Autoencoding Conditional Neural Processes for Representation Learning

Victor Prokhorov [1] **Ivan Titov** [1 2] **N. Siddharth** [1 3]

Abstract

Conditional neural processes (CNPs) are a flexible and efficient family of models that *learn to learn* a stochastic process from data. They have seen particular application in contextual image completion—observing pixel values at some locations to predict a distribution over values at other unobserved locations. However, the choice of pixels in learning CNPs is typically either random or derived from a simple statistical measure (e.g. pixel variance). Here, we turn the problem on its head and ask: which pixels would a CNP like to observe—do they facilitate fitting better CNPs, and do such pixels tell us something meaningful about the underlying image? To this end we develop the Partial Pixel Space Variational Autoencoder (PPS-VAE), an amortised variational framework that casts CNP context as latent variables learnt simultaneously with the CNP. We evaluate PPS-VAE over a number of tasks across different visual data, and find that not only can it facilitate better-fit CNPs, but also that the spatial arrangement and values meaningfully characterise image information—evaluated through the lens of classification on both within and out-of-data distributions. Our model additionally allows for *dynamic* adaption of context-set size and the ability to scale-up to larger images, providing a promising avenue to explore learning meaningful and effective visual representations.

1. Introduction

Conditional neural processes (Garnelo et al., 2018a, CNPs) are a family of models that learn distribution over functions. In contrast to conventional approaches such as Gaussian processes, which are effective but become computationally expensive once the data size increases, CNPs are both flexible regarding the functions they approximate, thanks to being neural networks, and scalable to large datasets. In the visual domain, they have been used for contextual image completion. Given a context set, a set of ordered pairs—observed pixel values and their image coordinates—CNPs learn to impute the other, unobserved, pixels.

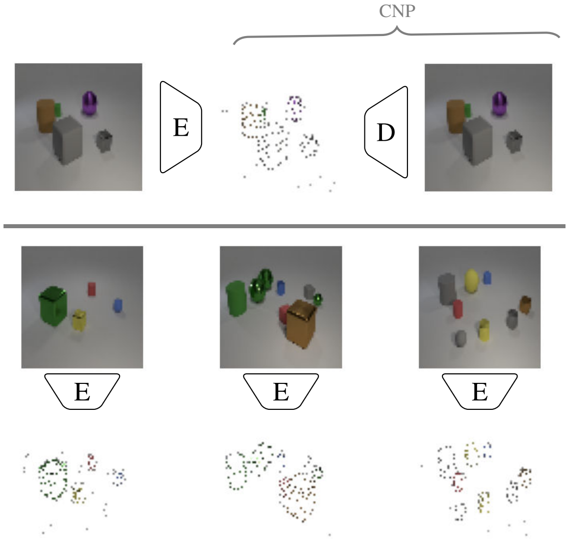


Figure 1: (top) The PPS-VAE framework. (bottom) Examples of meaningful context points induced by the encoder.

While prior work on CNPs primarily focusses on model choices such as inductive biases that allow capturing various properties of the context set better (Gordon et al., 2020) or dependencies between the unobserved pixel values (Garnelo et al., 2018b), we explore a key *dual* question—regarding the context set itself. Where the context set is typically chosen at random, or derived from some simple statistic (e.g. pixel variance) to train the CNP, we ask: *which pixels would a CNP like to observe?* Do such pixels allow better-fitting of CNPs, and do they tell us something meaningful about the underlying image? We explore these questions from the frame of representation learning, where the context can be viewed as *latent* representations of the image—one that happens to exist in the data space.

From a purely representation-learning perspective, one can relate the question above with that of learning (a) a discrete feature selector as in Concrete Autoencoder (Balın et al., 2019, CAE) and (b) a discrete latent 'code', as first established in (van den Oord et al., 2017, VQ-VAE), and subsequently popularised by approaches like DALL-E (Ramesh et al., 2021). Where the CAE employs a global feature selector, we approximate a posterior distribution and where

[1] School of Informatics, University of Edinburgh [2] ILLC, University of Amsterdam [3] The Alan Turing Institute. Correspondence to: Victor Prokhorov <email: victorprokhorov91@gmail.com, code: https://github.com/exlab-research/pps-vae>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

the VQ-VAE learns an arbitrary code, we learn one that directly corresponds the pixels in the image and is sufficiently expressive to capture image content—measured through reconstruction. Given the interpretation of our model as *imputing* the remainder of the observation from the given pixel 'codebook', we bring together the ideas of discrete representation learning and learning-to-learn stochastic processes (CNPs) into a single framework—the partial pixel specification variational autoencoder (PPS-VAE, shown in Figure 1).

Specifically in this work, we

- develop an amortised variational inference framework (PPS-VAE) to *learn to predict* context points that a CNP can faithfully complete (Section 2),
- provide evidence that learning context along with the CNP learns a better model over images (Section 3.1),
- demonstrate that the PPS-VAE encodes useful and meaningful information in the *learnt* context set—evaluated through both qualitative observation and a classification-probe task — both in-distribution and out-of-distribution settings (Section 3.3), and
- highlight the utility, flexibility, and scalability of PPS-VAE with improved performance using simple post-hoc augmentations such as dynamic resizing of context sets and reconfiguration of context sets as tiles (Section 3.3).

2. Model

CNPs. Given function $f : \mathcal{X} \to \mathcal{Y}$ mapping observations $x \in \mathcal{X}$ to targets $y \in \mathcal{Y}$, and *context set* $\mathcal{C} = \{(x_m, y_m)\}_{m=1}^{M}$, a CNP (Garnelo et al., 2018a) learns a distribution over functions $f(x; \mathcal{C})$—predicting targets conditioned on context $\mathcal{C}$. For unseen $\boldsymbol{x}_T = \{x_t\}_{t=1}^{T}$, the CNP defines the distribution over $\boldsymbol{y}_T = \{y_t\}_{t=1}^{T}$ as

Eq. 1 - CNP's Predictive Distribution

$$p_\theta(\boldsymbol{y}_T \mid \boldsymbol{x}_T, C) = \prod_{t=1}^{T} \mathcal{N}(y_t \mid \mu_t, \sigma_t)$$

$$\mu_t, \sigma_t = s_\theta(x_t, r_\theta(\mathcal{C})).$$

Crucially, it relies on transforming the entire context set $\mathcal{C}$ in a permutation-invariant fashion (Zaheer et al., 2017, DeepSet) using r_θ, to construct the parameters of the distribution through s_θ, using neural networks as parameters.

In the image domain, a CNP learns to predict the colour values $\boldsymbol{y}_T$ at unseen locations $\boldsymbol{x}_T$ given a set of observed pixel locations $\boldsymbol{x}_M$ and their corresponding values $\boldsymbol{y}_M$. By observing some small, sparse subset of the image itself, the task here is to impute the rest of the image. Note that, in this setting, knowing the set of observed locations $\boldsymbol{x}_M$ implies knowing the set of unseen locations $\boldsymbol{x}_T$, as for images of fixed size with $\boldsymbol{x}$[1] pixel locations, one is the complement ($\boldsymbol{x}_T = \boldsymbol{x}_M{}'$) of the other. Learning a CNP in this setting involves (random) sampling of different context sets and subsequent imputation of the values at unseen locations, across a dataset of images.

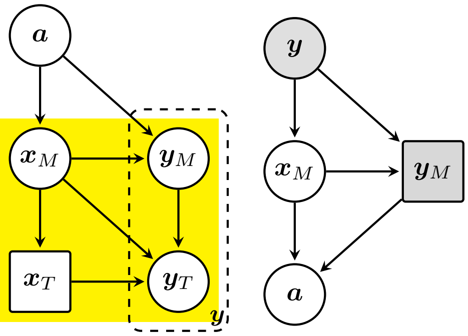


Figure 2: CNP generative model (left yellow); PPS-VAE generative (left) and inference (right) models.

PPS-VAE. To answer our question of what kinds of context the CNP would like to observe, and how meaningful this context is, we first cast the CNP as a fully generative model as shown in Figure 2 (left—yellow area),

Eq. 2 - CNP's Generative Model

$$p_\theta(\boldsymbol{x}, \boldsymbol{y}|M) = p_\theta(\boldsymbol{x}_M)\ p_\theta(\boldsymbol{y}_M|\boldsymbol{x}_M)\ p_\theta(\boldsymbol{y}_T|\boldsymbol{x}_T, \boldsymbol{x}_M, \boldsymbol{y}_M)$$

Here, M is taken to be a given fixed value, $p_\theta(\boldsymbol{x}_M)$ defines a distribution over *arrangements* of M pixel locations in an image, and $p_\theta(\boldsymbol{y}_M|\boldsymbol{x}_M)$ a distribution over values at the given locations. The model can be viewed as generating data in two stages (autoregressive): first generating the values corresponding to the context points, and subsequently, conditioning on these locations and values to impute the values elsewhere on the image. From this, to get to the full PPS-VAE generative model, we additionally introduce an *abstractive* latent variable $\boldsymbol{a}$[2] as shown in Figure 2 (left). The latent variable $\boldsymbol{a}$ acts as an abstraction of the context set/PPS, providing smooth control over different arrangements and values, while also allowing the model to flexibly learn the mapping between arrangement of pixel locations and corresponding pixel vales. The full PPS-VAE generative model can thus be defined as

Eq. 3 - PPS-VAE: Generative Model

$$p_\theta(\boldsymbol{a}, \boldsymbol{x}, \boldsymbol{y}|M)$$
$$= p_\theta(\boldsymbol{a})\ p_\theta(\boldsymbol{x}_M|\boldsymbol{a})\ p_\theta(\boldsymbol{y}_M|\boldsymbol{x}_M, \boldsymbol{a})\ p_\theta(\boldsymbol{y}_T|\boldsymbol{x}_T, \boldsymbol{x}_M, \boldsymbol{y}_M)$$

[1] $\boldsymbol{x}$ factorises into $\boldsymbol{x}_M$ and $\boldsymbol{x}_T$.

[2] The parameter θ of the $p_\theta(\boldsymbol{y}_M|\boldsymbol{x}_M)$ distribution is shared among all data instances. Given that the distribution of values in a pixel $\boldsymbol{y}_M$ can vary enormously depending on the (both global and local) arrangements of $\boldsymbol{x}_M$, the model will typically struggle to faithfully learn such a distribution across all data instances. We tackle this issue by introducing an *abstractive* latent variable $\boldsymbol{a}$.

$$\underset{\text{abstract.}}{p_\theta(\boldsymbol{a})} = \mathcal{N}(\boldsymbol{a}|\boldsymbol{0}, \boldsymbol{1})$$

$$\underset{\text{locations}}{p_\theta(\boldsymbol{x}_M|\boldsymbol{a})} = \prod_{m=1}^{M} GS(x_m|g_\theta^1(\boldsymbol{a}))$$

$$\underset{\text{pixel values}}{p_\theta(\boldsymbol{y}_M|\boldsymbol{x}_M, \boldsymbol{a})} = \prod_{m=1}^{M} \mathcal{N}(y_m|g_\theta^2(\boldsymbol{x}_M, \boldsymbol{a}))$$

$$\underset{\text{pixel values}}{p_\theta(\boldsymbol{y}_T|\boldsymbol{x}_T, \boldsymbol{x}_M, \boldsymbol{y}_M)} = \prod_{t=1}^{T} \mathcal{N}(y_t|g_\theta^3(\boldsymbol{x}_T, \boldsymbol{x}_M, \boldsymbol{y}_M))$$

where $g_\theta^1, g_\theta^2,$ andg_θ^3 are parametrised neural networks that transform input values to corresponding distribution parameters, and GS is the Gumbel-Softmax distribution (Maddison et al., 2017; Jang et al., 2017) which provides a continuous relaxation of the discrete distribution—enabling reparametrised gradient estimation.

The standard CNP formulation estimates the marginal $p_\theta(\boldsymbol{y}|M)$ by sampling uniformly at random from $p(\boldsymbol{x}_M)$. One can instead construct a more informative importance-sampled estimator by employing a variational posterior $q_\phi(\boldsymbol{x}_M|\boldsymbol{y}, M)$ in the vein of Kingma & Welling (2014, VAE). Crucially, given a means to generate locations $\boldsymbol{x}_M$, one can simply lookup the image $\boldsymbol{y}$ at those locations to derive $\boldsymbol{y}_M$—an observation itself—as shown in Figure 2 (right). From a representation-learning perspective, the context set can be seen as a *partial pixel specification* (PPS) of the image. The corresponding inference model is

Eq. 4 - PPS-VAE: Inference Model

$$q_\phi(\boldsymbol{a}, \boldsymbol{x}_M|\boldsymbol{y}, M) = q_\phi(\boldsymbol{x}_M|\boldsymbol{y})\; q_\phi(\boldsymbol{a}|\boldsymbol{x}_M, \boldsymbol{y}_M)$$

$$\underset{\text{locations}}{q_\phi(\boldsymbol{x}_M|\boldsymbol{y})} = \prod_{m=1}^{M} GS(x_m|h_\phi^1(\boldsymbol{y}, x_{<m})) \tag{4a}$$

$$\underset{\text{abstract.}}{q_\phi(\boldsymbol{a}|\boldsymbol{x}_M, \boldsymbol{y}_M)} = \mathcal{N}(\boldsymbol{a}|h_\phi^2(\boldsymbol{x}_M, \boldsymbol{y}_M)),$$

where the generative model independently factorises $p_\theta(\boldsymbol{x}_M|\boldsymbol{a})$, and the posterior uses an autoregressive formulation. Again, $h_\phi^1$and h_ϕ^2 are parametrised neural networks that transform inputs to distribution parameters. In Equation (4a), $x_{<m}$ is the set of pixel locations up to step m. For $m = 1$ it is assumed to be null.

Putting the generative and inference models together, we construct the variational evidence lower bound (ELBO) as

$$\log\ p_\theta(\boldsymbol{y}|M) \geq \mathbb{E}_{q_\phi(\boldsymbol{a}, \boldsymbol{x}_M|\boldsymbol{y}, M)}\left[\log \frac{p_\theta(\boldsymbol{a}, \boldsymbol{x}, \boldsymbol{y}|M)}{q_\phi(\boldsymbol{a}, \boldsymbol{x}_M|\boldsymbol{y}, M)}\right],$$

which can be further expanded as

Eq. 5 - PPS-VAE: ELBO

$$\mathbb{E}_{q_\phi(\boldsymbol{a}, \boldsymbol{x}_M|\boldsymbol{y})}\left[\log p_\theta(\boldsymbol{y}_T|\boldsymbol{x}_T, \boldsymbol{x}_M, \boldsymbol{y}_M) p_\theta(\boldsymbol{y}_M|\boldsymbol{x}_M, \boldsymbol{a})\right] - \\ - D_{\text{KL}}\left(q_\phi(\boldsymbol{a}, \boldsymbol{x}_M|\boldsymbol{y}) \| p_\theta(\boldsymbol{a}, \boldsymbol{x}_M)\right)$$

where $\boldsymbol{y}_M$ and $\boldsymbol{y}_T$ are observations derived as $\boldsymbol{y} \odot \boldsymbol{x}_M$ and $\boldsymbol{y} \odot \boldsymbol{x}_T$ respectively—lookups for complementary sets of pixel locations. Note that the abstractive latent $\boldsymbol{a}$ is reversed in the generative vs. inference models—a location x_m sampled from the posterior can only be scored in the generative model once the corresponding $\boldsymbol{a}$ has been sampled. This ensures that the complex transformation involved in $\boldsymbol{x}_M \to \boldsymbol{y}_M$ is captured by the abstractive latent.

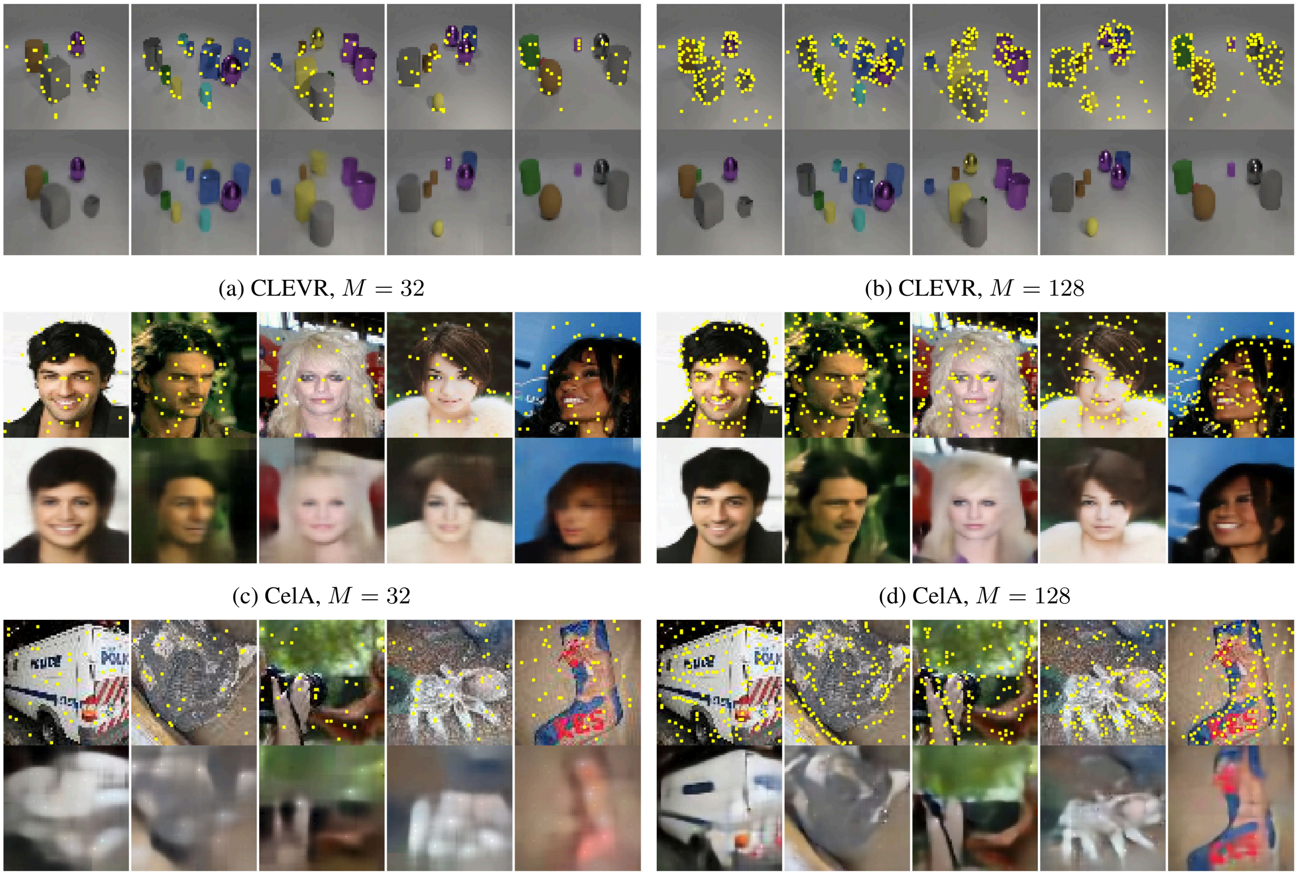
Inductive biases. In the first instance, given our focus on the visual domain, we employ a specific variant of CNPs called the ConvCNP (Gordon et al., 2020), which explicitly incorporates translation equivariance and locality constraints enforced by convolutional neural network (CNN) filters. We use this same inductive bias with CNNs in the inference model $q_\phi(\boldsymbol{x}_M|\boldsymbol{y})$. We find this to be an important design decision, as attempting to model these components using the standard multi-layer perceptron (MLP) Garnelo et al. (as in 2018a) causes issues, primarily with the model using the context set/PPS as a generic lookup table, with little to no spatial meaning (see Appendix D). The CNN-based setup provides the requisite inductive bias that allows meaningful spatial arrangement of points (see Section 3.2).

3. Experiments

Our primary goal here is to understand properties of the context set/PPS. For this we:

- Estimate the log marginal distribution to understand if the learned (rather than randomly sampled) context set helps better model the images (Subsection 3.1),
- Analyse the kinds of points the model chooses; 1-to-1 correspondence between the PPS and an image allows us to perform a visual inspection (see Subsection 3.2),
- Quantify how representative the context set is of the object classes. We do this through the lens of classification, by probing the context set on: 1) in-distribution —PPS-VAE pre-training dataset and the classification dataset are the same, 2) out-of-distribution (OOD) datasets—pre-training dataset differs from the classification dataset (see Subsection 3.3). Moreover we discuss **an ability of the PPS encoder to change capacity during inference**, and
- Demonstrate flexibility and scalability through larger images and OOD reconstruction (see Subsection 3.4).

Datasets. We use four standard vision datasets: FER2013 (Erhan et al., 2013), CelebA (Liu et al., 2015, CelA), CLEVR (Johnson et al., 2017) and Tiny ImageNet (Mn-



(a) CLEVR, $M = 32$

(b) CLEVR, $M = 128$

(c) CelA, $M = 32$

(d) CelA, $M = 128$

(e) t-ImageNet, $M = 32$

(f) t-ImageNet, $M = 128$

Figure 3: Visualisation of the spatial arrangement of the context set for PPS-VAE on three datasets (test images): CLEVR (a,b) and CelA (c,d) and t-ImageNet (f,e). In each figure [a-f] the first row corresponds to the original image, together with the inferred context set denoted by the yellow squares. The second row corresponds to the reconstructed images.

moustafa, 2017, t-ImageNet) with resolution at 64×64.

Models. For all datasets we train PPS-VAE with $M = \{32, 64, 128\}$. Where concerned with performance on a metric, PPS-VAE with $M = 128$ perform the best, results for other M's are in Appendix F. To better ground the experimental results, we employ three baselines: VQ-VAE (van den Oord et al., 2017), FSQ-VAE (Mentzer et al., 2023) and PPS-CAE a variant of CAE (Balın et al., 2019) where we use the same encoder and set ConvCNP (Gordon et al., 2020) as a decoder. Also, in Section 3.1 we fit ConvCNP with the random selection of points and in Subsection 3.3 we use RAND-PPS model — an encoder that randomly samples M points from an image. We train the models once and use them in all the experiments (details in Appendix A).

Table 1: Estimated $\log p_\theta(\boldsymbol{y}|M)(\uparrow)$ with 800 samples. For all models $M = 128$. 🟩 the best performance.

	FER2013	CelA	CLEVR	t-ImageNet
PPS-VAE	4951	14210	16611	16324
PPS-CAE	4471	12162	16089	15832
ConvCNP	4472	12064	15981	15793

3.1. Model Fit

Here we estimate $\log p_\theta(\boldsymbol{y}|M)$ (see Appendix B) and use it to compare the models (see Table 1). The first observation is that PPS-CAE outperforms ConvCNP on all dataset but one, which provides evidence that learning context set helps modelling distribution over the images. The second observation is that learning posterior of the context set (PPS-VAE) instead of just a prior (PPS-CAE) provides a further improvement. In Appendix C, for PPS-VAE, we further estimate $\log p_\theta(\boldsymbol{y}|M)$ for various values of M and find that for all datasets, increasing M results in better performance.

Findings: *Learning (instead of randomly sampling) context set helps modelling distribution over the images.*

3.2. Visual Inspection of PPS

Since there is 1-to-1 correspondence between pixels in the context set an the original image it allows us to perform a qualitative observation of the chosen pixels and put forward

Table 2: Object classification (in-distribution): Classifiers trained over three seeds with early stopping, reporting mean F1-macro scores. A:13—Chubby, A:20—Male, A:25—Oval Face. - ± - best performance among models with an encoder; - ± - absolute best performance. 128→256 means the model was trained on $M = 128$ and evaluated with $M = 256$.

		CelA (A:13)	CelA (A:20)	CelA (A:25)	FER2013	CLEVR	t-ImageNet
Baselines	PPS-RAND (points)	60.92 ± 1.28	90.89 ± 0.06	56.20 ± 0.87	34.97 ± 0.38	36.17 ± 3.39	21.86 ± 0.31
	PPS-RAND (post-hoc tiles)	66.91 ± 1.01	95.10 ± 0.16	60.09 ± 0.24	43.30 ± 0.43	63.20 ± 0.89	33.52 ± 0.23
	PPS-CAE (points)	61.29 ± 0.91	91.35 ± 0.14	58.36 ± 0.45	35.28 ± 0.50	48.50 ± 2.77	22.53 ± 0.30
	PPS-CAE (post-hoc tiles)	67.16 ± 0.90	95.46 ± 0.07	60.75 ± 0.67	44.32 ± 0.66	74.85 ± 0.49	33.55 ± 0.20
	VQ-VAE	68.59 ± 0.04	94.83 ± 0.13	62.44 ± 0.34	50.98 ± 0.52	75.91 ± 0.47	29.02 ± 0.08
	FSQ-VAE	68.19 ± 0.81	95.21 ± 0.11	62.28 ± 0.22	45.46 ± 0.15	73.27 ± 0.36	31.03 ± 0.40
Our	PPS-VAE (points)	69.00 ± 0.38	94.86 ± 0.12	62.13 ± 0.50	46.72 ± 0.62	90.21 ± 0.28	29.56 ± 0.27
	PPS-VAE (points) 128→256	69.94 ± 0.50	95.70 ± 0.07	62.02 ± 0.50	51.61 ± 0.57	93.38 ± 0.64	33.93 ± 0.16
	PPS-VAE (post-hoc tiles)	70.94 ± 0.09	96.21 ± 0.04	62.94 ± 0.10	49.38 ± 0.39	94.62 ± 0.28	35.00 ± 0.04
	Image	73.47 ± 0.49	97.55 ± 0.02	64.49 ± 0.25	61.56 ± 0.17	91.90 ± 0.30	43.68 ± 0.03

hypothesis regarding how PPS-VAE abstracts information for different settings of M. Results are shown in Figure 3, with additional examples given in Appendix L.

The patterns that context sets form can be summarised with the following observations: (1) boundary points between objects and the background generally describe shape, (2) points on the object can capture 'interior' colour, and part locations and (3) background points capture complexity outside the objects (e.g. uniform colour etc.).

We also emphasise that these patterns are more pronounced when M is sufficiently large (e.g. $M = 128$). However, when M relatively small compared to the complexity of an image, the context set appears scattered—possibly because the model tries to "cover" the complexity of the image, by exploring the image space rather than exploiting any region; the former is likely to reconstruct the *whole* image better.

Findings: *The analysis shows that, when M is sufficiently large, the context set forms pronounced patterns with the following three types of points: boundary points around objects, points inside an object and background points.*

3.3. Quantitative Analysis: PPS Probing

Having observed that the context sets/PPS do indeed appear to capture meaningful features, we conduct further analyses to quantify how meaningful they can be. We do this through the lens of classification, by probing the context set/PPS ($\boldsymbol{y}_M$) (in in-distribution and out-of-distribution settings) to see how well it captures class-relevant information. Note that we simply use this as a mechanism to evaluate how well the model captures class-specific information; we do not attempt to engineer a SOTA classifier.

PPS. To evaluate the utility of the context set/PPS $\boldsymbol{y}_M$ suffices (also referred to as points). Using the location variable $\boldsymbol{x}_M$ did not provide further benefit. For all the datasets we set $M = 128$, which is $\approx 3.13\%$ of the original number of pixels. As an additional experiment, we augment $\boldsymbol{y}_M$ at inference time by adding to each pixel in $\boldsymbol{y}_M$ 8 neighbouring pixels—creating 3×3 tiles after pre-training. We call these post-hoc tiles. This achieves two things: (1) increase the amount of information in the latent without retraining the model and (2) test if the points in $\boldsymbol{y}_M$ represent content well enough for a task and if surrounding points help. This augmentation increases the size of PPS to $\approx 28.13\%$ of the original number of pixels.

Baselines. The first baseline employs the whole image $\boldsymbol{y}$ (denoted Image), and is used as a yardstick to see how well a restricted context set does. The second baseline employs a random selection of context points $\boldsymbol{y}_M$ (denoted PPS-RAND) to provide contrast against a more informative selection of context set. Given the discussion in Section 4 of how the FSQ/VQ-VAE[3] can be seen as a selective codebook, but without spatial meaning, we employ it as an additional baseline to see how the constraint of spatial relevance affects classification. Finally, we use PPS-CAE to benchmark global vs instance-specific context sets.

Classification Tasks. The datasets we chose for the pre-training of PPS-VAE and the baseline models come with associated classification tasks. Such that t-ImageNet comes with labels of 200 different classes, FER2013 associate each facial expression with one of the seventh emotion categories, CLEVR comes with labels for number of objects in an image and finally CelebA includes 40 binary attributes associated with facial characteristic. We select 3 generic attributes: A:13 — Chubby, A:20 — Male, A:25 — Oval Face.

Classifier. As the base classifier, we employ the ConvMixer (Trockman & Kolter, 2023) architecture, training each instance entirely from scratch. The encoders of PPS-

[3] See Appendix A.1 for the information about the parametrisation and the codebook size.

Table 3: Object classification — out-of-distribution (OOD) setting. Classifiers trained over three seeds with early stopping, reporting mean F1-macro scores. A:13 — Chubby, A:20 — Male, A:25 — Oval Face. * indicates in-distribution performance copied from Table 2; - ± - indicates the best OOD performance. All PPS based models are evaluated on points.

		CelA (A:13)	CelA (A:20)	CelA (A:25)	CLEVR	t-ImageNet
PPS-VAE	CelA	*69.00 ± 0.38	*94.86 ± 0.12	*62.13 ± 0.50	80.27 ± 1.06	29.59 ± 0.25
	CLEVR	67.02 ± 0.38	93.39 ± 0.09	60.33 ± 0.25	*90.21 ± 0.28	25.05 ± 0.22
	t-ImageNet	67.09 ± 0.34	93.68 ± 0.14	61.28 ± 0.40	80.66 ± 0.59	*29.56 ± 0.27
FSQ-VAE	CelA	*68.19 ± 0.81	*95.21 ± 0.11	*62.28 ± 0.22	69.07 ± 0.63	29.24 ± 0.07
	CLEVR	69.21 ± 0.88	94.90 ± 0.03	62.58 ± 0.25	*73.27 ± 0.36	28.48 ± 0.37
	t-ImageNet	70.04 ± 0.24	95.07 ± 0.02	62.87 ± 0.24	69.35 ± 0.66	*31.03 ± 0.40
VQ-VAE	CelA	*68.59 ± 0.04	*94.83 ± 0.13	*62.44 ± 0.34	68.22 ± 0.31	28.56 ± 0.26
	CLEVR	66.28 ± 0.57	92.93 ± 0.12	60.82 ± 0.27	*75.91 ± 0.47	24.16 ± 0.22
	t-ImageNet	68.92 ± 0.24	94.40 ± 0.07	62.44 ± 0.13	68.26 ± 0.32	*29.02 ± 0.08
PPS-CAE	CelA	*61.29 ± 0.91	*91.35 ± 0.14	*58.36 ± 0.45	37.22 ± 2.53	23.00 ± 0.64
	CLEVR	61.59 ± 0.50	91.77 ± 0.05	58.09 ± 0.50	*48.50 ± 2.77	23.82 ± 0.08
	t-ImageNet	61.80 ± 0.91	90.78 ± 0.08	58.29 ± 0.46	37.22 ± 0.34	*22.53 ± 0.30

VAE and the baseline models: VQ-VAE, FSQ-VAE and PPS-CAE are held frozen during the training of the classifier — only the parameters of ConvMixer are trained. We **do not** perform additional data preprocessing or augmentation. This give us better signal of whether the performance gains are from information encoded in the representations.

In-Distribution vs Out-Of-Distribution Settings. In the in-distribution setting the data used to pre-train the model and the classifier are the same. In the out-of-distribution setting, we take a pre-trained model over a dataset, say t-ImageNet, and evaluate it on say, the CelA and CLEVR datasets. As before, the encoders stay frozen.

3.3.1. In-Distribution setting: Results

Based on Table 2 we make the following observations.

PPS vs Baselines. First, the arrangements of points inferred by PPS-VAE is more indicative of the class than of PPS-RAND. This indicates that the model performs abstraction to preserve the information related to class labels. Second, on average, PPS-VAE performs on par with the baseline models with the pre-trained encoder: FSQ-VAE, VQ-VAE on FER2013, t-ImageNet and CelA datasets, while outperforming the models with a large margin on CLEVR. The performance on CLEVR is associated with identifying a right number of objects, hence the high classification performance achieved by PPS-VAE shows that it has potential to represent abstract object information. Also, not surprisingly, context-set learned by empirical prior of PPS-CAE lags behind of PPS-VAE. This is because PPS-VAE allows us to infer an instance specific context-set while PPS-CAE infers global context set, which may lack instance specific information required for the task. Finally, ConvMixer trained on the original images performs the best on average, which isn't too surprising since $\mathbf{y}$ contains the original information, while the baselines and PPS-VAE learn abstractions which may result in information loss.

Post-hoc Tiles. When augmented with the post-hoc tiles representations inferred by PPS-VAE dominate the baselines only marginally lagging behind VQ-VAE on FER2013. Moreover, the PPS-VAE with post-hoc tiles outperforms the Image baseline on the CLEVR dataset.

Extrapolation of M at Inference. A differentiating property of our model is the ability to increase the capacity of the latent representation (PPS) at inference time. We can encode more information in the context set by simply increasing M, without retraining the model unlike in the case of VQ-VAE and FSQ-VAE. This can be beneficial in scenarios where a downstream task is complex and M used during the training is not high enough (e.g. due to the computational constraints) to encode all the relevant information to achieve desirable performance on the task. Figure 4 depicts what happens to PPS when the capacity is decreased (left im-

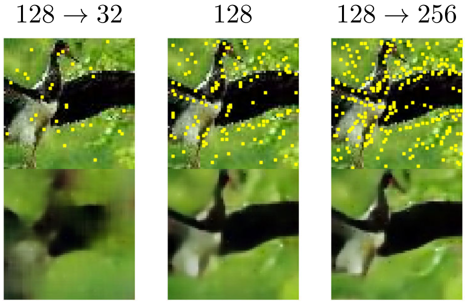


Figure 4: Visualisation of PPS for changing of M at inference time. PPS-VAE was pre-trained with $M = 128$.

Table 4: F1-macro scores. Object classification — out-of-distribution setting but with trained classifiers (trained in 128 → 256 Table 2 experiment).* indicates in-distribution performance copied from Table 2. A:13—Chubby, A:20—Male, A:25—Oval Face.

	CelA	CLEVR	t-ImageNet
CelA (A:13)	*69.94 ± 0.50	68.47 ± 2.39	67.98 ± 1.80
CelA (A:20)	*95.70 ± 0.07	95.71 ± 0.07	95.71 ± 0.05
CelA (A:25)	*62.02 ± 0.50	61.11 ± 0.83	61.28 ± 0.96
CLEVR	92.56 ± 0.40	*93.38 ± 0.64	92.60 ± 0.40
t-ImageNet	33.80 ± 0.40	33.59 ± 0.13	*33.93 ± 0.16

age) or increased (right image). Performance wise when the capacity is increased 128 → 256 the classification performance approaches the post-hoc tiles, even allows achieving best performance on FER2013 among the baselines (see Appendix G).

3.3.2. Out-Of-Distribution setting: Results

We provide results in Table 3 and make the following observations. First, PPS-VAE, FSQ-VAE and VQ-VAE still perform strongly compared to the in-distribution setting. Moreover, while PPS-CAE displays slight increase in performance on most of the datasets except CLEVR, for both in-distribution and OOD settings classification performance is close to random. Second for PPS-VAE, FSQ-VAE and VQ-VAE, pre-training on t-ImageNet allows better generalisation to OOD images than when pre-trained on the other two datasets. Overall we conclude that the context set learned by the PPS-VAE provides a degree of generalisation, but this varies with the dataset. The same applies to FSQ-VAE and VQ-VAE. We provide additional qualitative observations for PPS-VAE in Appendix M.

Extrapolation of M at Inference. We also test if the increased capacity of PPS can be used in the out-distribution setting. We reuse the pre-trained encoders and the classifiers—trained in the previous Subsection see (128 → 256 Table 2 experiment). We report results in Table 4 and note that these are very close to the in-distribution settings suggesting that increased capacity does not jeopardise out-of-distribution generalisation. The classifiers can be reused with minimal loss in the performance if any.

Findings: *In-distribution: probing reveals that 1) the context set preserves class label information which is on par or better than baselines 2) augmented or increased capacity $\mathbf{y}_M$ provides better features for the classifier than the original image $\mathbf{y}$ on CLEVR dataset. Out-of-distribution: representations learned by* PPS-VAE, *FSQ/VQ-VAE are robust to out-of-distribution images and can be used with a slight loss of performance on the tasks associated with the images. The degradation of performance depends on the pre-training dataset.*

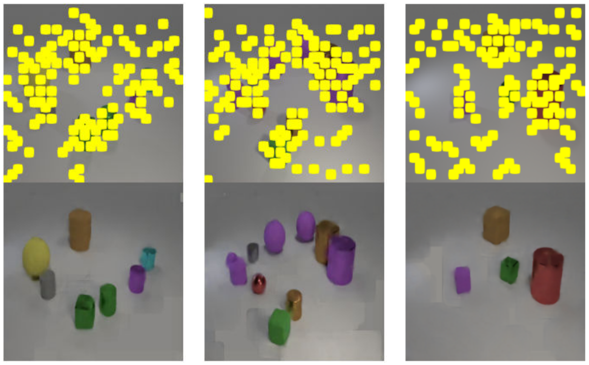


Figure 5: Spatial arrangement of the context set for PPS-VAE tiles. Image size is 256×256, with 8×8 tiles.

3.4. Miscellaneous Properties

3.4.1. Scalability

Encoder of PPS-VAE is autoregressive. As with any autoregressive model, a particular bottleneck is its computational complexity, which gets worse with increasing sequence length (M). Let T be a computation complexity of a computational block (e.g. CNN) and let the encoder and the decoder is build of the same block. Then the computational complexity will be $\mathcal{O}(M * T)$ assuming M is larger than the number of the blocks in the decoder. In this section we discuss how to ameliorate this.

Parallel Inference of Points. One way to speed up the encoder is to make inference of the points in the context set independent of each other — inference of all M points in one shot. However, in previous experiments, we found that it results in inferior performance compared to an autoregressive encoder (see Appendix E). Instead, we use mixture of the two — autoregressive encoder, which at each step infers K points in parallel instead of 1. This reduces complexity to $\mathcal{O}(M/K * T)$. In our experiments we set $K = 8$.

Tiles. PPS-VAE is also scalable to large image size. To achieve this we introduce an additional convolutional layer to the encoder that reduces the resolution of an image to specified size - otherwise the model stays the same. For example, given an image of resolution 256×256 the encoder reduces it to 32×32 by producing non-overlapping tiles of size 8×8 (see Figure 5). The decoding is happening in the original resolution 256×256.

3.4.2. Zero-shot Reconstruction

Additionally, we test if PPS-VAE can reconstruct an image from an out-of-distribution dataset. We take a pre-trained model on one of the three datasets and evaluate on the remaining two. The results can be found in Appendix M.

When PPS-VAE pre-trained on either CelA or t-ImageNet it can reconstruct images from an out-distribution dataset. For example when trained on CelA it can reconstruct geometric shapes of CLEVR or generic object such as car of t-ImageNet, though with a reduced quality. However, when pre-trained on CLEVR the reconstruction is poor and a lot of artefacts are introduced. The same is observed for FSQ/VQ-VAE (see Appendix M).

4. Related Work

CNPs (Garnelo et al., 2018a) are a flexible and scalable framework for modelling distributions over functions. The framework, now more generally referred to as Neural Process Family (NPF) have seen increased popularity, with the different approaches exploring a range of features of the model. One such approach is the adaption of the CNP to properties of the data (Gordon et al., 2020; Kawano et al., 2021; Nassar et al., 2018). Another approach seeks improved modelling of the output dependencies between function values (Garnelo et al., 2018b). Various other approaches exist; see Jha et al. (2022) for an extensive survey. While all such approaches explore the model's features, to the best of our knowledge, none explore the characteristics of the context set itself.

From a representation-learning perspective, the closest to ours is the VQ-VAE (van den Oord et al., 2017). The ability to discretise representation, and learn such a discrete 'codebook' through differentiable variational inference that the VQ-VAE employs, has seen successful use in more advance models such as DALL-E (Ramesh et al., 2021). However the types of codebooks that VQ-VAE learns are not interpretable, and it typically needs additional components, such as learning a separate prior, in order to truly function as a generative model over observed data.

The perspective of learning latent representations/features that apply directly on the data domain, can also be compared to work that exposes *attention* mechanisms (Bahdanau et al., 2015; Mnih et al., 2014) employed for tasks. The process of inferring context points can be interpreted as a locally-restrictive way of attending to relevant parts of the image data. Specifically, such a perspective aligns best with hard attention methods (Mnih et al., 2014) as opposed to soft-attention (Bahdanau et al., 2015) by virtue of explicitly selecting particular pixels.

Furthermore, inferring context points can also be viewed as a variant of Masked Image Modelling (Pathak et al., 2016, MIM). MIM involves learning models and representation in a self-supervised fashion by masking parts of an image and attempting to impute them. More recently, this has been studied extensively as masked autoencoders (He et al., 2022, MAE). The imputation task itself is strongly connected to what CNPs do, and one could ask a similar question of MIMs that we ask of CNPs: what kinds of masks do MIMs like to impute? In fact such a question was indeed asked in work by Shi et al. (2022, ADIOS) who learnt masks simultaneously with a feature extractor in an adversarial fashion. This however, is not generative, and as with as masking-based approaches, involves complications with how to specify and generate masks in a sensible manner. A key distinction is in terms of the sparsity of observed data—MIM and related approaches typically imputes a small part of the image, where CNPs have a more complex task given sparse input. PPS-VAE employs context points as weak specifiers of which parts of the image to contextualise, leaving to the CNP itself the question of how to use that specification to capture relevant local and global information from the data.

5. Discussion

We present PPS-VAE, a novel VAE framework that allows us to infer context set/PPS for conditional neural processes (CNPs). We formulate our model and evaluate it across multiple vision datasets, while exploring the utility of learning context sets in both unsupervised and supervised manner. First, we show that the learning distribution over PPS results in better models for images. Then, we observe that with the appropriate inductive biases and latent variables, the model is able to induce context sets that are visually meaningful. We validate this observation quantitatively through the lens of classification. On the classification tasks, PPS-VAE achieves superior performance against the baselines and PPS resulting in better features for a classifier than an original image is on the CLEVR dataset indicating that the framework has promise as a model for learning meaningful representations of data. Additionally, we test our model on the same classification tasks but in out-of-distribution settings showing that it can infer PPS that generalises to an out-of-distribution datasets. Also, we show a differentiating property of PPS-VAE — an ability to change the capacity of PPS at inference time. Our model, however, has a number of limitation which we would like to outline:

- Presently we provide an observatory analysis of the induced context set and put forward hypothesis regarding types of points the model learns. However, human level interpretability of the context set is limited. To improve it, instead of inferring a single location, a more interpretable encoder could capture M 'closed' regions. This would allow us to compare against the slot-attention models such as Locatello et al. (2020).
- Exploration of inductive biases, and modelling updates would be interesting avenues to see if the latent variable can capture relevant information more cleanly.
- Presently we fix M to a certain value and provide analysis

for its various values. However, it may be limiting to decide on the value of M beforehand because we do not know what value would be optimal for each image in a dataset. Allowing the model to decide on the value of M during the learning based on dataset may solve this issue.

6. Rebuttal Clarifications

6.1. How the three tasks for CelebA were chosen?

The subset of features we chose were intended to help discriminate generic whole-image attributes rather than simply the specific attributes e.g. presence of glasses. The idea being that identifying the presence of a feature/object that had no interesting localisation features would not tell us much about the usefulness of the PPS representation—glasses for people in CelebA occur roughly in the same image locations in the same form across the data, as does a beard, etc., whereas chubbiness, (labelled) gender, and facial shape types are more interesting from a spatial layout and associated-feature perspective.

6.2. The role of the latent variable a in the model (Section 2)

Consider a case where $\boldsymbol{x}_M$ denotes the locations of the useful pixels for an image of someone's face. While knowing the locations could indicate something useful about the structure of the face, just knowing the locations does not tell us about the eye colour, or hair colour, of the specific instance being reconstructed. Since the parameter θ is a globally learnt one (i.e it learns a single value across dataset, and is not conditioned on any one image instance), the distribution defined by $p(\boldsymbol{y}_M|\boldsymbol{x}_M)$ can be quite high variance.

To alleviate this problem, we use the abstractive variable $\boldsymbol{a}$ to help capture both arrangement of points and some information about their associated values—so something like blonde-hair and brown eyes can be better associated together if such associations are prevalent in the data. For example, one can view sampling from $\boldsymbol{a}$ as saying sample points for a 'chubby' face (spatial layout) with 'green eye' (feature value), and then use it to condition the actual sampling of $p(\boldsymbol{x}_M|\boldsymbol{a})$, and $p(\boldsymbol{y}_M|\boldsymbol{x}_M, \boldsymbol{a})$.

6.2.1. Why not using a NP (Garnelo et al., 2018b) for $p_\theta(\boldsymbol{y}_M|\boldsymbol{x}_M, \boldsymbol{a})$ that already has an inherent latent variable?

(C)NPs with an inherent latent variable motivates its use to help prediction given context points. That is, they use the entire context set $(\boldsymbol{x}_M, \boldsymbol{y}_M)$ to help predict the value $p(\boldsymbol{y}_T|\boldsymbol{x}_T, \boldsymbol{z})$ for the inherent latent $\boldsymbol{z}$. We on the other hand, employ the abstractive latent $\boldsymbol{a}$ to predict the context set itself; i.e. we wish to obtain $p(\boldsymbol{x}_M|\boldsymbol{a}), p(\boldsymbol{y}_M|\boldsymbol{x}_M, \boldsymbol{a})$, not the target predictions $(\boldsymbol{x}_T, \boldsymbol{y}_T)$. The difference is that in LNPFs (Dubois et al., 2020), the latent variable lies between the context set and target set. In our case, it comes before the context set.

6.3. Improved performance of PPS-VAE on CLEVR over the full image (Table 2)

Our hypothesis is that using a representative subset of pixels that encodes sufficient information to reconstruct the image helps regulate the signal-to-noise ratio of the information necessary for any given classifier. While the relevant information is likely present in both the PPS representation and the whole image, for a fixed classifier, lower noise, by avoiding redundant or extraneous pixels, can result in better performance. For example, a colour of an object may be similar to the background colour and this may confuse a classifier, while PPS-VAE helps focus on the objects.

6.4. Why we did not use the newer CNPs?

It is indeed true that our core idea is agnostic to the type of CNP employed in the decoder and the newer models e.g. Nguyen & Grover (2022); Feng et al. (2023a) can be incorporated into our PPS-VAE framework. However, the idea we explore is if, and how well, a model can learn to choose its own context set, given a sufficiently capable 'completion' model. As described in Appendix D, we find that the inductive bias provided by the CNNs plays an important role in ensuring that the model learns to choose context points that capture spatial regularities and patterns.

Acknowledgements

The authors gratefully acknowledge the support of ELIAI (The Edinburgh Laboratory for Integrated Artificial Intelligence) EPSRC (grant no EP/W002876/1). Ivan Titov also acknowledges Dutch National Science Foundation (NWO Vici VI.C.212.053). The authors are also thankful to Mattia Opper for providing useful feedback on the manuscript.

Impact Statement

The work we describe in this paper aims to improve the interpretability of the latent representations. We foresee that further development of the described algorithm may allow to overcome a number of drawbacks of Deep Generative Models (DGMs) e.g. Feng et al. (2023b); Conwell & Ullman (2022) by addressing these issues at the representation level (e.g. by explicit manipulation of latent representations to rectify mistakes DGMs). However, the current work is algorithmic in nature. And at present stage is not tied to particular applications, let alone deployments.

References

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations (ICLR)*, 2015.

Balın, M. F., Abid, A., and Zou, J. Concrete autoencoders: Differentiable feature selection and reconstruction. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 444–453. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/balin19a.html.

Conwell, C. and Ullman, T. Testing relational understanding in text-guided image generation, 2022.

Dubois, Y., Gordon, J., and Foong, A. Y. Neural process family. http://yanndubs.github.io/Neural-Process-Family/, September 2020.

Erhan, D., Goodfellow, I., Cukierski, W., and Bengio, Y. Challenges in representation learning: Facial expression recognition challenge, 2013. URL https://kaggle.com/competitions/challenges-inrepresentation-learning-facial-expression-recognition-challenge.

Feng, L., Hajimirsadeghi, H., Bengio, Y., and Ahmed, M. O. Latent bottlenecked attentive neural processes. In *The Eleventh International Conference on Learning Representations*, 2023a. URL https://openreview.net/forum?id=yIxtevizEA.

Feng, W., He, X., Fu, T.-J., Jampani, V., Akula, A., Narayana, P., Basu, S., Wang, X. E., and Wang, W. Y. Training-free structured diffusion guidance for compositional text-to-image synthesis, 2023b.

Garnelo, M., Rosenbaum, D., Maddison, C., Ramalho, T., Saxton, D., Shanahan, M., Teh, Y. W., Rezende, D. J., and Eslami, S. M. A. Conditional neural processes. In *International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1690–1699, 2018a.

Garnelo, M., Schwarz, J., Rosenbaum, D., Viola, F., Rezende, D. J., Eslami, S. M. A., and Teh, Y. W. Neural processes. *CoRR*, abs/1807.01622, 2018b.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Teh, Y. W. and Titterington, M. (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL https://proceedings.mlr.press/v9/glorot10a.html.

Gordon, J., Bruinsma, W. P., Foong, A. Y. K., Requeima, J., Dubois, Y., and Turner, R. E. Convolutional conditional neural processes. In *International Conference on Learning Representations (ICLR)*, 2020.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations (ICLR)*, 2017.

Jha, S., Gong, D., Wang, X., Turner, R. E., and Yao, L. The neural process family: Survey, applications and perspectives, 2022.

Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Lawrence Zitnick, C., and Girshick, R. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Kawano, M., Kumagai, W., Sannai, A., Iwasawa, Y., and Matsuo, Y. Group equivariant conditional neural processes. In *International Conference on Learning Representations (ICLR)*, 2021.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *International Conference on Computer Vision (ICCV)*, pp. 3730–3738, 2015.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. A convnet for the 2020s. 2022.

Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., and Kipf, T. Object-centric learning with slot attention, 2020.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Maddison, C. J., Mnih, A., and Teh, Y. W. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations (ICLR)*, 2017.

Mentzer, F., Minnen, D., Agustsson, E., and Tschannen, M. Finite scalar quantization: Vq-vae made simple, 2023.

Mnih, V., Heess, N., Graves, A., and Kavukcuoglu, K. Recurrent models of visual attention. In *Advances in Neural Information Processing Systems (NeuRIPS)*, pp. 2204–2212, 2014.

Mnmoustafa, M. A. Tiny imagenet, 2017. URL https://kaggle.com/competitions/tiny-imagenet.

Nassar, M., Wang, X., and Tumer, E. Conditional graph neural processes: A functional autoencoder approach. *Third Workshop on Bayesian Deep Learning (NeurIPS 2018)*, abs/1812.05212, 2018.

Nguyen, T. and Grover, A. Transformer neural processes: Uncertainty-aware meta learning via sequence modeling. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16569–16594. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/nguyen22b.html.

Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T., and Efros, A. A. Context encoders: Feature learning by inpainting. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2536–2544, 2016.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. Zero-shot text-to-image generation. In *International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8821–8831, 2021.

Reddi, S. J., Kale, S., and Kumar, S. On the convergence of adam and beyond. In *International Conference on Learning Representations (ICLR)*, 2018.

Shi, Y., Siddharth, N., Torr, P. H., and Kosiorek, A. R. Adversarial masking for self-supervised learning. In *International Conference on Machine Learning (ICML)*, 2022.

Trockman, A. and Kolter, J. Z. Patches are all you need? *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=rAnB7JSMXL. Featured Certification.

van den Oord, A., Vinyals, O., and Kavukcuoglu, K. Neural discrete representation learning. In *Advances in Neural Information Processing Systems (NeuRIPS)*, pp. 6306–6315, 2017.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017. URL https://arxiv.org/abs/1708.07747.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Póczos, B., Salakhutdinov, R., and Smola, A. J. Deep sets. In *Advances in Neural Information Processing Systems (NeuRIPS)*, pp. 3391–3401, 2017.

A. Implementation and Training of PPS-VAE

We parameterise the PPS-VAE with CNN neural networks. More concretely, we use convolutional blocks similar to ConvNetXt (Liu et al., 2022), with *Leaky ReLU* activation function. We found that with *GELU* activation function training of PPS-VAE can be unstable. Also, we do not decrease the H×W dimensions of the original image, hence the induced $x_{1:M} \in \{0, 1\}^{B\times H\times W\times 1}$ and $y_{1:M} \in [0, 1]^{B\times H\times W\times C}$, where B is the batch size. However, we represent the latent variable a in a vector such that $a \in \mathbb{R}^{B\times D}$. We set D to be 32, however any other values would work.

We optimise the parameters of the model with the AdamW (Loshchilov & Hutter, 2019) optimiser, setting the learning rate to $2 * 10^{-4}$ and we also enable the amsgrad (Reddi et al., 2018). We train the PPS-VAE for 200 epochs This is sufficient for the models to converge on the datasets (**we provide code, which includes the implementation of the model**).

A.1. Baseline Models

A.1.1. VQ-VAE

The encoder of the VQ-VAE comprises of the 2 vanilla convolutional layers and 3 ResNet blocks. The decoder comprises of the 3 ResNet blocks and two transposed convolutions. Between the layers we insert *ReLU* activation function. The codebook is initialised with the *xavier uniform* initialiser (Glorot & Bengio, 2010). The latent representation of an image $z \in \mathbb{R}^{B\times J\times S\times D}$, where J and S are the reduced height and width of the original image and D is the dimensionality of the vectors in the codebook. For each of the datasets the number of the scalars in the codebook matches the number of elements in an original image. For example if an image has 3 colour channels and resolution of 64x64 then the total number of the scalar elements in the code book will be 64*64*3. The are multiple of ways to achieve this, we stick to the following. We set the number of vectors in the codebook to 64 and the dimensionality of the vectors to 64*3. We optimise the parameters of the model with the AdamW optimiser, setting the learning rate to $2 * 10^{-4}$ and we also enable the amsgrad. We train the models for 200 epochs.

A.1.2. FSQ-VAE

For FSQ-VAE we reuse the encoder and the decoder architecture of the VQ-VAE. We set the following number of of levels per channel: for the colored images: $[8, 8, 7, 6, 5]$ and $[6, 6, 5, 5, 5]$ for black and white images. As in the VQ-VAE we choose the leves to roughly match the number of elements in an image. We optimise the parameters of the model with the AdamW optimiser, setting the learning rate to $2 * 10^{-4}$ and we also enable the amsgrad. We train the models for 200 epochs.

A.1.3. PPS-CAE

The encoder of PPS-CAE model comprises of M 64*64 learnable parameters. These parameters are used to parameterise the Gumbel-Softmax distribution. We use the same $p_\theta(\boldsymbol{y}_T|\boldsymbol{x}_T, \boldsymbol{x}_M, \boldsymbol{y}_M)$ decoder as for PPS-VAE. We optimise the parameters of the model with the AdamW optimiser, setting the learning rate to $2 * 10^{-4}$ and we also enable the amsgrad. We train the models for 200 epochs.

A.2. ConvMixer Classifier

Our ConvMixer classifier has four hyperparameters: depth, kernel size, number of filters and patch size.

Table 5: ConvMixer parameters.

	PPS-VAE	FSQ-VAE	VQ-VAE	PPS-CAE	Image
depth:	5	5	5	5	5
kernel size:	11	3	3	11	11
number of filters:	256	256	256	256	256
path size:	1	1	1	1	1

We had to reduce the kernel size for the VQ-VAE and FSQ-VAE because it resulted in overfitting and mainly inferior performance.

B. Log Marginal Likelihood Estimation

In this section we show how we estimate log marginal likelihood of $\log p_\theta(\boldsymbol{y}|M)$.

B.1. Estimation for PPS-VAE

$$\log p_\theta(\boldsymbol{y}|M) \approx \log \left[\frac{1}{N} \sum_{i=1}^{N} \frac{p_\theta(\boldsymbol{a}^i, \boldsymbol{x}^i, \boldsymbol{y}|M)}{q_\phi(\boldsymbol{a}^i, \boldsymbol{x}^i_M|\boldsymbol{y}, M)} \right]; \qquad \boldsymbol{a}^i, \boldsymbol{x}^i_M \sim q_\phi(\boldsymbol{a}, \boldsymbol{x}_M|\boldsymbol{y}, M)$$

B.2. Estimation for ConvCNP and PPS-CAE

Given the CNP's generative model:

$$p_\theta(\boldsymbol{x}, \boldsymbol{y}|M) = p_\theta(\boldsymbol{x}_M)\ p_\theta(\boldsymbol{y}_M|\boldsymbol{x}_M)\ p_\theta(\boldsymbol{y}_T|\boldsymbol{x}_T, \boldsymbol{x}_M, \boldsymbol{y}_M)$$

we estimate the log marginal likelihood as:

$$\log p_\theta(\boldsymbol{y}|M) \approx \log \left[\frac{1}{N} \sum_{i=1}^{N} p_\theta(\boldsymbol{y}_T|\boldsymbol{x}^i_T, \boldsymbol{x}^i_M, \boldsymbol{y}^i_M) \right]; \qquad \boldsymbol{x}^i_M \sim p_\theta(\boldsymbol{x}_M),$$

where $p_\theta(\boldsymbol{y}_M|\boldsymbol{x}_M)$ is delta function (=1), because of the deterministic look of $\boldsymbol{y}_M$ values.

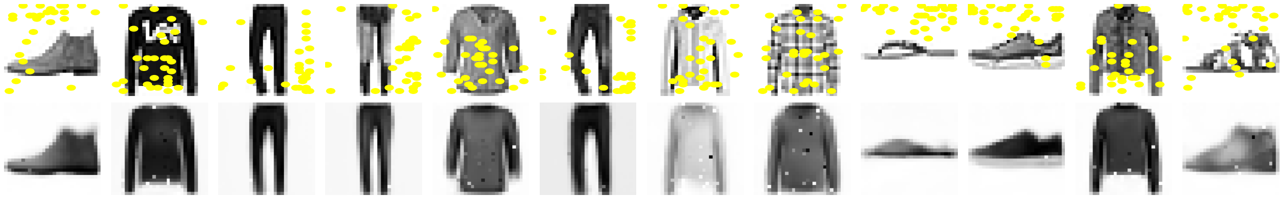
C. Log Marginal Likelihood vs M

Table 6: Estimated $\log p_\theta(\boldsymbol{y}|M)(\uparrow)$ for PPS-VAE (see Appendix B) with 800 samples.

	FER2013	CelA	CLEVR	t-ImageNet
$M = 32$	4111	11569	16529	15645
$M = 64$	4711	13251	16604	16269
$M = 128$	4951	14210	16611	16324

D. Inductive Bias: MLP CNP

In the earlier version of the PPS-VAE model, we found that the parametisation of the model with MLP layers as in (Garnelo et al., 2018a) may bias the model to infer points around the edges of an image (see Figure 6a).



(a) $M = 30$

Figure 6: Visualisation of the spatial arrangement of points in the context sets for the CNP decoder parameterised by the MLP — conducted on the FashionMNIST dataset. The first row corresponds to the original image, together with the inferred context set denoted by the yellow circles. The second row corresponds to the reconstructed images. The context sets inferred on the test images.

Table 7: Object classification for two datasets: FashionMNIST (Xiao et al., 2017) and CIFAR10 (Krizhevsky & Hinton, 2009). $M = 60$. Resnet-18 classifiers trained from scratch over three seeds with early stopping, reporting mean F1-macro scores. PPS-VAE$^{\text{a}}$ is the PPS-VAE with the autoregressive encoder used in the main paper. PPS-VAE$^{\text{i}}$ is the PPS-VAE with independent encoder over $\boldsymbol{x}_M$: $q_\phi(\boldsymbol{x}_M|\boldsymbol{y}) = \prod_{m=1}^{M} GS(x_m|h_\phi(y))$, where h_ϕ is a parametrised neural network that transform inputs to distribution parameters

	FashionMNIST	**CIFAR-10**
PPS-VAE$^{\text{a}}$ (points)	88.0 ± 0.0	76.7 ± 0.5
PPS-VAE$^{\text{i}}$ (points)	86.0 ± 0.0	68.0 ± 0.0

Table 8: Object classification. PPS-VAE (M vs F1). Classifiers trained over three seeds with early stopping, reporting mean F1-macro scores. A:13 — Chubby, A:20 — Male, A:25 — Oval Face.

	CelA (A:13)	CelA (A:20)	CelA (A:25)	FER2013	CLEVR	t-ImageNet
$M = 32$	57.62 ± 0.85	88.28 ± 0.07	57.57 ± 0.88	29.19 ± 3.73	68.58 ± 0.17	13.59 ± 1.72
$M = 64$	63.46 ± 0.66	91.81 ± 0.03	60.04 ± 0.70	40.18 ± 0.41	84.91 ± 1.43	21.49 ± 1.75
$M = 128$	69.00 ± 0.38	94.86 ± 0.12	62.13 ± 0.50	46.72 ± 0.62	90.21 ± 0.28	29.56 ± 0.27

E. Parallel vs Autoregressive Encoder

F. Classification Results: Number of Points in Context Sets vs Classification Performance

G. Classification Results: Increasing Number of Points in Context Sets at Inference Time

Table 9: Object classification. PPS-VAE (M vs F1). Classifiers trained over three seeds with early stopping, reporting mean F1-macro scores. A:13 — Chubby, A:20 — Male, A:25 — Oval Face.

	CelA (A:13)	CelA (A:20)	CelA (A:25)	FER2013	CLEVR	t-ImageNet
$M = 32 \rightarrow 64$	62.81 ± 0.50	91.34 ± 0.13	58.74 ± 0.70	39.57 ± 0.21	82.61 ± 1.65	20.30 ± 1.23
$M = 32 \rightarrow 128$	65.54 ± 0.32	92.80 ± 0.11	60.26 ± 0.99	45.16 ± 0.33	87.59 ± 0.32	25.78 ± 0.20
$M = 64 \rightarrow 128$	67.50 ± 0.68	94.11 ± 0.06	61.68 ± 0.42	47.23 ± 0.32	91.15 ± 0.37	27.98 ± 0.70
$M = 128 \rightarrow 256$	69.94 ± 0.50	95.70 ± 0.07	62.02 ± 0.50	51.61 ± 0.57	93.38 ± 0.64	33.93 ± 0.16

H. Classification Results: Evaluating Latent Variable $\boldsymbol{a}$

Table 10: Object classification. Benchmarking latent variable $\boldsymbol{a}$ against vanilla VAE. Classifiers trained over three seeds with early stopping, reporting mean F1-macro scores. A:13 — Chubby, A:20 — Male, A:25 — Oval Face.

	CelA (A:13)	CelA (A:20)	CelA (A:25)	FER2013	CLEVR	t-ImageNet
VAE	59.04 ± 0.66	86.26 ± 0.12	58.60 ± 0.27	36.06 ± 0.34	41.88 ± 0.28	10.05 ± 0.05
PPS-VAE ($\boldsymbol{a}$)	54.26 ± 0.42	83.52 ± 0.09	55.42 ± 0.24	20.83 ± 0.13	39.62 ± 0.11	8.50 ± 0.13

VAE Model: The encoder of the VAE baseline comprises of five convolutional layers: 3 are the vanilla convolutions with *Leaky ReLU* activation function inserted between them and 2 vanilla convolutions to model the mean and variance of the variational posterior distribution, which is Gaussian. Both the mean and the variance are 32 dimensional vectors. The architecture of the encoder resembles the parametrisation of $q_\phi(\boldsymbol{a}|\boldsymbol{x}_M, \boldsymbol{y}_M)$. The decoder comprises of five transposed convolutions with *Leaky ReLU* activation function inserted between them. We optimise the parameters of the model with the AdamW optimiser, setting the learning rate to $2 * 10^{-4}$ and we also enable the amsgrad.

I. Compute

We run each experiment using the hardware specified in Table 11.

Table 11: Computing infrastructure.

hardware	specification
CPU	AMD® EPYC 7413 24-Core Processor
GPU	NVIDIA® A40 x 1

J. Parameters Count

Table 12: Number of parameters in a model.

	PPS-VAE	FSQ-VAE	VQ-VAE	PPS-CAE
# parameters	6,183,740	10,541,832	11,511,747	5,278,982

To calculate total number of parameters in the model we use:

```
params = sum(p.numel() for p in model.parameters() if p.requires_grad)
```

K. Algorithm

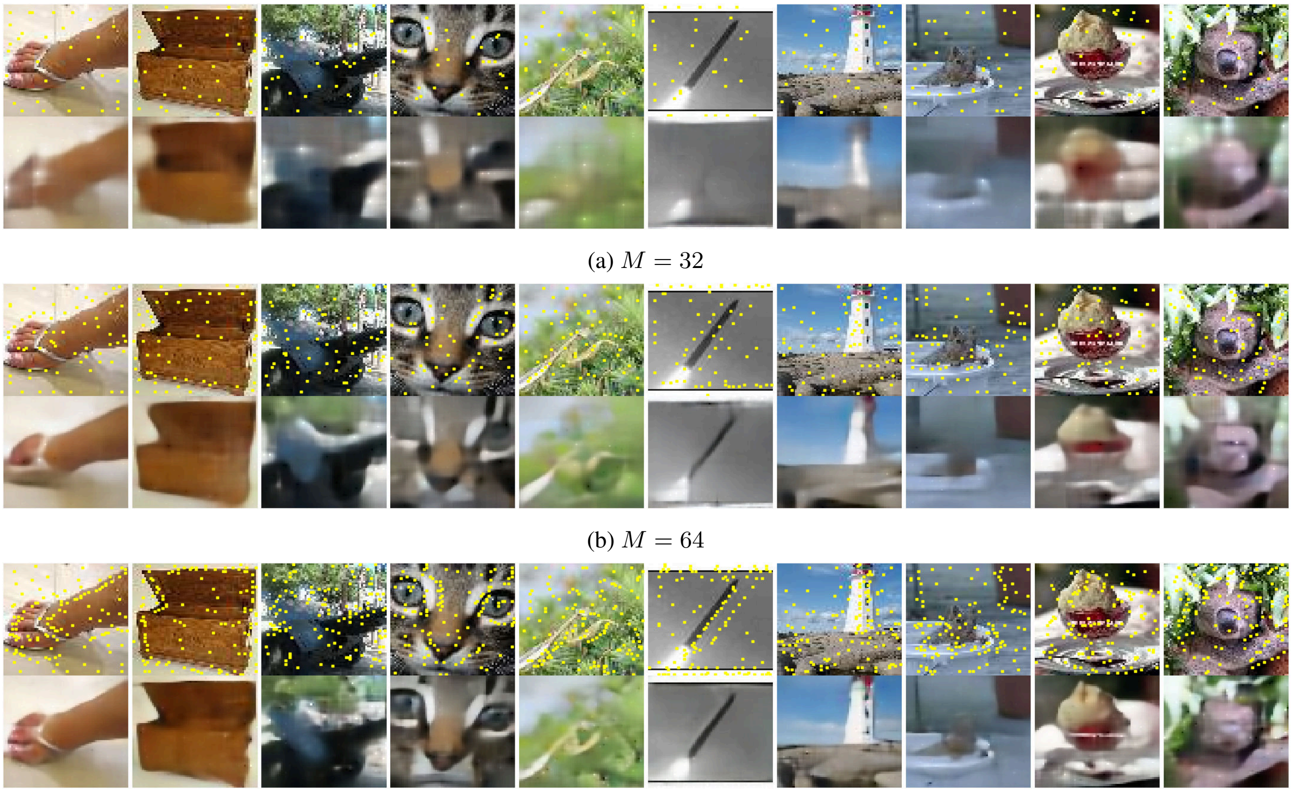
Algorithm 1 PPS-VAE

```
// ** Inference **
Input: $y \in \mathbb{R}^{C \times H \times W}$
Initialize $x_0 \in \{0, 1\}^{C \times H \times W} = 0$, $x_{1:M} = [x_0]$
for $i = 1$ to $M$ do
    $x_i \sim q_\phi(x_i | y, x_{1:M}[0 : i])$
    $x_{1:M}.append(x_i)$
end for
$x_{1:M} = \text{sum}(x_{1:M}, \text{axis} = 0) \in \{0, 1\}^{1 \times H \times W}$
// points can be sampled twice, remove duplicates
$x_{1:M} = x_{1:M} / x_{1:M}$
$y_{1:M} = y * x_{1:M}$
$a \sim q_\phi(a | x_{1:M}, y_{1:M})$
// ** Scoring **
$D_{KL}$ = $(\log q_\phi(x_{1:M} | y) - \log p_\theta(x_{1:M} | a))$ +
          +$(q_\phi(a | y_{1:M}, x_{1:M}) - p_\theta(a))$
// get the target variables
$x_{1:T} = 1 - x_{1:M}$, $y_{1:T} = y * x_{1:T}$
loss($y_{1:M}$) = $\log p_\theta(y_{1:M} | x_{1:M}, a)$
loss($y_{1:T}$) = $\log p_\theta(y_{1:T} | y_{1:M}, x_{1:M}; x_{1:T})$
```

L. Visualisation of Reconstructed Images

L.1. PPS-VAE

L.1.1. DATASET: T-IMAGENET



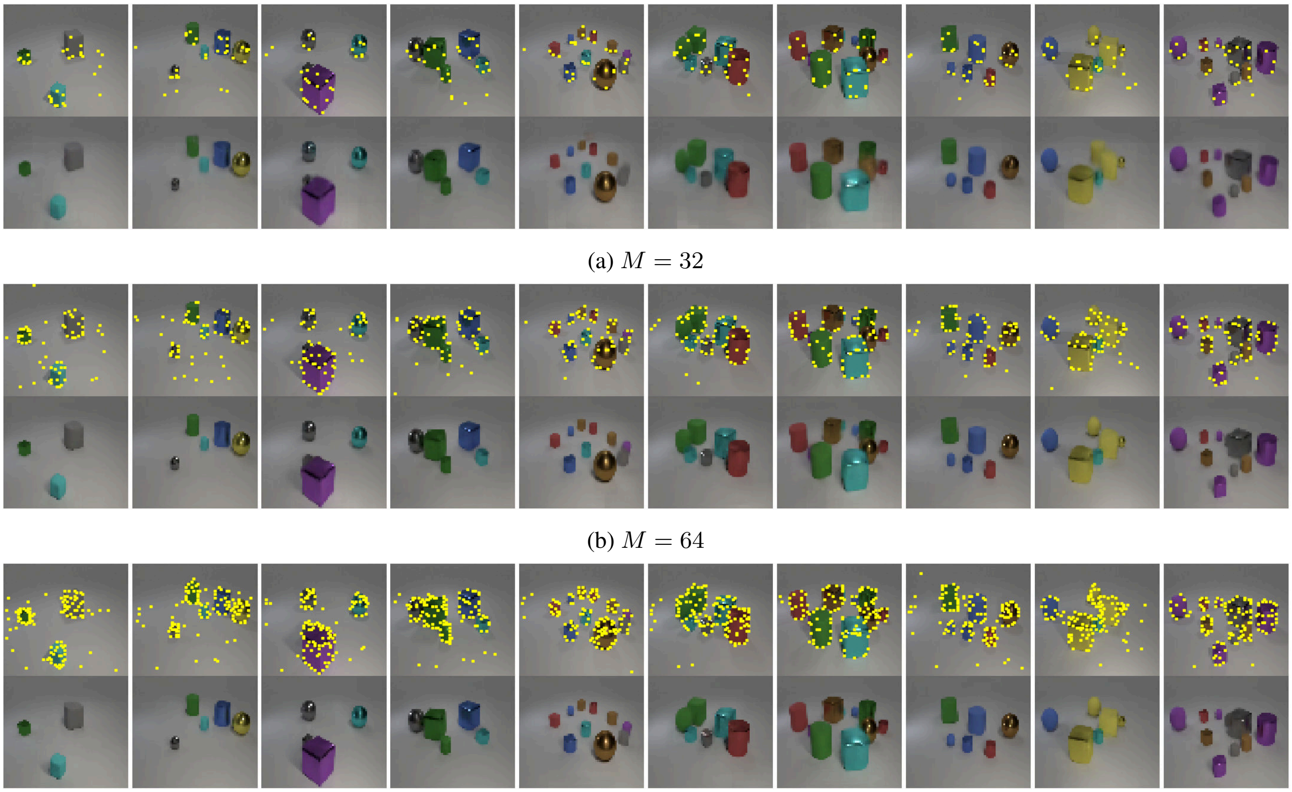
(a) $M = 32$

(b) $M = 64$

(c) $M = 128$

Figure 7: Visualisation of the spatial arrangement of points in the context sets. The first row corresponds to the original image, together with the inferred context set denoted by the yellow circles. The second row corresponds to the reconstructed images. The context sets inferred on the test images.

L.1.2. Dataset: CLEVR



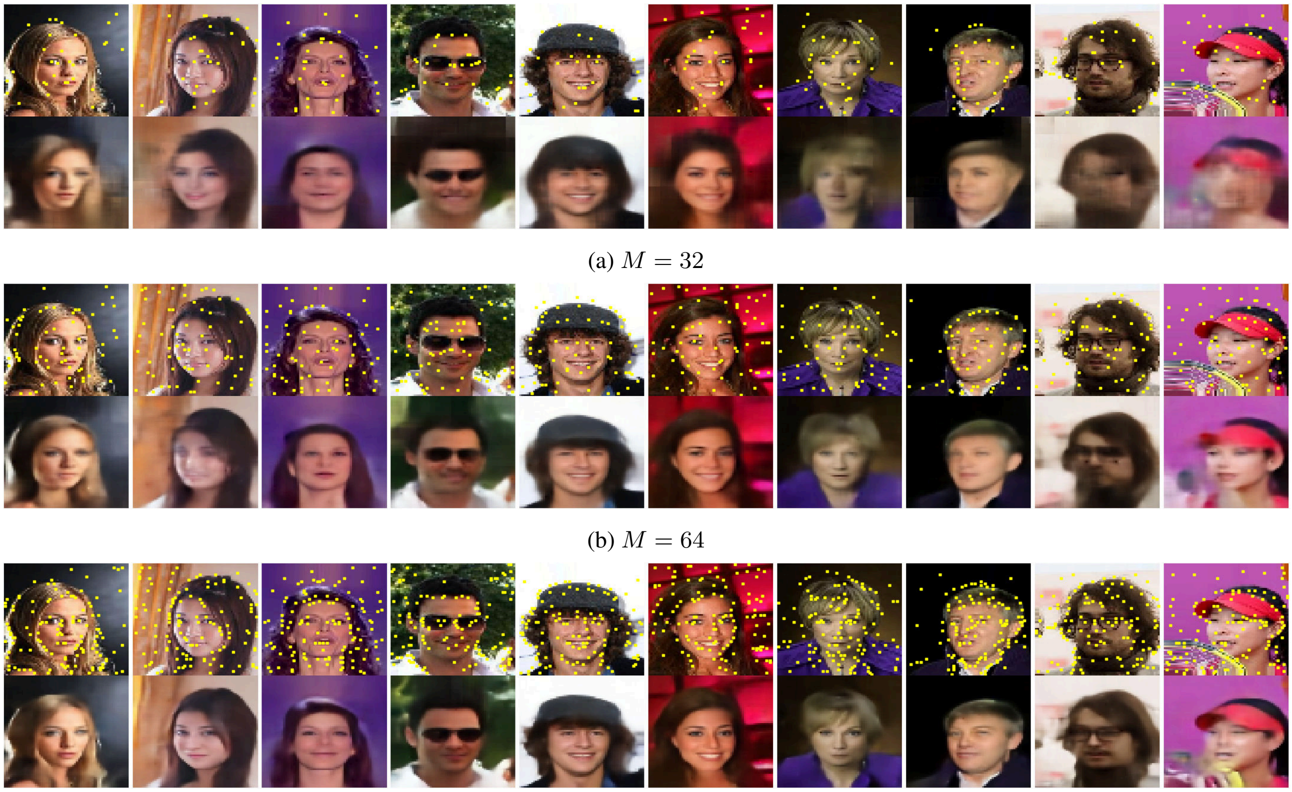
(a) $M = 32$

(b) $M = 64$

(c) $M = 128$

Figure 8: Visualisation of the spatial arrangement of points in the context sets. The first row corresponds to the original image, together with the inferred context set denoted by the yellow circles. The second row corresponds to the reconstructed images. The context sets inferred on the test images.

L.1.3. DATASET: CELEBA



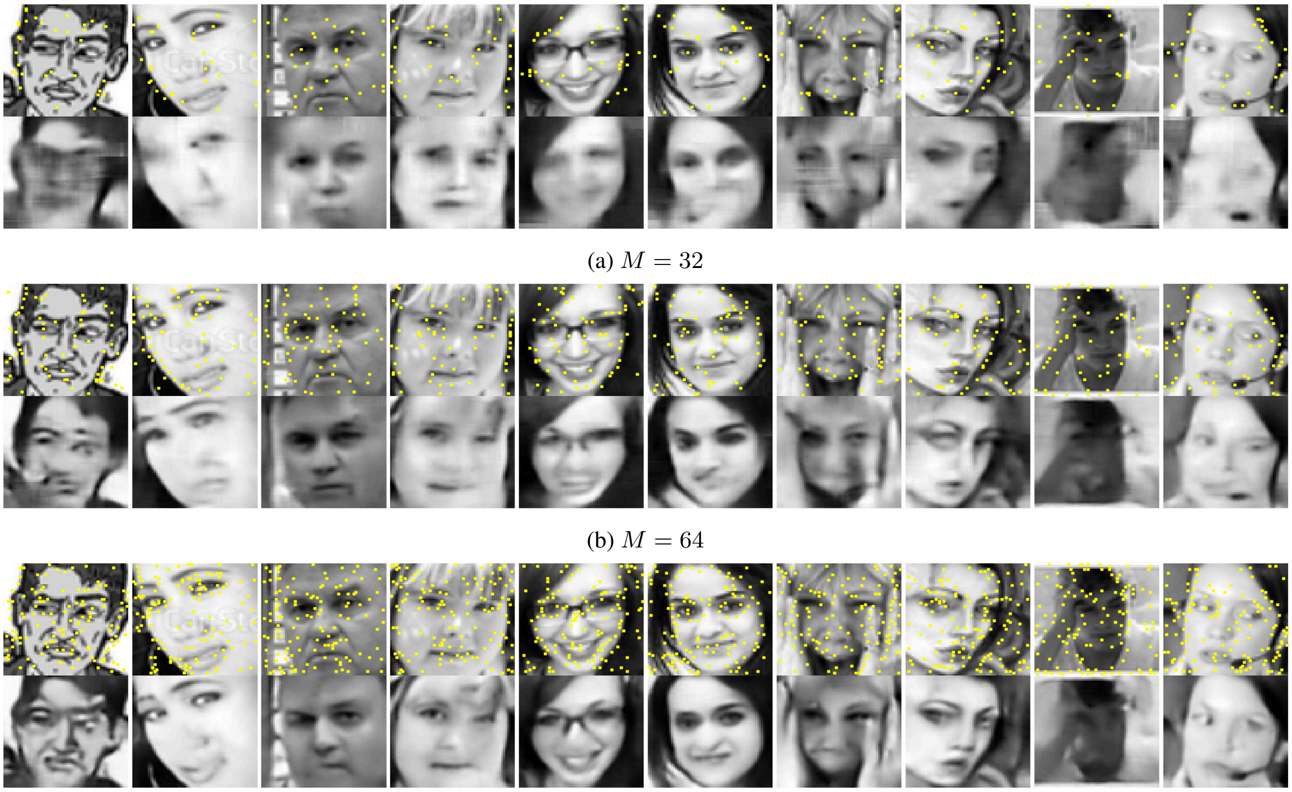
(a) $M = 32$

(b) $M = 64$

(c) $M = 128$

Figure 9: Visualisation of the spatial arrangement of points in the context sets. The first row corresponds to the original image, together with the inferred context set denoted by the yellow circles. The second row corresponds to the reconstructed images. The context sets inferred on the test images.

L.1.4. DATASET: FER2013



(a) $M = 32$

(b) $M = 64$

(c) $M = 128$

Figure 10: Visualisation of the spatial arrangement of points in the context sets. The first row corresponds to the original image, together with the inferred context set denoted by the yellow circles. The second row corresponds to the reconstructed images. The context sets inferred on the test images.

L.2. VQ-VAE

L.2.1. DATASET: T-IMAGENET

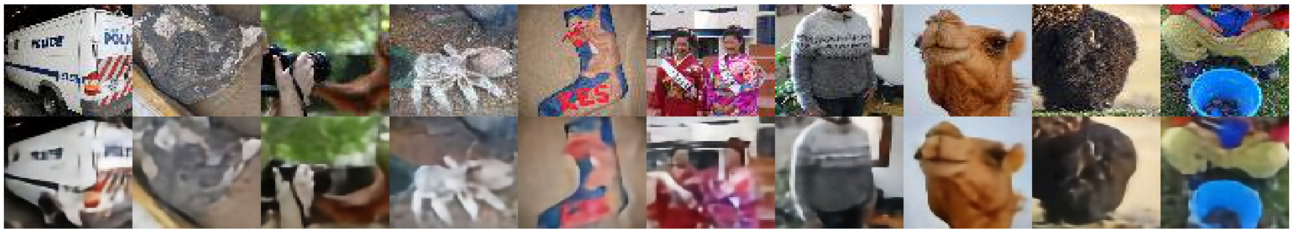


Figure 11: The first row corresponds to the original image. The second row corresponds to the reconstructed images. Evaluated on the test images.

L.2.2. Dataset: CLEVR

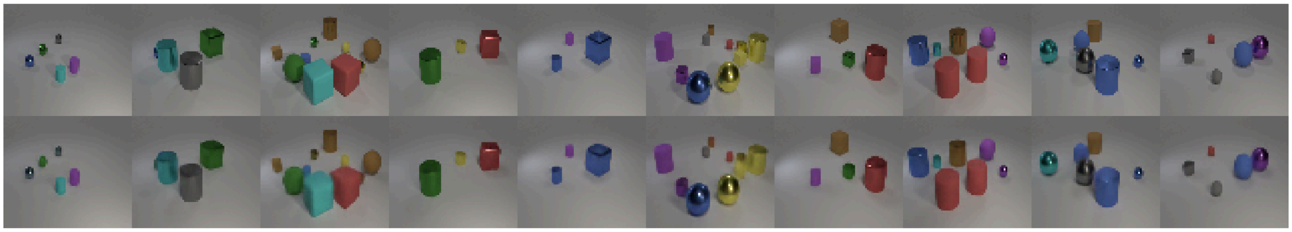


Figure 12: The first row corresponds to the original image. The second row corresponds to the reconstructed images. Evaluated on the test images.

L.2.3. Dataset: CelebA

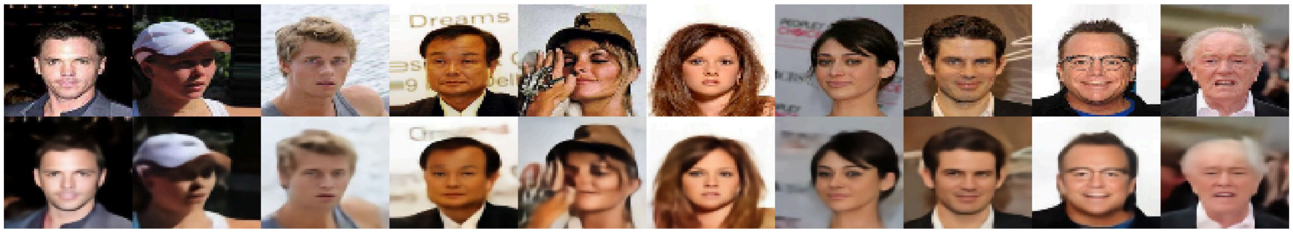


Figure 13: The first row corresponds to the original image. The second row corresponds to the reconstructed images. Evaluated on the test images.

L.2.4. Dataset: FER2013

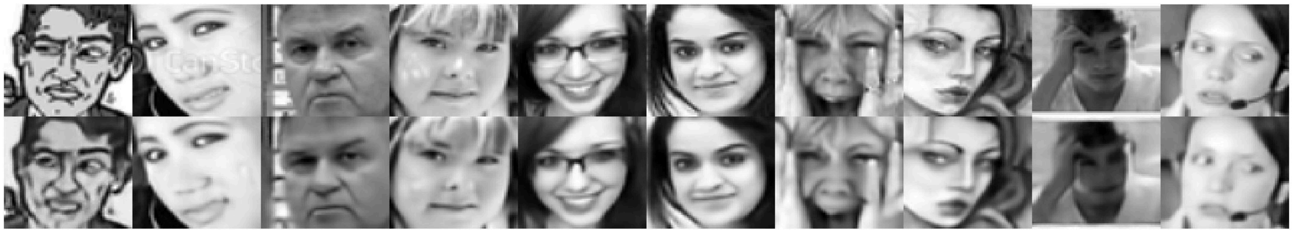


Figure 14: The first row corresponds to the original image. The second row corresponds to the reconstructed images. Evaluated on the test images.

L.3. FSQ-VAE

L.3.1. Dataset: t-ImageNet

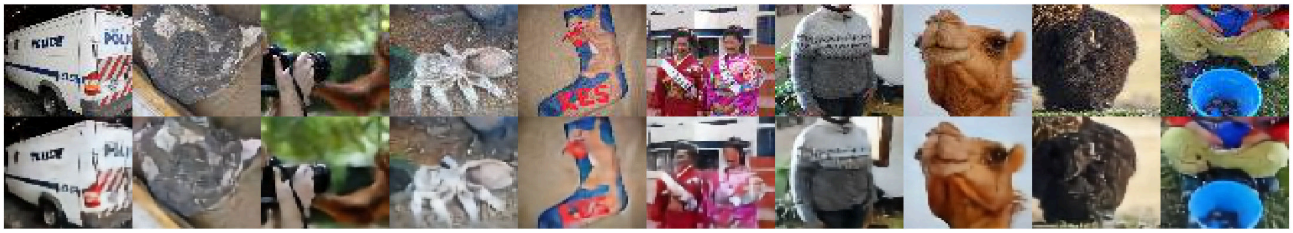


Figure 15: The first row corresponds to the original image. The second row corresponds to the reconstructed images. Evaluated on the test images.

L.3.2. DATASET: CLEVR

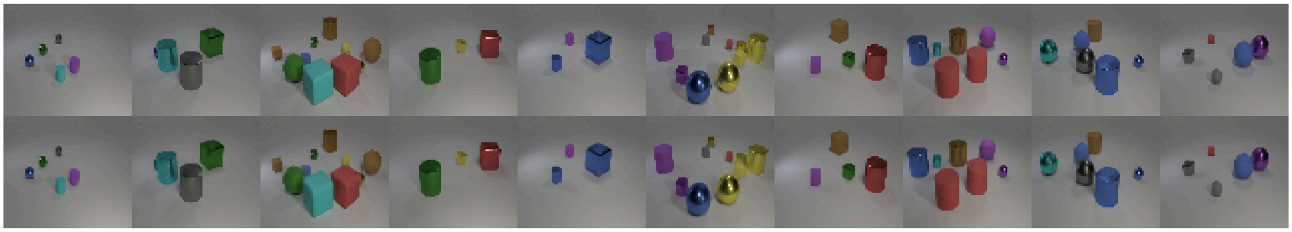


Figure 16: The first row corresponds to the original image. The second row corresponds to the reconstructed images. Evaluated on the test images.

L.3.3. DATASET: CELEBA

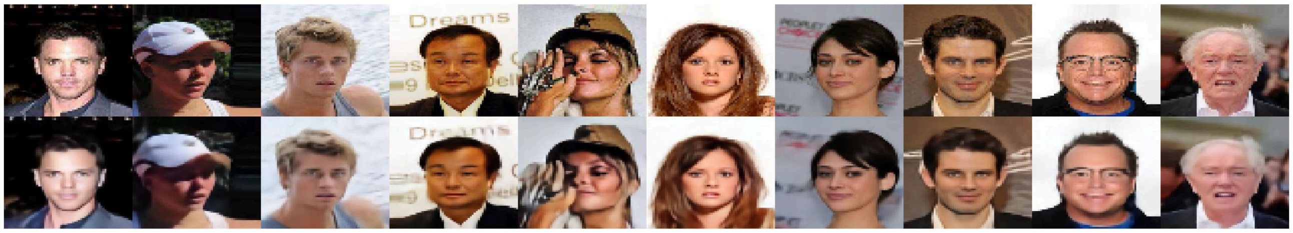


Figure 17: The first row corresponds to the original image. The second row corresponds to the reconstructed images. Evaluated on the test images.

L.3.4. DATASET: FER2013

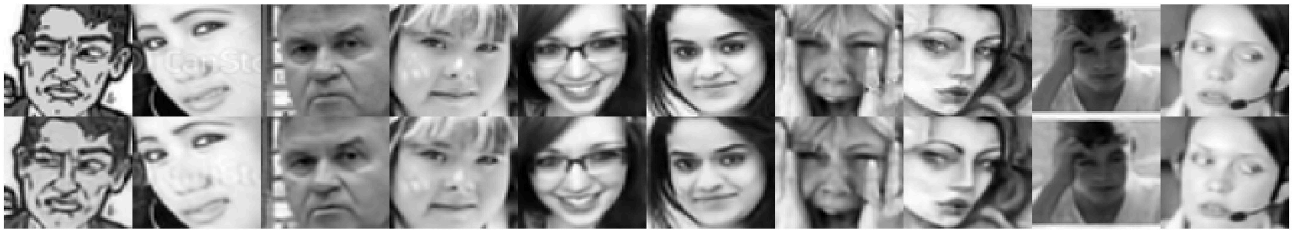


Figure 18: The first row corresponds to the original image. The second row corresponds to the reconstructed images. Evaluated on the test images.

M. Visualisation of Out-of-Distribution Reconstruction

M.1. PPS-VAE

M.1.1. TRAINING DATASET: T-IMAGENET

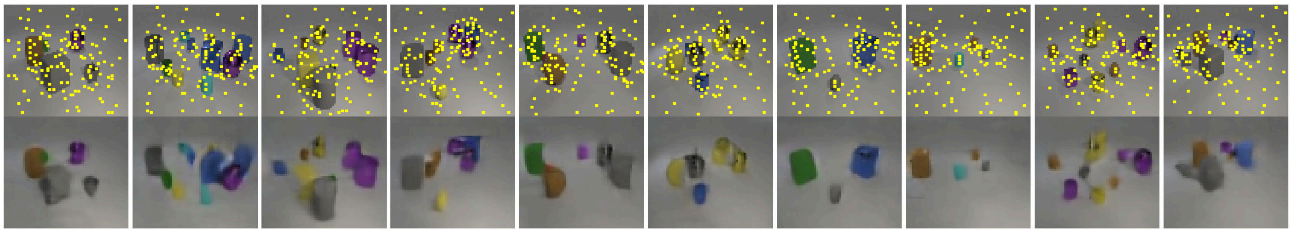


Figure 19: Test dataset CLEVR. Visualisation of the spatial arrangement of points in the context sets. The first row corresponds to the original image, together with the inferred context set denoted by the yellow circles. The second row corresponds to the reconstructed images. The context sets inferred on the test images.

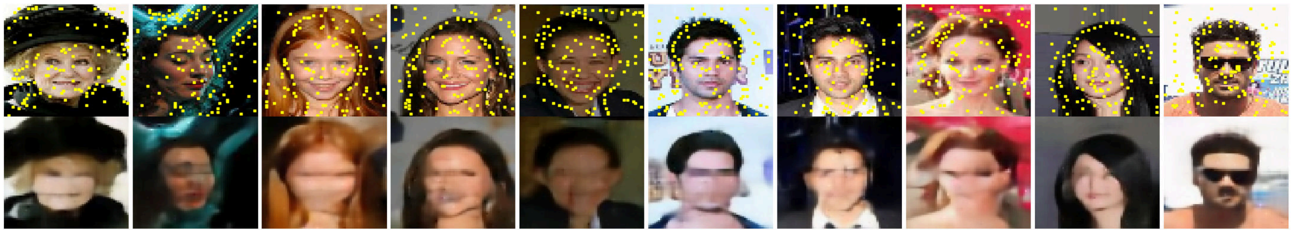


Figure 20: Test dataset CelA. Visualisation of the spatial arrangement of points in the context sets. The first row corresponds to the original image, together with the inferred context set denoted by the yellow circles. The second row corresponds to the reconstructed images. The context sets inferred on the test images.

M.1.2. Training Dataset: CLEVR

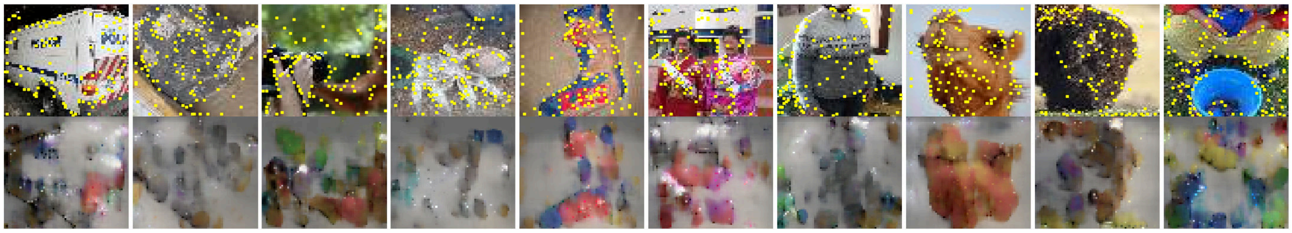


Figure 21: Test dataset t-ImageNet. Visualisation of the spatial arrangement of points in the context sets. The first row corresponds to the original image, together with the inferred context set denoted by the yellow circles. The second row corresponds to the reconstructed images. The context sets inferred on the test images.

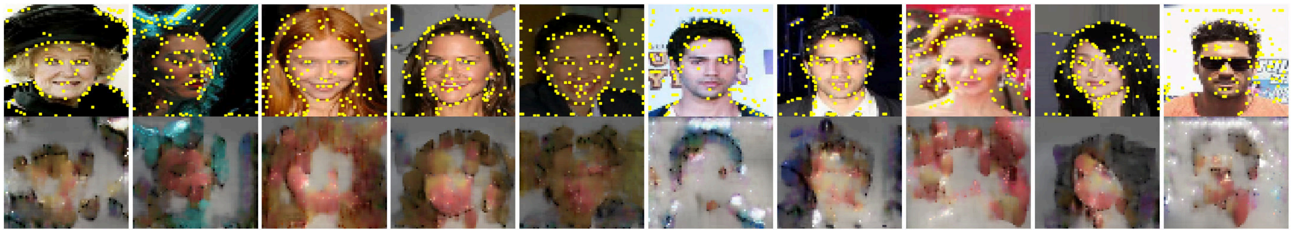


Figure 22: Test dataset CelA. Visualisation of the spatial arrangement of points in the context sets. The first row corresponds to the original image, together with the inferred context set denoted by the yellow circles. The second row corresponds to the reconstructed images. The context sets inferred on the test images.

M.1.3. Training Dataset: CelebA

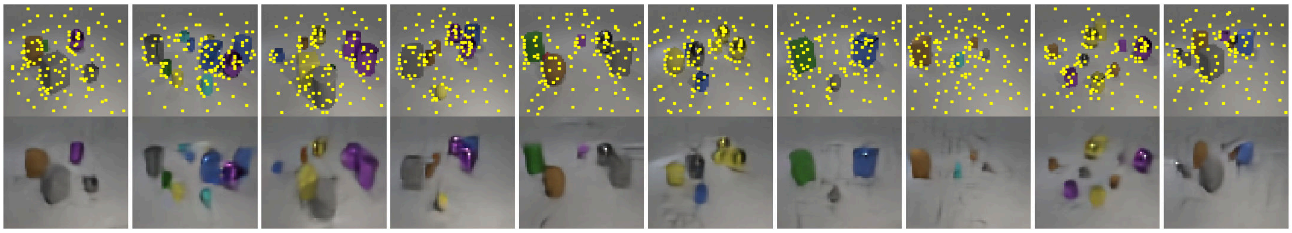


Figure 23: Test dataset CLEVR. Visualisation of the spatial arrangement of points in the context sets. The first row corresponds to the original image, together with the inferred context set denoted by the yellow circles. The second row corresponds to the reconstructed images. The context sets inferred on the test images.

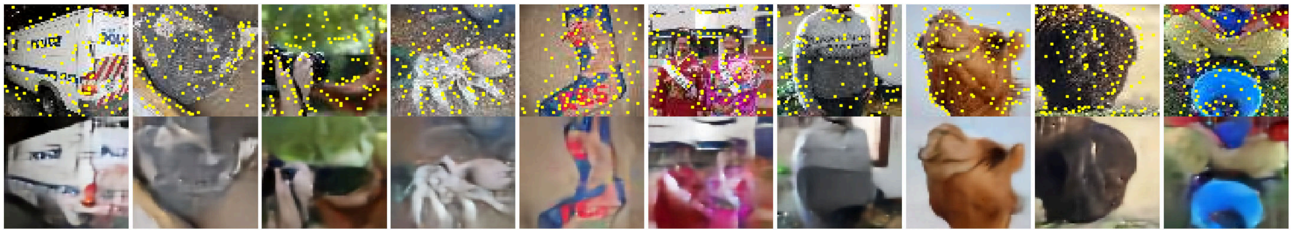


Figure 24: Test dataset t-ImageNet. Visualisation of the spatial arrangement of points in the context sets. The first row corresponds to the original image, together with the inferred context set denoted by the yellow circles. The second row corresponds to the reconstructed images. The context sets inferred on the test images.

M.2. FSQ-VAE

M.2.1. Training Dataset: t-ImageNet

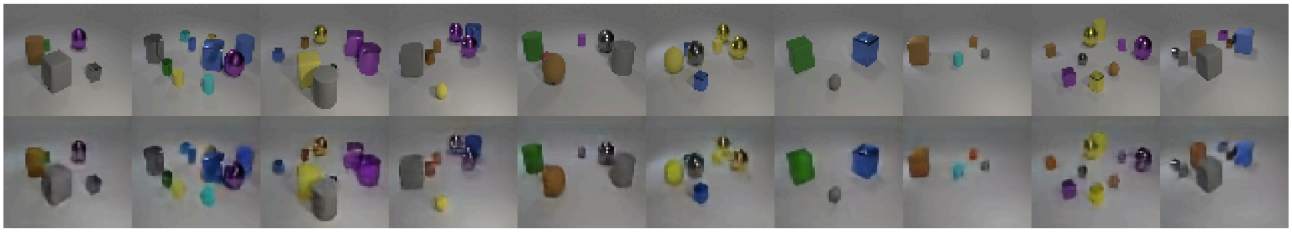


Figure 25: The first row corresponds to the original image. The second row corresponds to the reconstructed images. Evaluated on the CLEVR test images.

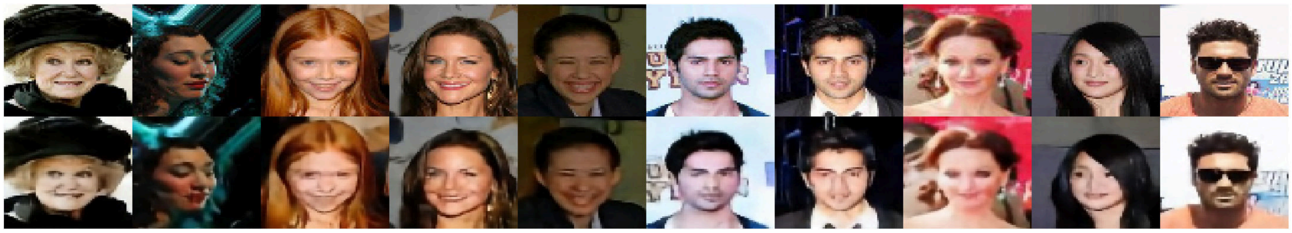


Figure 26: The first row corresponds to the original image. The second row corresponds to the reconstructed images. Evaluated on the CelA test images.

M.2.2. Training Dataset: CLEVR

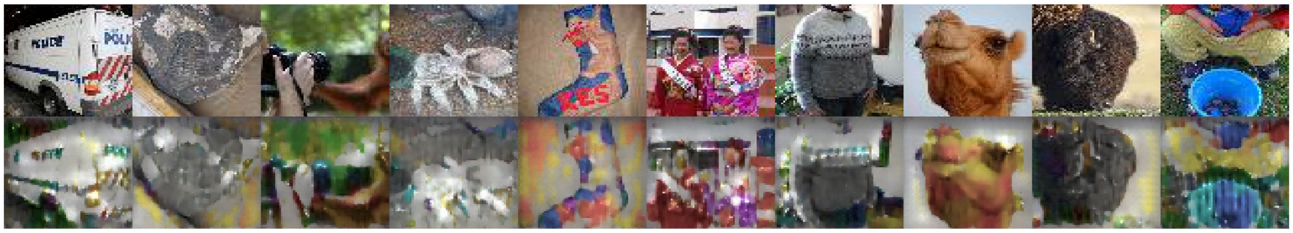


Figure 27: The first row corresponds to the original image. The second row corresponds to the reconstructed images. Evaluated on the t-ImageNet test images.

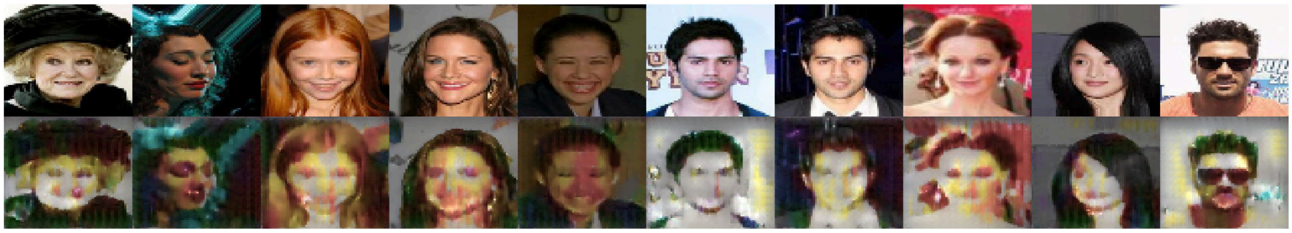


Figure 28: The first row corresponds to the original image. The second row corresponds to the reconstructed images. Evaluated on the CelA test images.

M.2.3. TRAINING DATASET: CELEBA

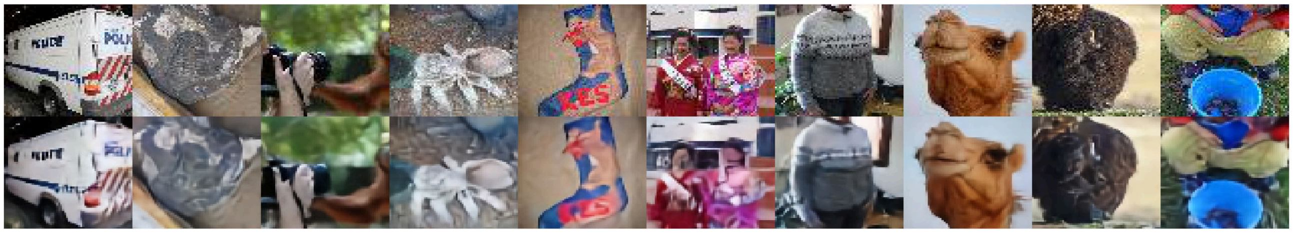


Figure 29: The first row corresponds to the original image. The second row corresponds to the reconstructed images. Evaluated on the t-ImageNet test images.

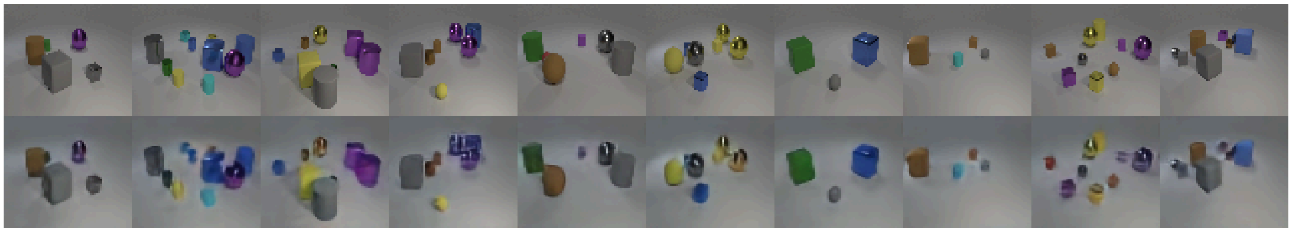


Figure 30: The first row corresponds to the original image. The second row corresponds to the reconstructed images. Evaluated on the CLEVR test images.

M.3. VQ-VAE

M.3.1. TRAINING DATASET: T-IMAGENET

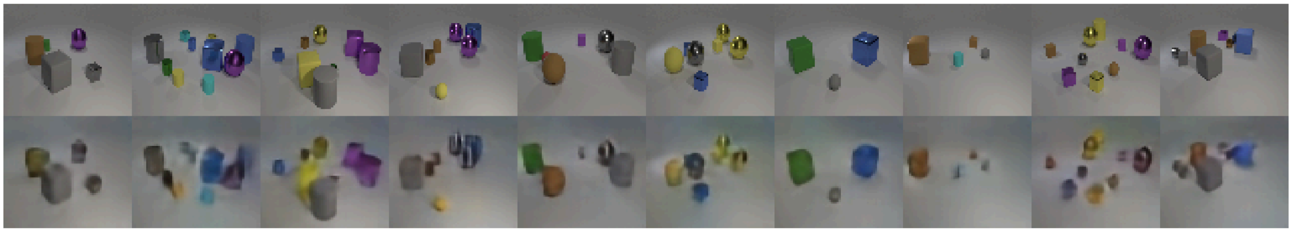


Figure 31: The first row corresponds to the original image. The second row corresponds to the reconstructed images. Evaluated on the CLEVR test images.

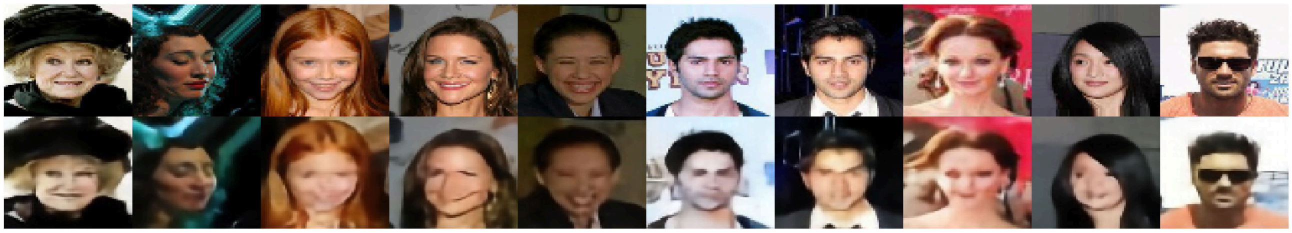


Figure 32: The first row corresponds to the original image. The second row corresponds to the reconstructed images. Evaluated on the CelA test images.

M.3.2. TRAINING DATASET: CLEVR

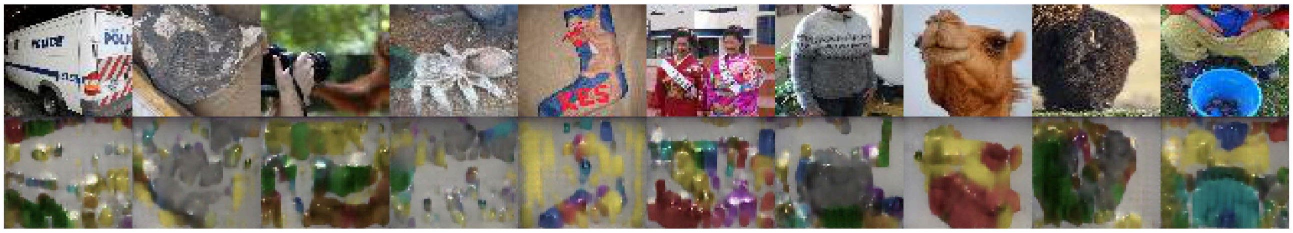


Figure 33: The first row corresponds to the original image. The second row corresponds to the reconstructed images. Evaluated on the t-ImageNet test images.

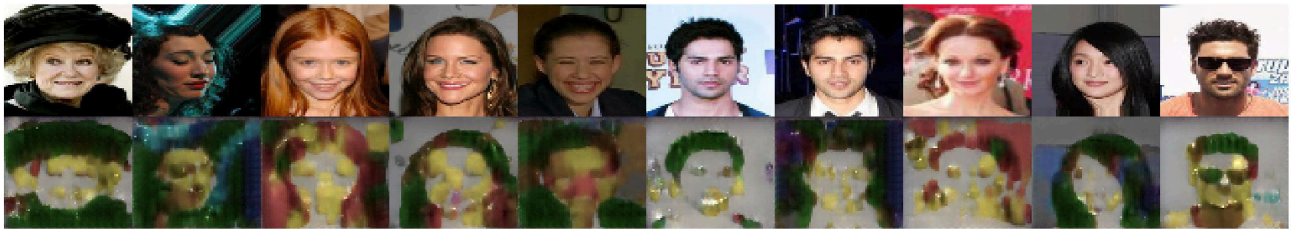


Figure 34: The first row corresponds to the original image. The second row corresponds to the reconstructed images. Evaluated on the CelA test images.

M.3.3. TRAINING DATASET: CELEBA

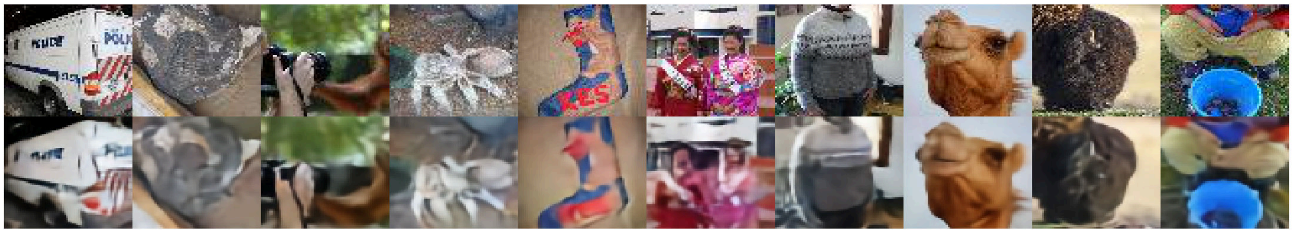


Figure 35: The first row corresponds to the original image. The second row corresponds to the reconstructed images. Evaluated on the t-ImageNet test images.

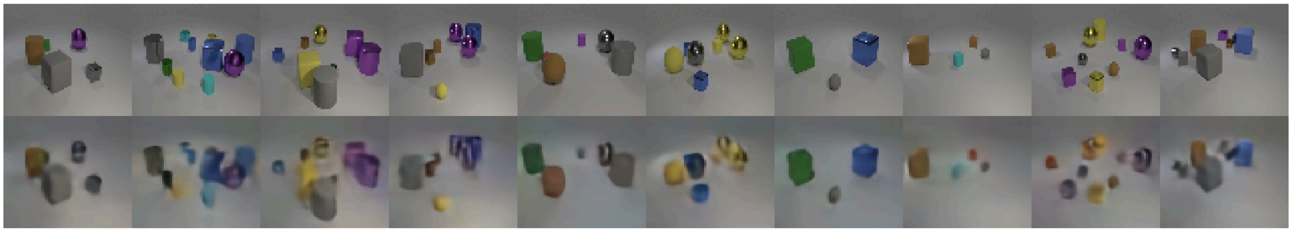


Figure 36: The first row corresponds to the original image. The second row corresponds to the reconstructed images. Evaluated on the CLEVR test images.