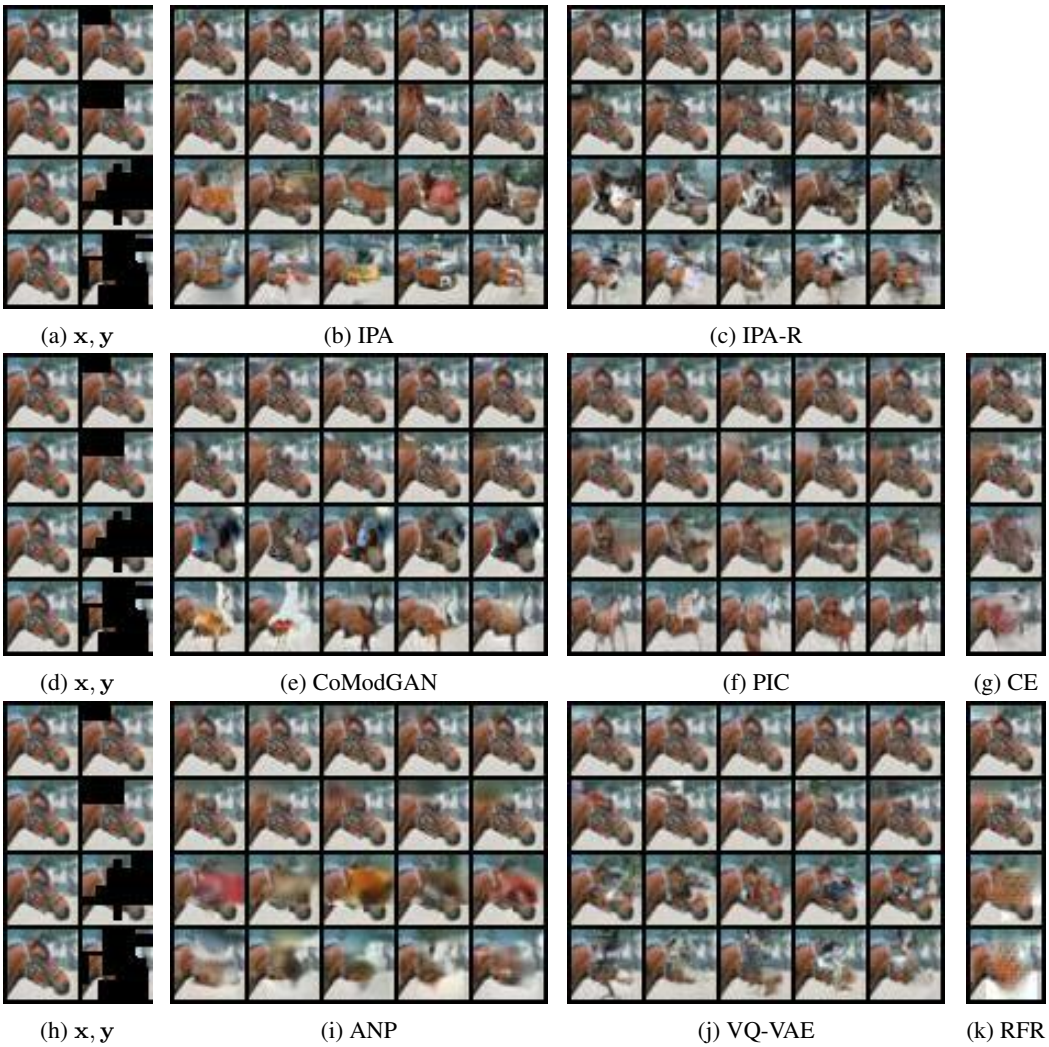



Figure 22: Sampled completions from each method for a CIFAR-10 image. Panels (a), (d) and (h) all show the true image and the masked image on which the samples in each row are conditioned. Panels (b) and (c) show completions from IPA and IPA-R. (g) and (k) show the single deterministic completion produced by CE and RFR respectively. Each of the remaining panels show five completions for the stochastic baselines.

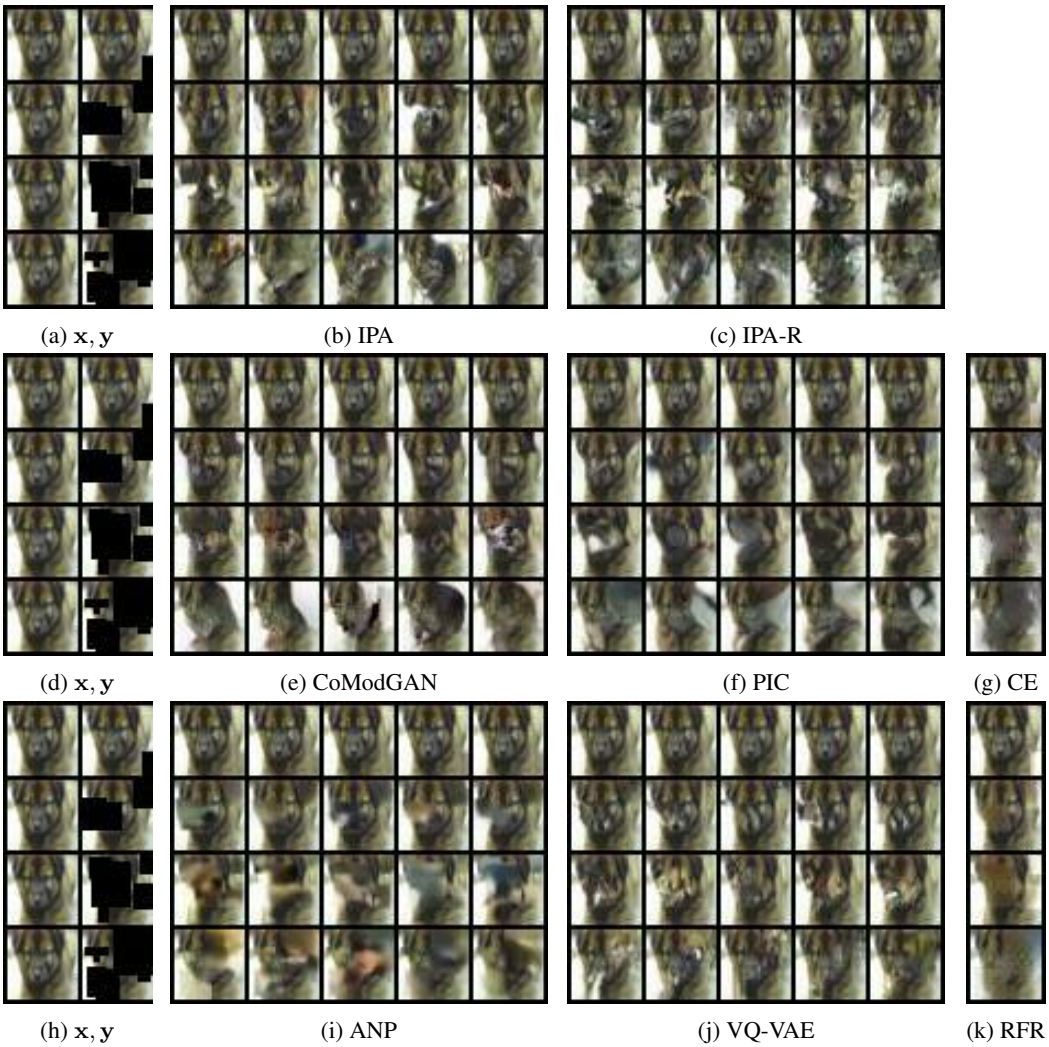
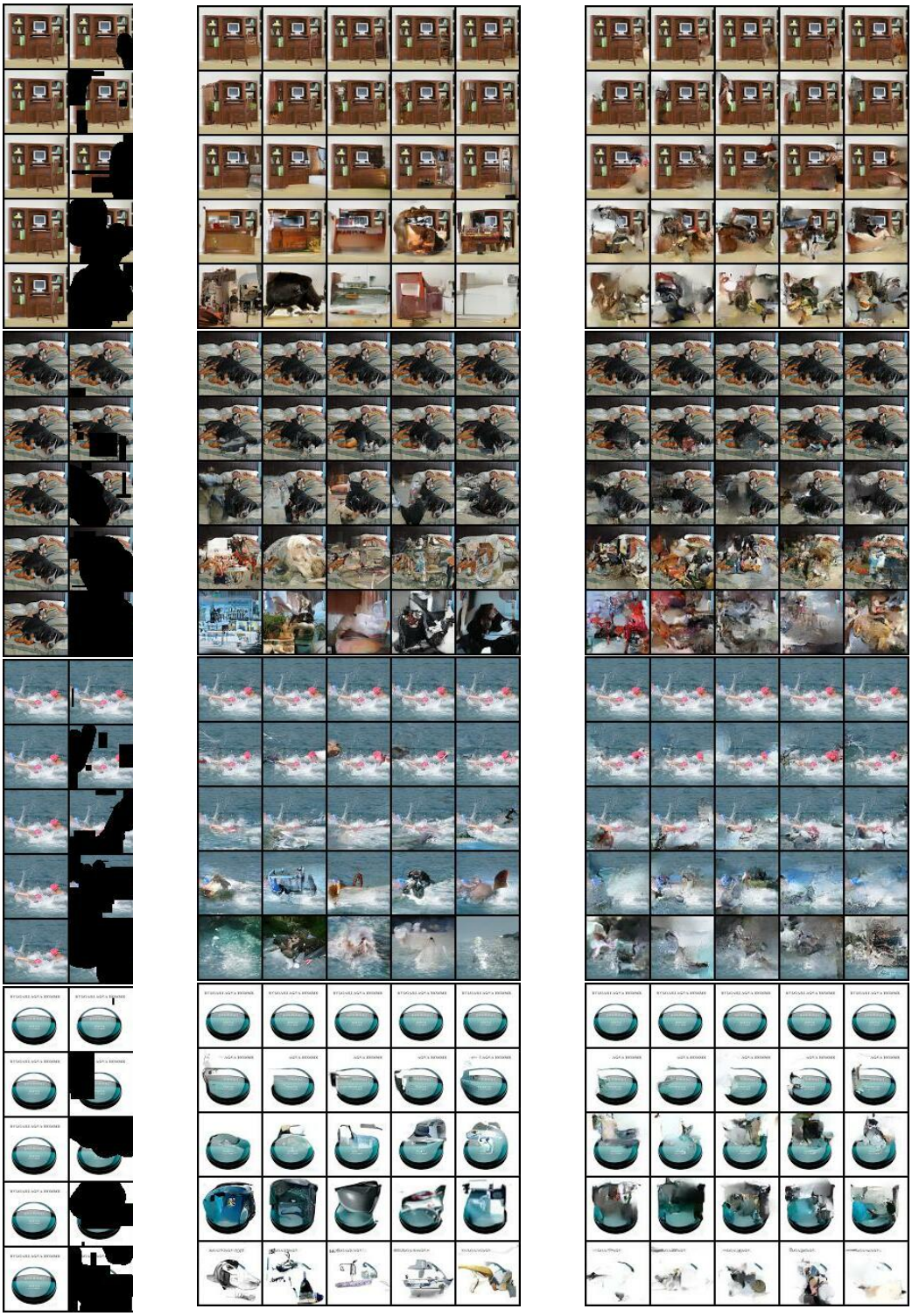



Figure 23: Sampled completions from each method for a CIFAR-10 image. Panels (a), (d) and (h) all show the true image and the masked image on which the samples in each row are conditioned. Panels (b) and (c) show completions from IPA and IPA-R. (g) and (k) show the single deterministic completion produced by CE and RFR respectively. Each of the remaining panels show five completions for the stochastic baselines.

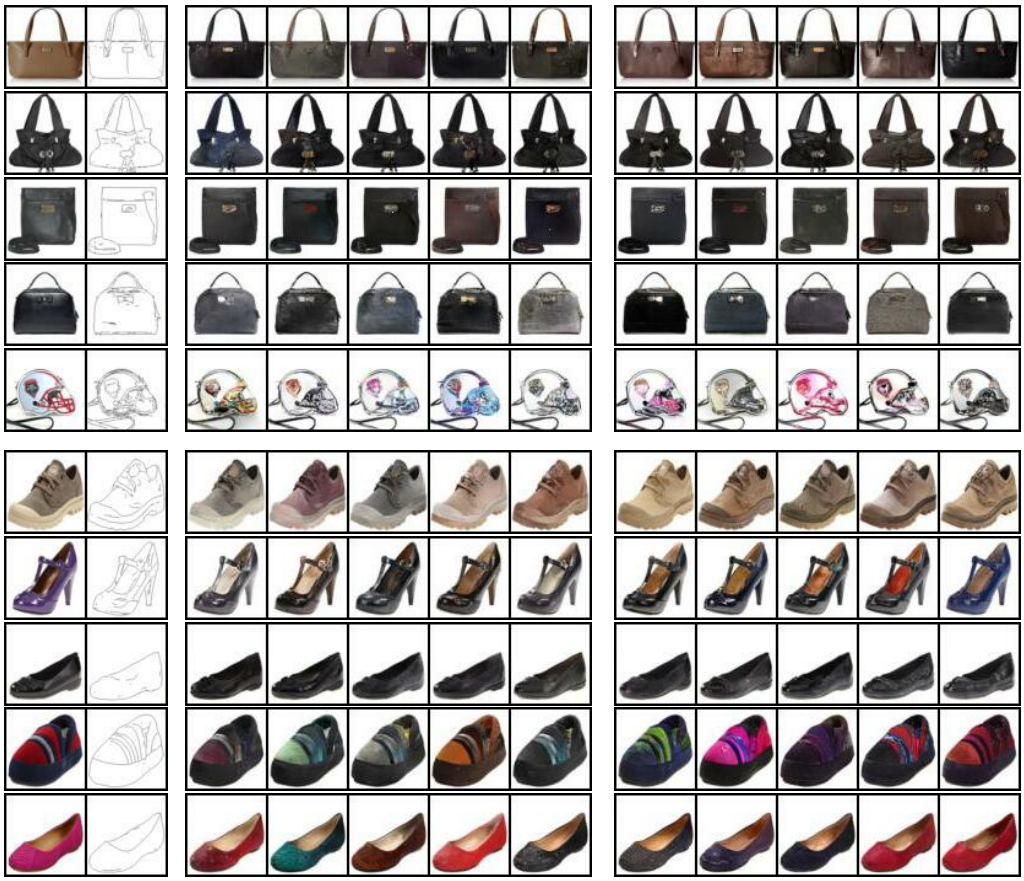


(a) $\mathbf{x}, \mathbf{y}$

(b) IPA (after 102 hours).

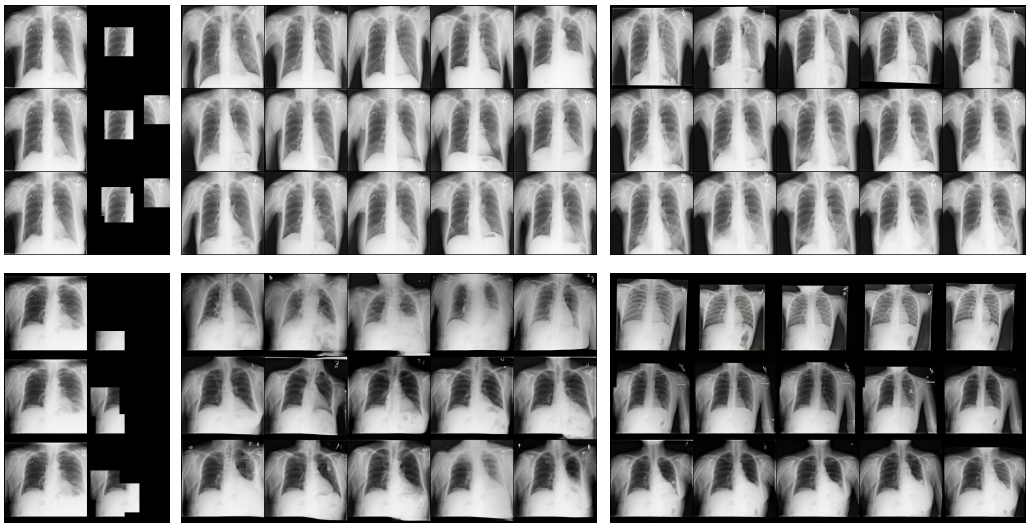
(c) Trained from scratch (after 102 hours).

Figure 24: Sampled image completions from IPA, compared with completions by a network trained from scratch. Panel (a) shows the ground truth and masked image. Panel (b) shows completions by IPA, and panel (c) shows completions by the same conditional VAE architecture trained from scratch.



(a) $\mathbf{x}, \mathbf{y}$ (b) Trained from scratch (c) IPA pretrained on ImageNet64

Figure 25: Conditional generation for the Edges2Handbags (top 5 rows) and Edges2Shoes (bottom 5 rows) datasets. The left two columns show the ground truth image $\mathbf{x}$ and the extracted edges, $\mathbf{y}$. The next 5 columns show samples using networks trained from scratch for 1 week. The rightmost 5 columns show samples using IPA with a pretrained ImageNet64 artifact and about 2 days of training. There is little visible difference in quality, although the FID scores in Fig. 4 suggest that samples from IPA are slightly better.



(a) $\mathbf{x}, \mathbf{y}$ (b) IPA (c) CoModGAN

Figure 26: Panels of sampled image completions from IPA and CoModGAN. We show completions given three different masks for each of two ground truth images. For the first ground truth image (top three rows), there is little obvious difference between the completions produced by IPA and by CoModGAN. However, for each of the masks applied to the second ground truth image (bottom three rows), CoModGAN produces a posterior with minimal diversity, and which does not appear to include the ground truth. We inspected completion panels for CoModGAN on 20 different ground truth images, and found that this type of mode collapse occurred in 5 of them. IPA, in contrast, produces reliably diverse image completions for any $\mathbf{y}$ or ground truth image inspected.