Multimodal Controller for Generative Models

Enmao Diao
Department of Electrical and Computer Engineering
Duke University
Durhm, NC 27705
enmao.diao@duke.edu

Jie Ding
Department of Statistics
University of Minnesota-Twin Cities
Minneapolis, MN 55455
dingj@umn.edu

Vahid Tarokh
Department of Electrical and Computer Engineering
Duke University
Durhm, NC 27705
vahid.tarokh@duke.edu

Abstract

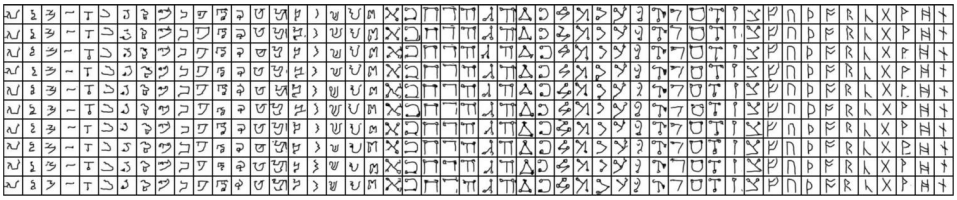
Class-conditional generative models are crucial tools for data generation from user-specified class labels. A number of existing approaches for class-conditional generative models require nontrivial modifications of existing architectures, in order to model conditional information fed into the model. In this paper, we introduce a plug-and-play module called 'multimodal controller' in order to generate multimodal data without introducing additional learning parameters. In the absence of the controllers, our model reduces to non-conditional generative models. We test the efficacy of multimodal controller on CIFAR10 and Omniglot datasets, and experimentally demonstrate that multimodal controlled generative models (including VAE, PixelCNN, Glow, and GAN) are capable of generating class-conditional images of better or comparable quality when compared with the state-of-the-art conditional generative models. Moreover, we show that multimodal controlled models are also capable of transiting images between classes and creating images from novel data modalities.

1 Introduction

In recent years, many generative models based on neural networks have been proposed and achieved remarkable performance. The main backbones of generative models include Autoencoder, Autoregression, Normalization Flow, and Adversarial generative models. Perhaps the most well-known representatives of them are Variational Autoencoder (VAE) [1], PixelCNN [2], Glow [3], and Generative Adversarial Network (GAN) [4], respectively. VAE learns a parametric distribution over an encoded latent space, samples from this distribution, and then constructs generations from decoded samples. PixelCNN uses autoregressive connections to factorize the joint image distribution as a product of conditionals over sub-pixels. Glow optimizes the exact log-likelihood of the data with a deterministic and invertible transformation. GAN is introduced as a generative framework where intractable probabilistic distributions are approximated through adversarial training.

In many application scenarios, we are interested in constructing generations based on a conditional distribution. For instance, we may be interested in generating human face images conditional on some given characteristics of faces such as hair color, eye size, gender, etc. A systematic way to incorporate conditional information may enable us to control the data generating process with more flexibility. In this direction, conditional generative models including Conditional Variational Autoencoder (CVAE) [5], Conditional Generative Adversarial Network (CGAN) [6], and Conditional PixelCNN (CPixelCNN) [7] have been proposed which model conditional information by learning the associated

Preprint. Under review.



(a) Generation

(b) Creation

Figure 1: Snapshots of (a) generated images and (b) created images from the challenging Omniglot dataset [10], using our *multimodal controlled* generative adversarial networks. Each column is a modality-specific generation, attempting to exhibit the same concept [9].

embeddings. The learned features are usually concatenated or added with non-conditional features at various network layers. Conditional Glow (CGlow) learns a class-conditional prior distribution and an optional auxiliary classifier.

In this paper, we propose a plug-and-play module named Multimodal Controller (MC) to control the parameter allocation for training multimodal generative models. Each mode of data will be granted a unique randomly sampled sub-network. The training procedure is identical to non-conditional generative models. Our main contributions of this work are three-fold.

- We provide a novel method to transform non-conditional generative models into class-conditional generative models, by simply attaching a multimodal controller at each layer. Unlike classical conditional generative models, our method does not introduce additional learning parameters, and it can be easily incorporated into existing implementations.
- We demonstrate that the controlled network can allocate specialized sub-networks to each mode of the data in order to generate class-conditional samples. Our method performs comparable or better than various well-known classical conditional generative models, especially when the number of modalities is large, and the data size for each modality is small (as demonstrated in Figure 1(a)).
- We propose a new type of data generation, herein referred to as *data creation*. We show that our method is capable of creating data from novel data modalities (as demonstrated in Figure 1(b)). On the contrary, the existing conditional generative models can only generate data from the modalities predetermined in the training dataset.

We experiment with CIFAR10 and Omniglot datasets [8, 9] at the 32×32 resolution. Our baseline models include CVAE, CPixelCNN, CGlow, and CGAN. The outline of the remaining part of this paper is given next. In Section 2, we review the related work. In Section 3, we introduce our proposed multimodal solution. In Section 4, we provide experimental results demonstrating the performance of our approach. Finally, we make our concluding remarks in Section 5.

2 Related Work

Variational Autoencoder (VAE) [1] is a directed generative model with probabilistic latent variables. Latent variables z are sampled from a latent prior distribution $p(z)$, and the generated data x follow a generative distribution $p(x|z)$. In general, this optimization problem is challenging due to the intractability of computing posterior distributions. However, a variational lower bound in conjunction with stochastic gradient variational Bayes (SGVB) [1] can be used for efficient optimization, based on the re-parameterization trick. The objective function can be written as

$$\min_{q(\cdot|\cdot),p(\cdot|\cdot)} \left\{ D_{\mathrm{KL}} \left(q(z|x) \| p(z) \right) - \mathbb{E}_{q(z|x)} \log p(x|z) \right\}.$$

We note that discrete latent distributions can also be handled by using the Gumbel-softmax trick [11].

Conditional Variational Autoencoder (CVAE) [5] approximates the conditional distribution $p(x \mid h)$ with the conditional information h. The latent variables z allows for the modeling of multi-modality in conditional distributions, so that diverse and perceptually realistic data can be sampled in a supervised manner. In practice, we learn the embedding layers to model the conditional information h, and concatenate or add corresponding conditional features into both the encoder and decoder.

PixelCNN, originally proposed in [2], is an autoregressive generative model that fully factorizes the joint probability density function of images into a product of conditional distributions over all sub-pixels. The conditional distributions $p(x_i|x_{<i})$ can be parameterized by various kinds of neural networks such as recurrent neural networks in PixelRNN [2], convolutional neural networks in PixelCNN [2], residual blocks in CPixelCNN and PixelCNN++ [7, 12], and residual blocks with attention blocks in PixelSNAIL [13]. The autoregressive likehood function is given by

$$p(x) = p\left(x_1, \ldots, x_n\right) = \prod_{i=1}^{n} p\left(x_i | x_1, \ldots, x_{i-1}\right).$$

Conditional PixelCNN learns the embedding layers to model the conditional distribution $p(x|h)$ and adds conditional features to the non-conditional gated convolutional layers [7]. Many applications use PixelCNN as a prior model for the latent space as in VQ-VAE [14, 15]. Autoregressive generative models are also being adopted in generating audio and texts in addition to images [16, 17].

Glow [3] is a normalization flow-based generative model originally proposed in NICE [18] and extended in RealNVP [19]. The flow-based generative model enables exact and tractable log-likelihood and latent space inference. Glow demonstrates that optimizing the plain log-likelihood objective is capable of generating realistic-looking images. By ensuring a bijective neural network mapping, we can directly calculate the likelihood with the change of variables in the following way:

$$\log p(x) = \log p(z) + \textstyle\sum_{i=1}^{k} \log \left|\det\left(J\left(f_i^{-1}(x)\right)\right)\right|, \quad z = f_k^{-1} \circ f_{k-1}^{-1} \circ \ldots f_0^{-1}(x),$$

where $\circ$ denotes the composition operator. Due to the exact latent space inference, the flow-based generative models can synthesize data conditionally by training a prior modeling for the conditional information h with an optional auxiliary classifier. WaveGlow [20] combines the gist of Glow and WaveNet [16] to synthesize high-quality audio. We can also construct autoregressive flow with lower triangular Jacobian by factorizing the joint probability distribution as a product of conditionals [21, 22].

Generative Adversarial Network (GAN) [4] is formed by a Generator (G) network and a Discriminator (D) network. Here, G learns a mapping from a prior distribution $p(z)$ to the data distribution; D outputs a single scalar that estimates the probability of a sample being an authentic training data (instead of a fabricated/generated data from G). The training goal is to find a Nash equilibrium to the following two-player min-max problem:

$$\min_G \max_D \big\{\mathbb{E}_{q(x)}\big[\log D(x)\big] + \mathbb{E}_{p(z)}\big[\log(1 - D(G(z)))\big]\big\}.$$

Conditional Generative Adversarial Network (CGAN) [6] learns the embedding layers to model the conditional information h and attach them to both discriminator and generator. There exist many other ways of incorporating conditional information. In [23], the one-hot class vector is concatenated with the noise vector for G, and the objective is modified to encourage the training of an auxiliary classifier. In [24] and [25], class-conditionals with gains and biases in BatchNorm [26] are provided. In [27], D is conditioned by using the cosine similarity between its features and a learned class-conditional embedding. A hybrid approach that combines the previous two methods was proposed in [28]. Recently, StyleGAN [29] automatically learns unsupervised separation of high-level attributes and enables scale-specific synthesis. MSGAN [30] proposes a mode-seeking regularization term to address the 'mode collapse' issue of CGAN. STGAN [31] tackles image attribute-editing by selectively modeling the difference between target and source attributes.

Evaluating generative models is in general challenging [32]. Throughout our experiments, we use the Inception Score (IS) [33] and Fréchet Inception Distance (FID) [34] in order to approximately measure the quality of generated samples and to compare with the baseline.

3 Proposed Method

3.1 Multimodal Controller

To generate class-conditional generative data, a vanilla approach is to train individual generative models to model the marginal probability distribution for each mode of data, without considering

any inter-modality dependency. In practice, we may not have enough data points for every mode of data, and the computational complexity significantly increases as the number of modes increases. Thus, it is crucial to rely on the association among different data modality to avoid overfitting and reduce computational complexity. To address these issues, Multimodal Controlled (MC) generative models will randomly allocate different sets of model parameters for each mode of data. By allowing parameter sharing and specialization, we are able to generate class-conditional data without introducing additional learning parameters and much computational overhead.

Suppose that there is a dataset X with C data modalities. Each mode of data $X_c = \{x_c^i\}_{i=1}^{N_c}$ consists of N_c i.i.d. samples of a (continuous or discrete) random variable. Given a set of learning model parameters $\theta \in \mathbb{R}^D$ with size D, each mode of data is modeled with a random subset $\theta_c \subset \theta$ without replacement. For notational convenience, here we interchangeably used the notions of subset and subvector. In this way, the allocated parameters for each mode will represent both the inter-mode association and intra-mode variation thanks to parameter sharing and specialization.

Next, we discuss technical details in the specific context of neural networks. Suppose that a data matrix $X \in \mathbb{R}^{N \times K}$ is the input of a network layer, where N is the sample size and K is the input dimension. Suppose that the layer is parameterized by a weight matrix $W \in \mathbb{R}^{D \times K}$ and bias vector $b \in \mathbb{R}^D$, where D is the hidden dimension. We randomly draw C modality codewords $e_c \in \mathbb{F}_2^D$ to construct a modality codebook $E \in \mathbb{F}_2^{C \times D}$, where $\mathbb{F}_2$ denotes the binary field. Note that each row of the codebook is a binary vector that represents a mode. Let $\times$ denote the usual matrix multiplication, and $\odot$ denote the element-wise product. For notational brevity, given a matrix U and a vector u, we let $U \odot v$ denote $U \odot V$ where V is broadcast from u (if the dimension matches). A similar notion applies to the operations of multiplication and addition. Let X_c denote the $N_c \times D$ submatrix of X whose rows correspond to the c-th mode. A multimodal controlled linear layer is given by:

$$\hat{W}_c^{D \times K} = W \odot e_c, \quad \hat{b}_c^D = b \odot e_c, \quad y_c^{N_c \times D} = X_c \times \hat{W}_c^T + \hat{b}_c,$$

where $y_c^{N_c \times D}$ is the output from the c-th mode, and we used subscripts to highlight the dimension.

Our calculation shows that the above formulation requires a time complexity linear in the number of modes. Alternatively, we provide a parallel formulation that is constructed using a Hadamard representation of the above linear transformations. In particular, we may write

$$X^{N \times K} = \begin{bmatrix} X_1^{N_1 \times K} \\ \vdots \\ X_C^{N_C \times K} \end{bmatrix}, \hat{W}^{N \times D \times K} = \begin{bmatrix} \hat{W}_1^{N_1 \times D \times K} \\ \vdots \\ \hat{W}_C^{N_C \times D \times K} \end{bmatrix}, \hat{b}^{N \times D} = \begin{bmatrix} \hat{b}_1^{N_1 \times D}, \\ \vdots \\ \hat{b}_C^{N_C \times D} \end{bmatrix}, y^{N \times D} = \sum_{k=1}^K (X \odot \hat{W}^T)_{ijk} + \hat{b}.$$

The above alternative formulation may be used to compute the multimodal controlled linear layer in parallel in order to reduce the overall time cost. Nevertheless, this parallelization cannot be directly applied to other types of parametric layers such as convolution and transposed convolution layers. Also, it requires N times more memory by creating unnecessary copies of weight matrices.

In practical implementations, we suggest the following *multimodal controlled linear layer*:

$$y^{N \times D} = \phi(\text{BN}(X \times W^T + b)) \odot (H \times E)$$

where $\text{BN}(\cdot)$ denotes the batch normalization [26], $\phi(\cdot)$ is the activation function being used, and $H \in \{0, 1\}^{N \times C}$ is the one-hot indicator of class-conditional information. Our suggestion is mainly based on two considerations. First, H can easily zero-out class-conditional channels of features and make it convenient for parallel computing. Second, we attach MC after activation function to avoid interfering with the calculation of running estimates in batch normalization and an activation function. The above multimodal controlled linear layer can be easily extended to other parametric modules such as convolution and transposed convolution layers.

3.2 Multimodal Controlled Generative Models

Multimodal controlled generative models attach the proposed MC modules to various parametric layers, with some customization depending on their architectures. By controlling the allocation of model parameters, randomly sampled sub-networks are able to synthesize class-conditional data. We describe the usage of MC for the most well-known representatives of generative models including VAE, PixelCNN, Glow, and GAN.

Multimodal Controlled Variational Autoencoder (MCVAE). MCVAE appends an MC module to every layer of VAE. The number of model parameters remains the same as non-conditional VAE. If $E = \mathbf{1}$, the all-one vector, our proposed method produces a non-conditional VAE. For each mode of data x_c, we allocate a unique subset of learning parameters θ_c randomly sampled from the overall model parameters θ, and obtain class-conditional latent variables z_c. The objective can be written as:

$$\min_{\theta} \left\{ \sum_{c=1}^{C} \frac{N_c}{N} \left[\mathbb{E}_{q_{\theta_c}(z_c|x_c)} \log p_{\theta_c}(x_c|z_c) - D_{\mathrm{KL}} \left(q_{\theta_c}(z_c|x_c) \| p_{\theta_c}(z_c) \right) \right] \right\}.$$

Thanks to the end-to-end structure of VAE, the proposed MCVAE is able to class-conditionally zero out the channels of features layer by layer. For the bottleneck layer, we obtain class-conditional latent variables z_c by attaching the MC module after applying the reparameterization trick, instead of directly attaching to the parameters of the latent distribution. The model parameters θ_c can be optimized in the same way as in non-conditional VAE.

Multimodal Controlled PixelCNN (MCPixelCNN). PixelCNN combines two gated convolutional network stacks in order to remove blind spots in the receptive field [7]. MCPixelCNN allocates sub-networks for each mode by introducing the following multimodal controlled gated activation unit:

$$y = \left\{ \tanh \left(W_{k,f} * x \right) \odot \left(H \times E \right) \right\} \odot \sigma \left(W_{k,g} * x \right),$$

where σ denotes the sigmoid function, k is the number of layers, and $*$ is the convolution operator. Our various experimental results show that controlling the $\tanh(\cdot)$ activation unit is sufficient for synthesizing class-conditional data. Other convolution layers used in connecting two stacks and residuals do not necessarily need to be multimodally controlled. The class-conditional likelihood function can be written as

$$p_{\theta_c}(x_c) = p_{\theta_c}\left(x_{1,c}, \ldots, x_{n,c}\right) = \textstyle\prod_{i=1}^{n} p_{\theta_c}\left(x_{i,c} | x_{1,c}, \ldots, x_{i-1,c}\right).$$

Multimodal Controlled Glow (MCGlow). Glow has three major components including Actnorm, Invertible 1×1 convolution, and affine coupling layer. The number of channels of features passing Actnorm and Invertible 1×1 convolution is usually a multiple of the number of color channels due to the squeeze operation. It leads to a small number of channels of features and is incapable of allocating parameters for a large number of data modalities. MCGlow allocates sub-networks only for $\mathrm{NN}(\cdot)$ in affine coupling layer, where $\mathrm{NN}(\cdot)$ is a shallow and wide convolutional neural network block often used in ResNet [35] and RealNVP [19]. As a result, the parameters used in Actnorm and Invertible 1×1 convolution are shared across all modes. The class-conditional likelihood function for a k-layer Glow can be written as:

$$\log p_{\theta_c}(x_c) = \log p(z) + \textstyle\sum_{i=1}^{k} \log \left| \det \left(J \left(f_{i,c}^{-1}(x_c) \right) \right) \right|, \quad z = f_{k,c}^{-1} \circ f_{k-1,c}^{-1} \circ \ldots f_{0,c}^{-1}(x_c).$$

Multimodal Controlled Generative Adversarial Network (MCGAN). MCGAN appends MC module to every layer of the discriminator and generator. We experimentally found that using MC on either generator or discriminator only does not produce class-conditional synthesis. We do not mask out the latent variable z because there exist no preceding parametric layers as in VAE. The objective function of a multimodal controlled two-player minimax game is formulated as:

$$\min_G \max_D \left\{ \textstyle\sum_{c=1}^{C} \frac{N_c}{N} \left[\mathbb{E}_{q_{\theta_c}(x_c)} \left[\log D_{\theta_c}(x_c) \right] + \mathbb{E}_{p_{\theta_c}(z)} \left[\log(1 - D_{\theta_c}(G_{\theta_c}(z))) \right] \right] \right\}.$$

4 Experiments

In this section, we demonstrate applications of our proposed generative models to data generation, transition, and creation. We compare the result of our proposed multimodal controlled generative models with that of classical conditional generative models. We illustrate our results for four different types of multimodal controlled generative models on CIFAR10 and Omniglot datasets [8, 9]. In particular, we use VQ-VAE to reduce computational complexity and use PixelCNN to learn an autoregressive prior distribution for generating the latent space. We do not train an auxiliary classifier for Glow networks, as was suggested in the original paper for fair comparisons with other generative models. Due to the random nature of our proposed method, we conduct 12 random experiments for each generative model on each task. The results show that random sub-network sampling is robust

enough to produce stable results for either a small or large number of data modality. We also note that it is always possible for conditional generative models to introduce more learning parameters to model conditional information and produce better results. Details regarding the experimental settings, network architecture, and model complexity for comparisons are described in the supplementary document.

4.1 Generation

In this section, we present quantitative and qualitative results of conditional and multimodal controlled generative models. From our results, multimodal controlled generative models are able to provide samples of comparable or better fidelity and diversity especially for the Omniglot dataset which has a challenging number of data modality.

We show the means and standard errors (from 12 random experiments) of the Mean Squared Error (MSE) for VAE and Negative Log-likelihood (NLL) for PixelCNN and Glow in Table 1 and 2. Theses quantitative results show our proposed method is able to produce similar reconstructions as conditional generative models. We note that these NLL values are only comparable between conditional and multimodal controlled generative models that use the same pre-trained VQ-VAE encoder and decoder. To further assess the fidelity and diversity of our generated samples, we report our results in Table 3 with Inception Score (IS) [33] and Fréchet Inception Distance (FID) [34] which are perhaps the two most common metrics for comparing generative models.

Our quantitative results show that multimodal controlled generative models perform comparably with conditional generative models for CIFAR10, which has sufficient shots for a small number of data modality. On the other hand, our proposed method performs considerably better than the baseline for Omniglot, which has inadequate shots of a large number of data modality. It is worth mentioning that the MCPixleCNN trained on Omniglot performs worse than the CPixelCNN because the number of learning parameters of CPixelCNN is twice as large as MCPixelCNN due to the size of conditional embeddings. Note that randomly sampled sub-networks are using approximately half the number of learning parameters as the original network. A similar observation is made on MCGAN, which has a higher discriminator loss and a lower IS in training CIFAR10 because of a weaker discriminator in adversarial training. On the contrary, when trained with a dataset that is less difficult to discriminate (e.g. Omniglot), MCGAN is able to generate class-conditional data with high fidelity and diversity. Random experiments show that randomly sampled sub-networks indeed produce less stable but comparable or better results when compared with conditional generative models. We provide qualitative results in Figure 1(a) and 2. More generations can be found in the supplementary document. Both our quantitative and qualitative results demonstrate the efficacy of the proposed MC.

	CIFAR10	Omiglot
CVAE	0.032 (3.3E-5)	0.015 (4.8E-5)
MCVAE	0.032 (5.5E-5)	0.022 (1.2E-3)
VQ-VAE	0.012 (1.2E-4)	0.002 (5.9E-5)

Table 1: Mean Squared Errors of various VAEs.

	CIFAR10	Omiglot
CPixelCNN	2.53 (0.047)	0.30 (0.021)
MCPixelCNN	2.38 (0.044)	0.35 (0.027)
CGlow	3.39 (0.001)	0.97 (0.012)
MCGlow	3.42 (0.002)	1.03 (0.025)

Table 2: Negative Log-Likelihood (NLL) of multi-modal controlled and conditional PixelCNN/Glow.

4.2 Transition

In this section, we provide qualitative results of data transition in Figure 3. More results are included in the supplementary document. For a given fixed latent vector, VAE, Glow, and GAN are able to transit one generated data from one data modality to another. Conditional generative models achieve this task by interpolating between conditional embeddings. Inspired by genetic methods, multimodal controlled generative models transit data with a step rate α. In particular, we use the source modality codeword e_s and the target modality codeword e_t to make an intermediate modality codeword e_i by $e_i = [e_{t,0:\lfloor \alpha D \rceil}, e_{s,\lfloor \alpha D \rceil:D}]$.

4.3 Creation

Data creation means that a generative model is able to class-conditionally synthesize from a novel data modality that is not even prescribed in the training dataset. As shown in the supplementary

	CIFAR10		Omniglot	
	IS	FID	IS	FID
CVAE	3.4 (0.01)	129.2 (0.42)	163.1 (0.74)	370.6 (1.86)
MCVAE	3.2 (0.01)	132.2 (0.36)	555.1 (23.94)	354.8 (22.56)
CPixelCNN	4.6 (0.01)	76.4 (0.22)	463.7 (6.73)	27.8 (0.87)
MCPixelCNN	4.4 (0.02)	81.1 (0.34)	394.0 (6.62)	46.2 (1.74)
CGlow	4.3 (0.02)	68.1 (0.43)	312.2 (2.20)	36.7 (0.68)
MCGlow	4.7 (0.02)	68.1 (0.34)	499.1 (26.40)	43.2 (1.66)
CGAN	8.1 (0.03)	16.2 (0.15)	317.4 (0.99)	41.0 (0.68)
MCGAN	7.2 (0.04)	28.6 (0.66)	916.2 (9.80)	23.6 (0.76)

Table 3: Inception Score (IS) and Fréchet Inception Distance (FID) for generative models trained with CIFAR10 and Omniglot. Means and standardard errors (in brackets) are calculated across 12 random experiments.

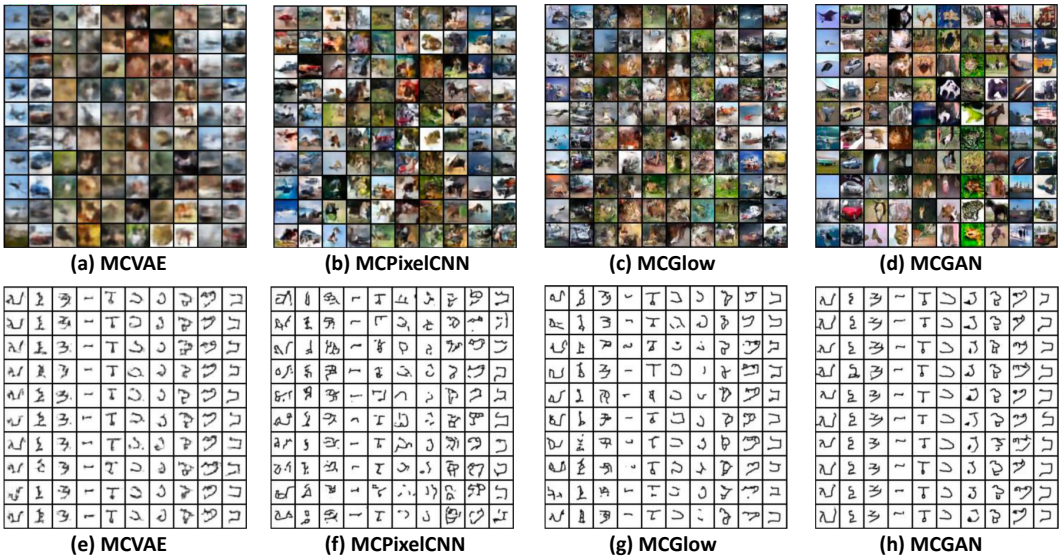


Figure 2: Generations from multimodal controlled generative models trained with CIFAR10 (the first row) and Omniglot (the second row). Each column are generations from one data mode. Generations from conditional generative models are included in the supplementary material.

document, a convex combination of conditional embeddings is not able to create novel data. We propose to create a new data modality by randomly sampling new modality codewords e_n. This method is a natural consequence of the proposed MC architecture, and it does not rely on learning parameters. To our best knowledge, the task of data creation is novel and does not have a standard benchmark. For now, We evaluate the quality of clustering quantitatively with Davies-Bouldin Index (DBI) [36] as shown in Table 4. It needs to be pointed out that this metric does not draw a connection between the created data and the original training data. We provide qualitative results of data creation as shown in Figure 1(b) and 4, and more results in the supplementary document. The results show the new modality codewords e_n is capable of creating novel data from a data modality that has never been trained before. Creations trained from CIFAR10, although clustering consistently, are not structurally similar to the original training data. A possible reason is that a small number of data modality is insufficient to learn adequate variations of sub-networks. On the contrary, the MCGAN trained with Omniglot is able to create novel data with high fidelity and diversity mainly because the variations of sub-networks have been sufficiently exploited.

5 Conclusions

In this work, we propose Multimodal Controller (MC) for generative models for the generation of class-conditional data. Unlike classical conditional generative models which introduce additional learning parameters in order to model class-conditional information, our method takes the advantage of parameter sharing and specialization from variations of sub-networks. Our method is also generic

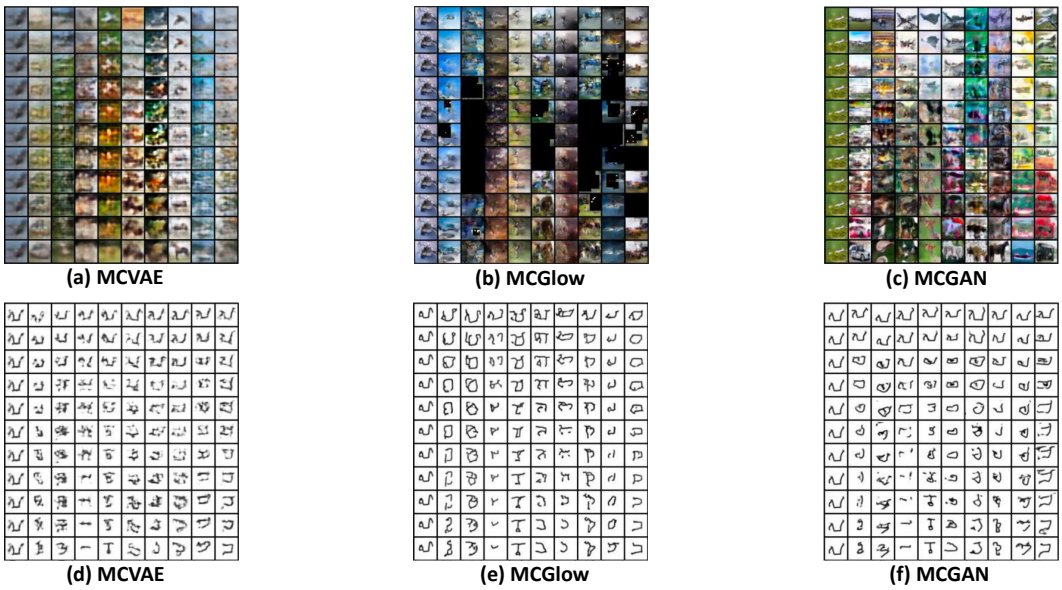


Figure 3: Transitions from multimodal controlled generative models trained with CIFAR10 and Omniglot. Each column has transitions starting from the first mode to its corresponding index. The most left column thus has no transitions.

	CIFAR10	Omniglot
Raw dataset	11.20	5.38
MCVAE	1.9 (0.04)	2.5 (0.05)
MCPixelCNN	4.6 (0.24)	4.3 (0.05)
MCGlow	4.5 (0.15)	5.0 (0.17)
MCGAN	3.5 (0.11)	3.9 (0.05)

Table 4: Davies-Bouldin Index (DBI) for multimodal controlled generative models on the raw and created datasets of CIFAR10 and Omniglot. The created dataset has the same number of modalities as the raw dataset. Small DBI values indicate that our creations are properly clustered on novel data modalities. We note that the quantity of DBI does not fully characterize the 'goodness' of creation, and other possible evaluation criteria are left as future work.

for various well-known generative models, with performance comparable or better than classical conditional generative models. Moreover, a novel machine learning task is proposed and coined as 'data creation'. Multimodal controlled generative models are naturally capable of creating data from novel data modalities by leveraging massive combinations of inherent sub-networks. We believe that this work will shed light on machine learning creativity with a natural and simple usage of neural networks.

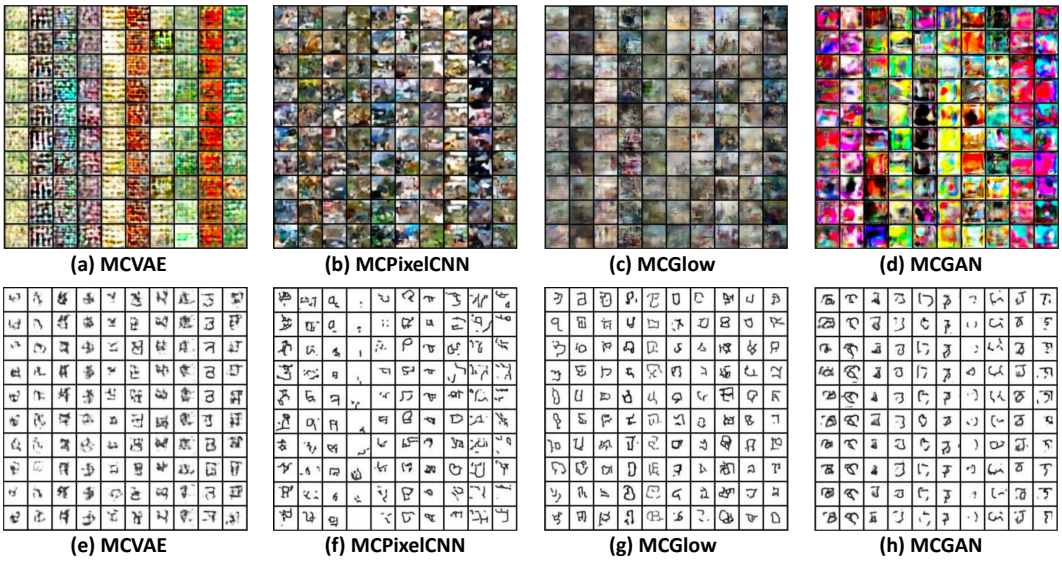


Figure 4: Creations from multimodal controlled generative models trained with CIFAR10 and Omniglot. Each column are creations from one data mode.

Broad Impact

Our new method for creating generative models may be used to facilitate the design and deployment of sophisticated generative models. For the social aspects, a potential positive outcome is the significant reduction of R&D costs in many fields that will benefit from this technique. A potential negative societal consequence is that it may facilitate the auto-generation of unwanted content on a massive scale.

Acknowledgement

This work was supported by the Office of Naval Research Grant No. N00014-18-1-2244.

References

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

[2] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.

[3] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pages 10215–10224, 2018.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

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

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

[7] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in neural information processing systems*, pages 4790–4798, 2016.

[8] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.

[9] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[10] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. The omniglot challenge: a 3-year progress report. *Curr. Opin. Behav. Sci.*, 29:97–104, 2019.

[11] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[12] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017.

[13] Xi Chen, Nikhil Mishra, Mostafa Rohaninejad, and Pieter Abbeel. Pixelsnail: An improved autoregressive generative model. *arXiv preprint arXiv:1712.09763*, 2017.

[14] Aaron van den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, pages 6306–6315, 2017.

[15] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *Advances in Neural Information Processing Systems*, pages 14837–14847, 2019.

[16] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[17] Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. Neural machine translation in linear time. *arXiv preprint arXiv:1610.10099*, 2016.

[18] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

[19] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[20] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3621. IEEE, 2019.

[21] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in neural information processing systems*, pages 4743–4751, 2016.

[22] Chin-Wei Huang, David Krueger, Alexandre Lacoste, and Aaron Courville. Neural autoregressive flows. *arXiv preprint arXiv:1804.00779*, 2018.

[23] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2642–2651. JMLR. org, 2017.

[24] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629*, 2016.

[25] Harm De Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C Courville. Modulating early visual processing by language. In *Advances in Neural Information Processing Systems*, pages 6594–6604, 2017.

[26] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[27] Takeru Miyato and Masanori Koyama. cgans with projection discriminator. *arXiv preprint arXiv:1802.05637*, 2018.

[28] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

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

[30] Qi Mao, Hsin-Ying Lee, Hung-Yu Tseng, Siwei Ma, and Ming-Hsuan Yang. Mode seeking generative adversarial networks for diverse image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1429–1437, 2019.

[31] Ming Liu, Yukang Ding, Min Xia, Xiao Liu, Errui Ding, Wangmeng Zuo, and Shilei Wen. Stgan: A unified selective transfer network for arbitrary image attribute editing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3673–3682, 2019.

[32] Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2015.

[33] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

[34] 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. In *Advances in neural information processing systems*, pages 6626–6637, 2017.

[35] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[36] David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227, 1979.

[37] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[38] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

Supplementary Material for "Multimodal Controller for Generative Models"

Abstract of the supplement. In this supplementary document, we provide 1) more experimental details including the metrics and network architecture, and 2) more qualitative results to demonstrate the multimodal controlled generative models presented in the main paper as well as conditional generative models. Our codes and comments to reproduce the presented results have been uploaded as part of the supplementary material.

6 Experimental Details

6.1 Metric

Inception Score (IS) was originally introduced by [33] and formulated as $\exp\left(\mathbb{E}\left(D_{\mathrm{KL}}(p(y|x)\|p(y))\right)\right)$, where $p(y|x)$ is the output of pretrained Inception network [37]. [37] shows that IS is strongly correlated with subjective human views of image quality. Fréchet Inception Distance (FID) [34] uses the information of the final layer of the inception model to measure the quality of the generated examples. It measures the Wasserstein distance between two distributions p_1 and p_2 assuming that they are both multivariate Gaussian distributions, expressed by

$$\|\mu_{p_1} - \mu_{p_2}\|_2^2 + \mathrm{trace}\left(C_{p_1} + C_{p_2} - 2\left(C_{p_1}C_{p_2}\right)^{1/2}\right),$$

where (μ_{p_1}, μ_{p_2}) and (C_{p_1}, C_{p_2}) are the mean and covariance of samples from p_1 and p_2 respectively.

We evaluate IS for CIFAR10 with the standard Inception network pretrained with the ImageNet dataset. We train our own classifier as shown in Table 10 for Omniglot dataset with all the training and test data. We use both training and test data from Omniglot throughout our experiments, and as a result each generative model is trained with 20 images across 1623 modes. We calculate the score for randomly generated 1000 examples per mode for CIFAR10, and 20 per mode for Omniglot. For each experiment we split our generated samples into 10 parts and calculate the average of IS. We repeat each experiment 12 times with different random seeds and report the means and standard errors as in Table 3.

6.2 Network Architecture

We show hyperparameters for training generative models in Table 5. We use exponential moving average to update the vector quantizer in VQ-VAE with a dictionary size of 512 and embedding dimension of 64 [14]. We show multimodal controlled layers in Figure 4. The network architectures of MCVAE and MCGAN used in our experiments are shown in Table 6 to 9, where n_c is the number of image channels and M is the image shape. We use the standard architecture of PixelCNN and Glow described in the original work [7, 3]. The number of layers, embedding size, and hidden channel size of PixelCNN are 15, 512, and 128, respectively. The depth of a flow K, the number of levels L, and the hidden channel size of Glow are 3, 16, and 512, respectively. We show that the number of learning parameters of generative models in Table 11. The embedding size of embeddings used for modeling conditional information h in CVAE and CGAN is 32. We concatenate conditional embeddings to the first layer of encoder (discriminator) and decoder (generator) as described in [5, 6]. Following the implementation suggested in [7], we add conditional embeddings to every gated activation in CPixelCNN. As a result, CPixelCNN has more learning parameters for modeling h than other conditional generative models do. Moreover, the CPixelCNN for Omniglot also has twice the number of learning parameters as MCPixelCNN does. Therefore, it is foreseeable that the CPixelCNN for Omniglot outperforms the MCPixelCNN as shown in 3. We conditional embeddings to the prior trained in the first level of CGlow [3]. Multimodal controlled generative models always have less learning parameters than conditional generative models because we do not train any conditional embeddings to model h.

7 More Qualitative Results

In this section, we provide more qualitative results that are demonstrated in the main paper. We show the qualitative results from conditional generative models and results with more modes. We show

	VAE	VQ-VAE	PixelCNN	Glow	GAN
Batch size	128	128	128	128	64
Loss	BCE	MSE	NLL	NLL	Hinge
Optimizer	Adam	Adam	Adam	Adam	Adam
α	2E-4	2E-4	2E-4	2E-4	2E-4
(β_1, β_2)	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)	(0.5, 0.999)
Scheduler	Exponetial	Exponetial	Exponetial	Exponetial	None
γ	0.99	0.99	0.99	0.99	None

Table 5: Hyperparameters for training generative models.

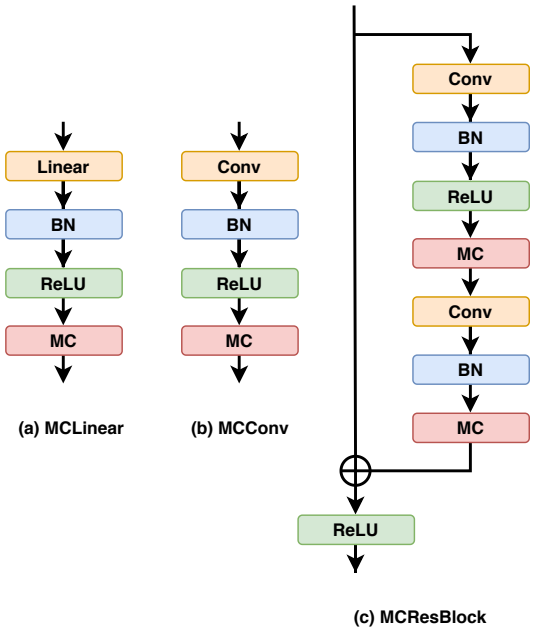


Figure 5: Multimodal Controlled Linear (MCLinear), Convolution (MCConv) and ResBlock (MCResBlock) architecture used in all our experiments.

Image $x \in \mathbb{R}^{M \times M \times n_c}$
MCConv (n_c, 64, 4, 2, 1)
MCConv (64, 128, 4, 2, 1)
MCConv (128, 256, 4, 2, 1)
MCResBlock (256, 256, 3, 1, 1)
MCResBlock (256, 256, 3, 1, 1)
Linear (μ) ($256 \times M_e \times M_e$, 128)
Linear ($\log \sigma^2$) ($256 \times M_e \times M_e$, 128)

Table 6: Encoder used in MCVAE. $M_e = 4$ for CIFAR10 and Omniglot.

$z \in \mathbb{R}^{128} \sim \mathcal{N}(0, I)$
MC (128, 128)
Linear (128, $256 \times M_e \times M_e$), BN, ReLU
MC (256, 256)
MCResBlock (256, 256, 3, 1, 1)
MCResBlock (256, 256, 3, 1, 1)
MCConvTranspose (256, 128, 4, 2, 1)
MCConvTranspose (128, 64, 4, 2, 1)
ConvTranspose (64, n_c, 4, 2, 1), Sigmoid

Table 7: Decoder used in MCVAE. $M_e = 4$ for CIFAR10 and Omniglot.

$z \in \mathbb{R}^{128} \sim \mathcal{N}(0, I)$
MCConvTranspose (128, 512, 4, 1, 0)
MCResBlock Up (512, 256, 3, 1, 1)
MCResBlock Up (256, 128, 3, 1, 1)
MCResBlock Up (128, 64, 3, 1, 1)
Conv (64, n_c, 3, 1, 1), Tanh

Table 8: Generator used in MCGAN. ResBlock Up interpolates input feature map at the beginning of block by scale 2.

Image $x \in \mathbb{R}^{M \times M \times n_c}$
MCResBlock Down (n_c, 64, 3, 1, 1)
MCResBlock Down (64, 128, 3, 1, 1)
MCResBlock Down (128, 256, 3, 1, 1)
MCResBlock (256, 512, 3, 1, 1)
Global Average Pooling
Linear (512, 1)

Table 9: Discriminator used MCGAN. All BN layers are replaced with Spectral Normalization (SN) [38]. ResBlock Down average pools feature map at the end of block by scale 2.

Image $x \in \mathbb{R}^{M \times M \times n_c}$
Conv (n_c, 8, 3, 1, 1)
MaxPool (8, 8, 2)
Conv (8, 16, 3, 1, 1)
MaxPool (16, 16, 2)
Conv (16, 32, 3, 1, 1)
MaxPool (32, 32, 2)
Conv (32, 64, 3, 1, 1)
Linear ($64 \times M_e \times M_e$, C)

Table 10: Classifier used to train Omniglot for evaluating IS and FID. $M_e = 4$ for CIFAR10 and Omniglot.

	CIFAR10	Omiglot
CVAE	7.79	7.89
MCVAE	7.63	7.62
CPixelCNN	6.40	12.60
MCPixelCNN	6.36	6.36
CGlow	21.97	15.79
MCglow	21.97	15.73
CGAN	8.69	8.79
MCGAN	8.41	8.41

Table 11: Number of learning parameters used in generative models in M (millions)

additional generations and transitions from conditional and multimodal controlled generative models in Figure 6 to 10. The CGAN trained with Omniglot suffers from mode collapse and fails to generate and transit data. We transit data in conditional generative models by fixing z and interpolating between two conditional embedding vectors. Note that the original paper of CGlow proposed to interpolate the sampling space instead of conditional embeddings thanks to its invertibility. We show the qualitative results of creations in Figure 11 and 12. To our best knowledge, the way to create novel conditional embeddings is not well studied. Our result shows that random convex combination of trained conditional embeddings does not produce novel modes. On the contrary, *by random sampling a new set of codewords*, multimodal controlled generative models are able to create data clustered on unforeseen centers.

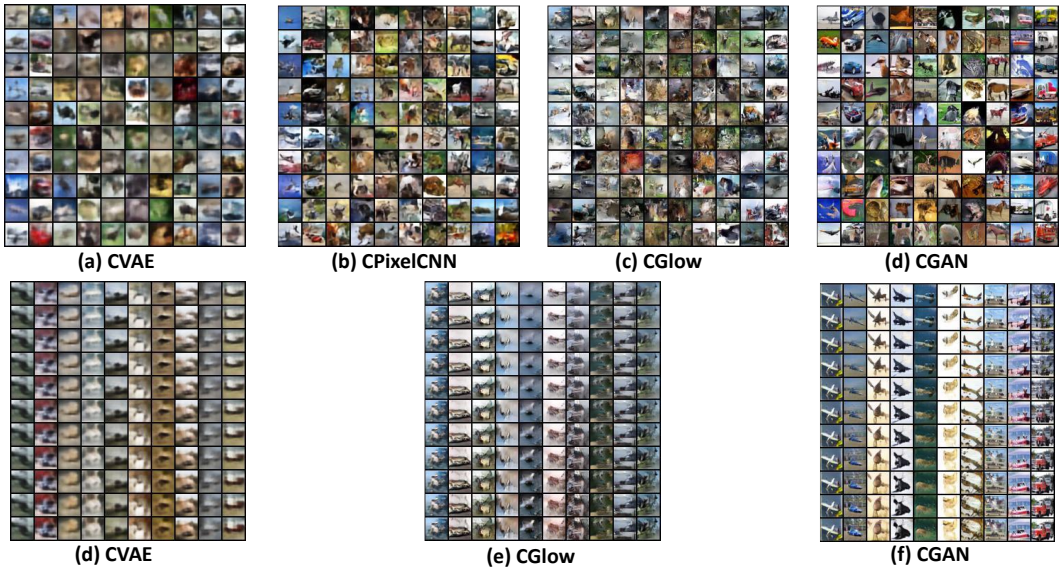


Figure 6: Generations and Transitions from conditional generative models trained with CIFAR10.

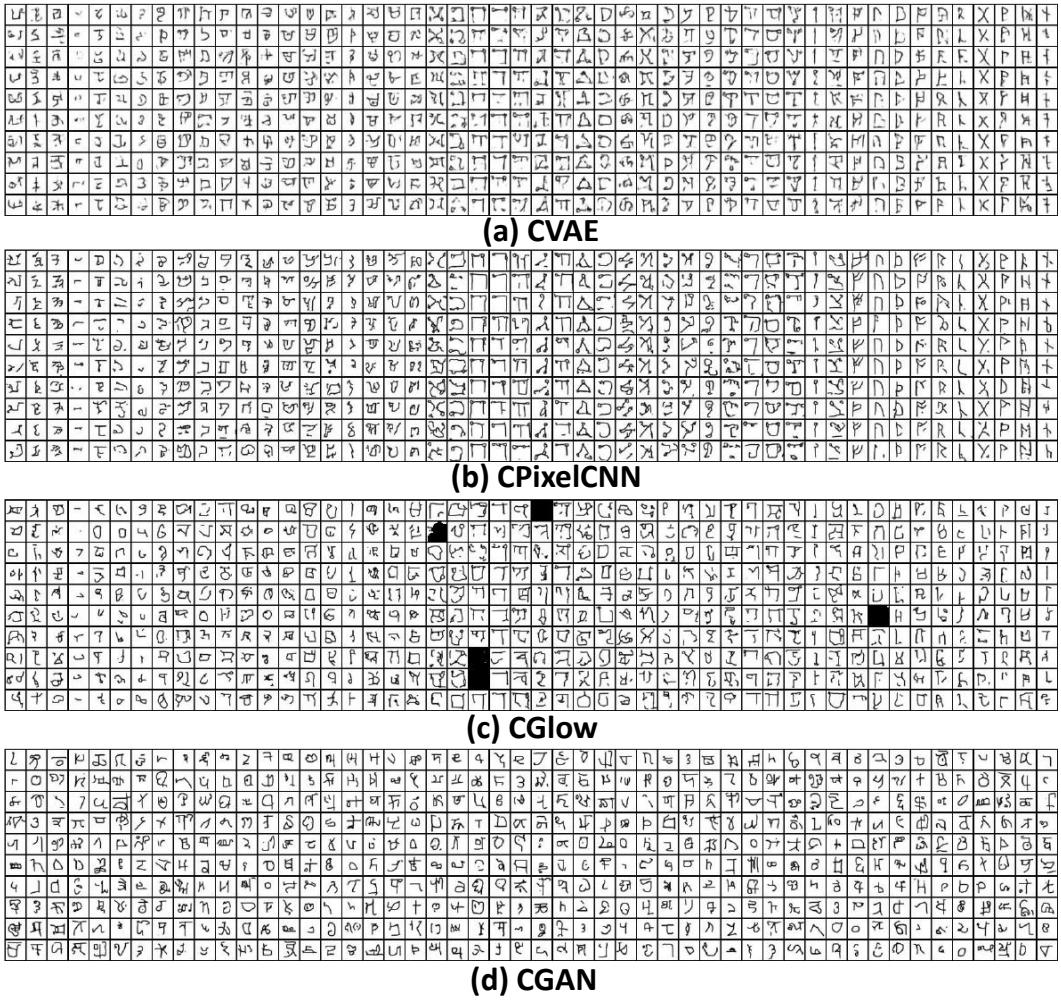
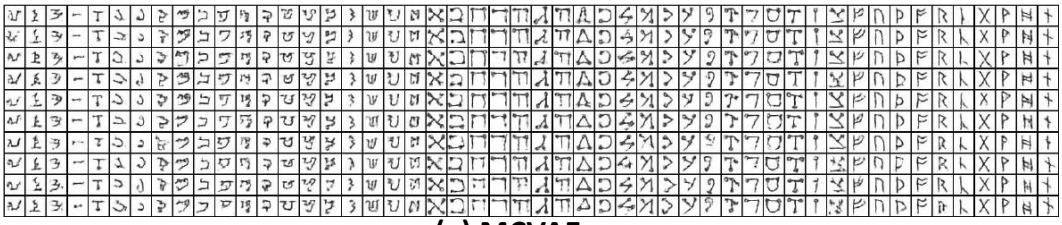
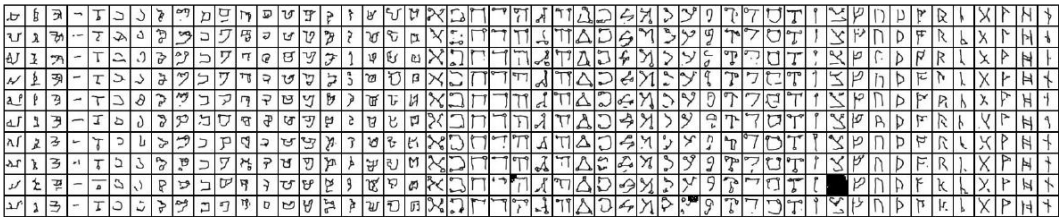


Figure 7: Generations from conditional generative models trained with Omniglot.

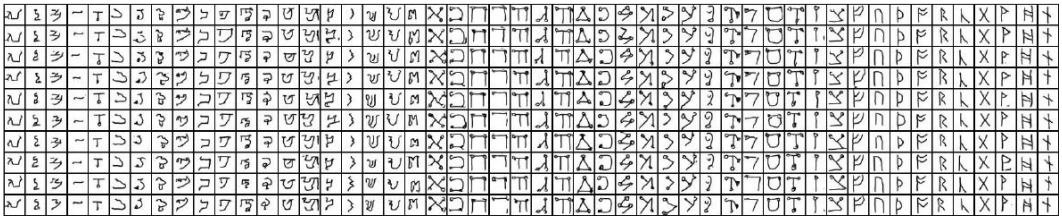


(a) MCVAE

(b) MCPixelCNN

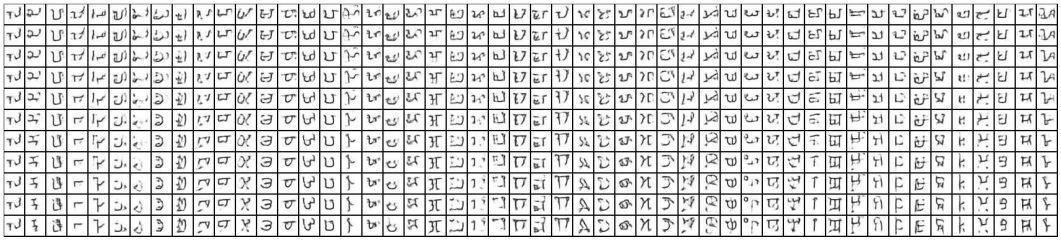


(c) MCGlow

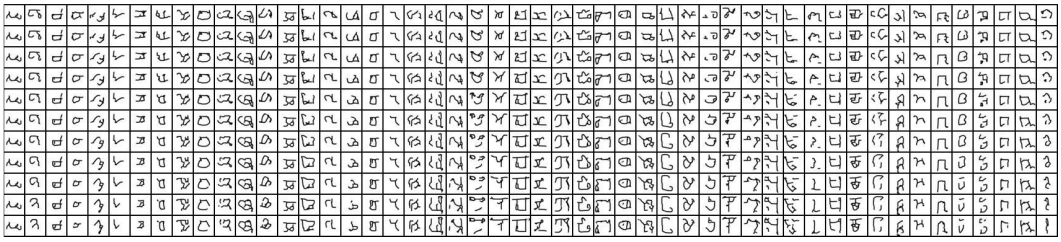


(d) MCGAN

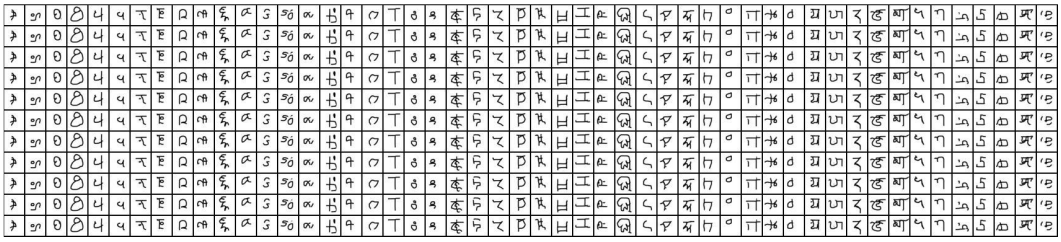
Figure 8: Generations from multimodal controlled generative models trained with Omniglot.



(a) CVAE

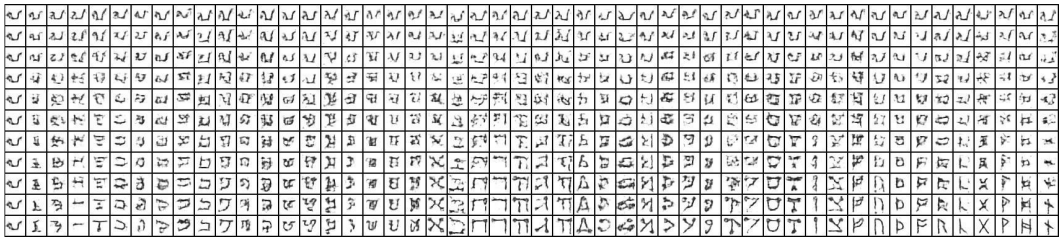


(b) CGlow

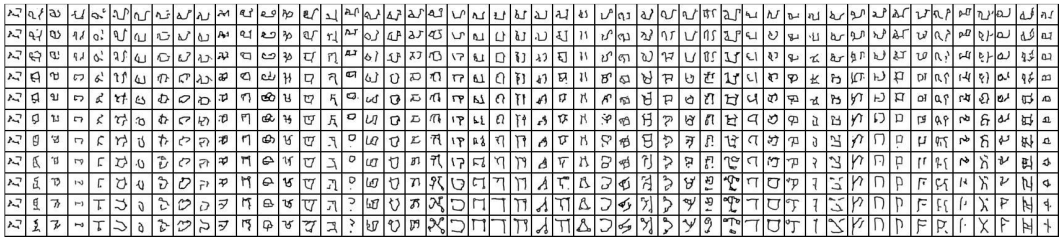


(c) CGAN

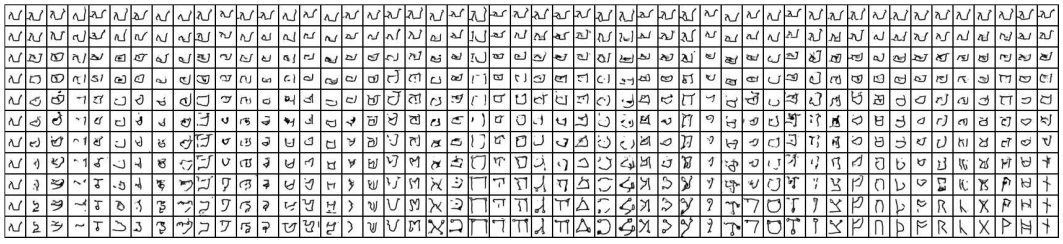
Figure 9: Transitions from conditional generative models trained with Omniglot.



(a) MCVAE



(b) MCGlow



(c) MCGAN

Figure 10: Transitions from multimodal controlled generative models trained with Omniglot.

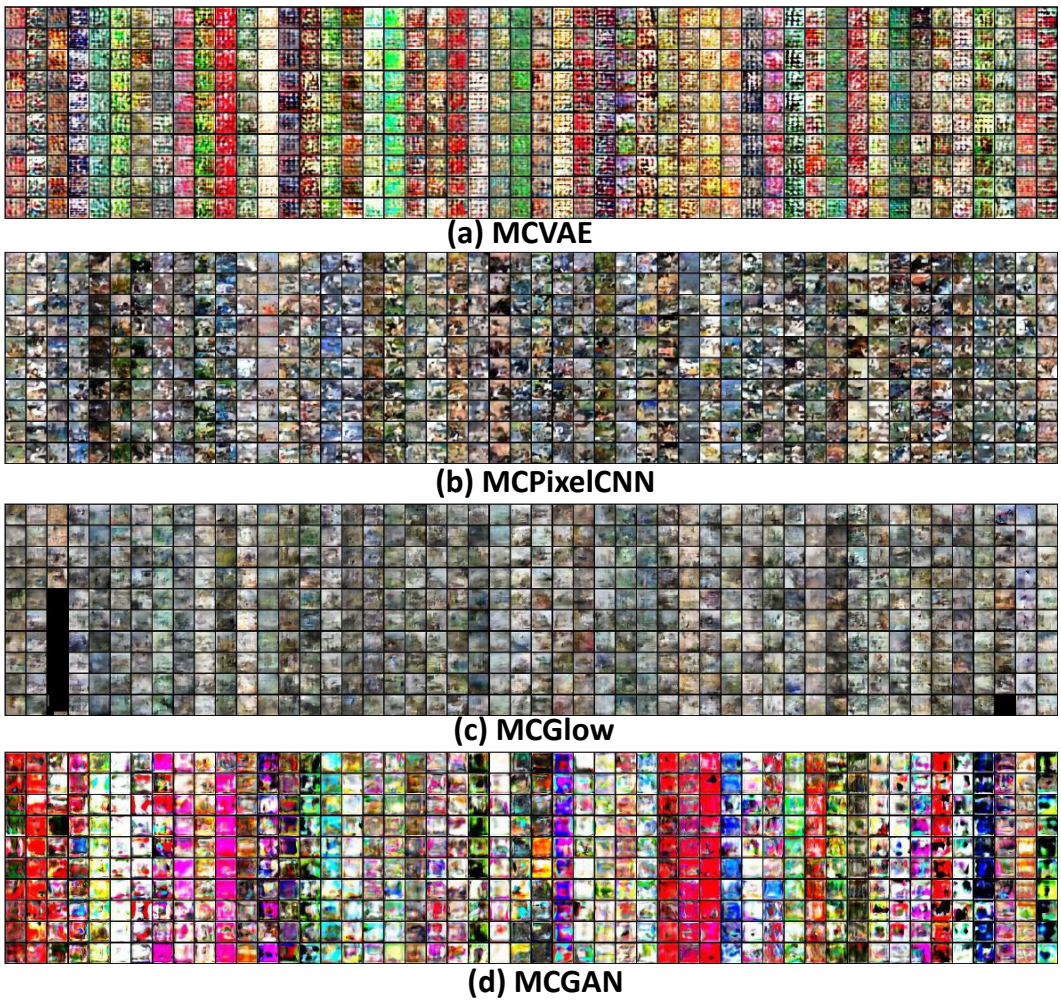
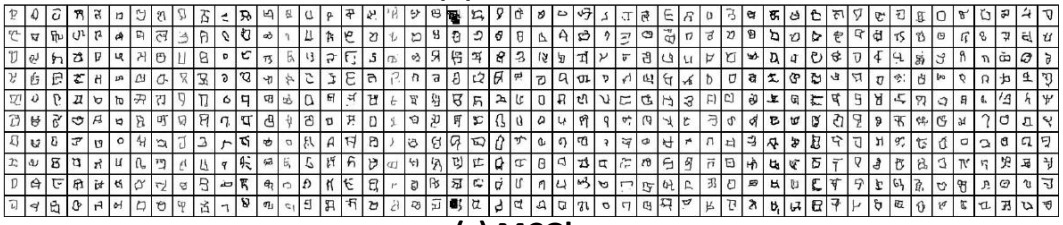


Figure 11: Creations from multimodal controlled generative models trained with CIFAR10.

(a) MCVAE

(b) MCPixelCNN



(c) MCGlow

(d) MCGAN

Figure 12: Creations from multimodal controlled generative models trained with Omniglot.