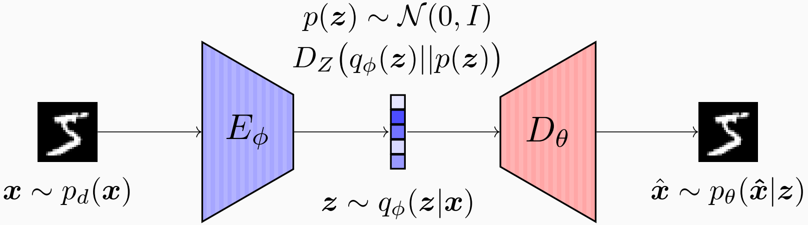
The landscape of Regularized Auto-Encoders for Generative modeling

Dr. Prathosh A.P. ,
Assistant Professor,
Department of Electrical Engineering,
Indian Institute of Technology (IIT), Delhi.

Collaborators and students:
Prof. Himanshu Asnani, TIFR and Prof. Parag Singla, IITD.
Arnab Kumar Mondal and Aravind. J, IITD.

Regularized Auto-Encoder based Generative Models

AE with Regularized Latent Space



$p(\boldsymbol{z}) \sim \mathcal{N}(0, I)$
$D_Z\big(q_\phi(\boldsymbol{z})||p(\boldsymbol{z})\big)$
E_ϕ
D_θ
$\boldsymbol{x} \sim p_d(\boldsymbol{x})$
$\boldsymbol{z} \sim q_\phi(\boldsymbol{z}|\boldsymbol{x})$
$\hat{\boldsymbol{x}} \sim p_\theta(\hat{\boldsymbol{x}}|\boldsymbol{z})$

Figure 1: Architectural diagram of a Regularized Auto-Encoder [1].

- The Objective - Given $\{x_i\}_{i=1}^{i=n} \sim p_d(x)$, learn to sample from $p_d(x)$.
- $p_\theta(x) = \int_Z p_\theta(x|z)p(z)\,dz$ (Generative data distribution)
- E_ϕ and D_θ - Probabilistic/Deterministic Encoder and Decoder.
- $p(z) \sim \mathcal{N}(0, \mathrm{I})$, is the latent prior, acts as regularizer.
- $Q_\phi(z) = \int q_\phi(z|x)p_d(x)\,dx$ (aggregated encoded posterior

The Objective Functions (VAE and variants):

Log-likelihood $LLE(\theta)$ of the data distribution under a model $p_\theta(x)$:

$$\begin{aligned} LLE(\theta)/D_{KL}\left[p_d(x)||p_\theta(x)\right] = & \underbrace{\mathop{\mathbb{E}}_{p_d(x)q_\phi(z|x)}\left[\log p_\theta(x|z)\right]}_{\text{I}} - \underbrace{D_{KL}(q_\phi(z)||p(z))}_{\text{II}} \\ & - \underbrace{\mathbb{I}(x; z_\phi)}_{\text{III}} + \underbrace{\mathop{\mathbb{E}}_{p_d(x)}\left[D_{KL}(q_\phi(z|x)||p_\theta(z|x))\right]}_{\text{IV}} \end{aligned} \tag{1}$$

(I+II+III) is the Evidence Lower bound $ELBO(\theta, \phi) \leq LLE(\theta)$, $\because$ $\mathrm{D}_{KL} \geq 0$.

$$\begin{aligned} ELBO(\theta, \phi) &= \mathop{\mathbb{E}}_{p_d(x)q_\phi(z|x)}\left[\log p_\theta(x|z)\right] - D_{KL}(q_\phi(z)||p(z)) - \mathbb{I}(x; z_\phi) \\ &= \mathop{\mathbb{E}}_{p_d(x)q_\phi(z|x)}\left[\log p_\theta(x|z)\right] - D_{KL}(q_\phi(z|x)||p(z)) \end{aligned} \tag{2}$$

The Objective Functions (AAE/WAE)

Given the data distribution $p_{(x)}$ and the distribution learned by a model $p_\theta(x)$:

$$D_{WD}\left[p_d(x), p_\theta(x)\right] = \inf_{Q_\phi(z|x)\sim\mathcal{Q}} \left(\mathbb{E}_{P_d}\, \mathbb{E}_{Q_\phi(z|x)} \left[c\big(x, D_\theta(z)\big)\right]\right.$$

$$\text{such that } \mathrm{Q}_\phi(z) = P(z)$$

$$= \inf_{Q_\phi(z|x)\sim\mathcal{Q}} \left(\underbrace{\mathop{\mathbb{E}}_{P_d}\, \mathop{\mathbb{E}}_{Q_\phi(z|x)} \left[c\big(x, D_\theta(z)\big)\right]}_{\text{a}} + \lambda \cdot \underbrace{D_Z\big(Q_\phi(z), P(z)\big)}_{\text{b}} \right)$$

$c : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}^+$ is any measurable cost function and D_Z is any divergence metric.

AE-based Generative Models: Background

- All AE-based generative models optimize likelihood/divergence metric or its lower bound.
- First term in the ELBO, approximated by MSE, is the conditional generated data likelihood.
- Second term, D_{KL}, acts as the regularizer on the latent space.
- Variational Auto Encoder (VAE) [1]: Assumes Gaussian Encoder and Decoder with stochastic reparameterization.
- Adversarial Auto Encoder (AAE) and Wasserstein Auto Encoder (WAE) [2, 3] exploits adversarial training to match the aggregated posterior with the prior.
- Stable training, efficient sampling, flexible architectural choices and richer/interpretable latent space, still not reached GAN-level performance.

AE based generative models: Issues and remedies

- Likelihood (Term 1) and KL terms at loggerheads (Term 2).
- Distributional choices for Encoder and Decoder are restrictive.
- Aggregated latent posterior $Q(\boldsymbol{z})$ doesn't match with the prior.
- β-VAE [4]: Introduces a tunable parameter in the second term.
- InfoVAE/FactorVAE- Additional penalties such as mutual information [5], total correlation [6].
- Many works [7, 8, 9, 10, 11] implement non Gaussian distributional choices for Encoder/Decoder models.
- [12, 13, 14] uses a richer class of priors on the latent space (GMMs, hierarchical models) to match aggregated posterior.
- [15, 16, 17] implements a post-hoc sampler in the latent space without regularizing it.

Focus of the current work

- Examine two questions on the latent space of AE models:
    1. What is the effect of latent space dimensionality on AE-based generative models?
    2. Whats are the ‘optimal’ latent prior for RAEs?
- Discuss two novel AE models: MaskAAE and FlexAE to address these issues.

Effect of the Latent Space Dimensionality on AEs (MaskAAE)

Motivation

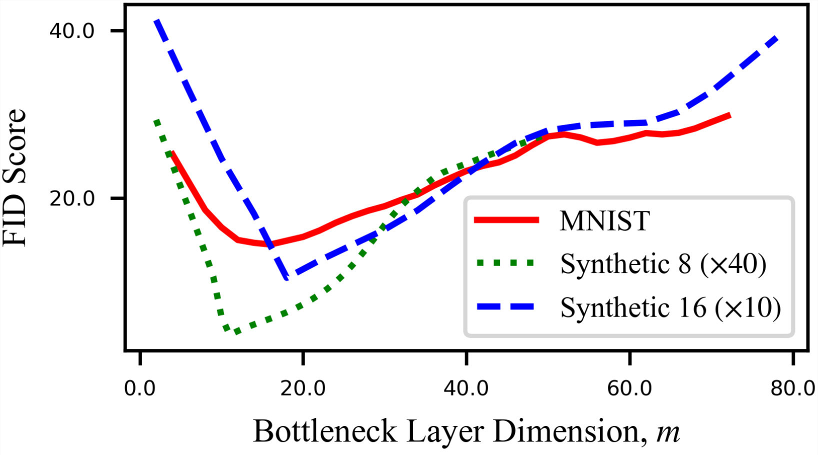
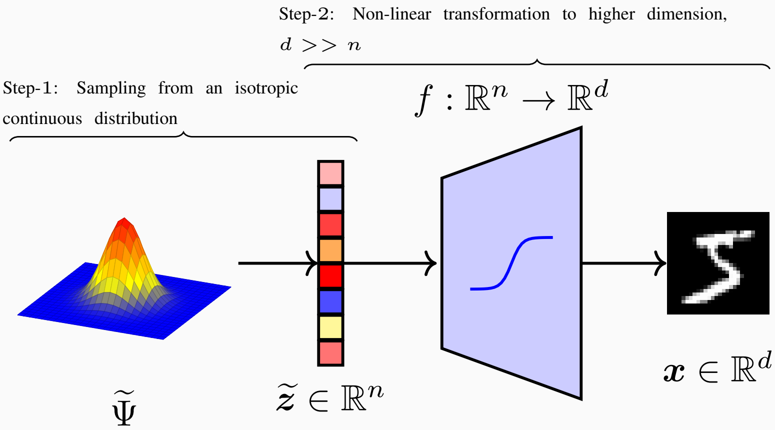


Figure 2: Scaled FID score for a WAE with varying latent dimensionality m for 2 synthetic datasets of 'true' latent dimensions, $n = 8$ and $n = 16$ and MNIST. It is seen that the generation quality gets worse on both the sides of a certain latent dimensionality.

Data Generation Hypothesis



Step-2: Non-linear transformation to higher dimension, $d >> n$
Step-1: Sampling from an isotropic continuous distribution
$f : \mathbb{R}^n \rightarrow \mathbb{R}^d$
$\widetilde{\Psi}$
$\widetilde{\boldsymbol{z}} \in \mathbb{R}^n$
$\boldsymbol{x} \in \mathbb{R}^d$

Figure 3: Depiction of the assumed two-step data generation process. Samples drawn from a 'true' latent distribution $\widetilde{\Psi}(\widetilde{z})$ are passed through a function f to obtain $\boldsymbol{x}$.

Assumptions on the generative function f

A1 f is *L-lipschitz*: $\exists$ some finite $L \in \mathbb{R}^+$ satisfying $||f(\widetilde{\mathbf{z}}_1) - f(\widetilde{\mathbf{z}}_2)|| \leq L||\widetilde{\mathbf{z}}_1 - \widetilde{\mathbf{z}}_2||,\ \forall \widetilde{\mathbf{z}}_1, \widetilde{\mathbf{z}}_2 \in \widetilde{\mathcal{Z}}$.

A2 There does not exist $f^* : n' \to d, n' < n$ satisfying A1 such that the range of f is a subset of the range of f^*.

Requirements for Good Generation

The goal of latent variable generative models is to minimize the negative log-likelihood of $\Gamma'(x,z)$ under $\Gamma(x,z)$:

$$\mathcal{L}(\Gamma,\Gamma') = -\mathop{\mathbb{E}}_{x,z\sim\Gamma}\left[\log(\Gamma'(x,z))\right] \tag{3}$$

Equation 3 can be broken down as follows:

$$\min\left(\underbrace{\mathop{\mathbb{E}}_{\Gamma}[-\log(\Gamma'(x|z))]}_{\text{R1}} + \underbrace{\mathop{\mathbb{E}}_{\Gamma}[\log\frac{1}{\Gamma'(z)}]}_{\text{R2}}\right) \tag{4}$$

R1 $f(\widetilde{z}) = g'(g(f(\widetilde{z})))\ \forall\,\widetilde{z}\ \in n$. This condition states that the reconstruction error between the real and generated data should be minimal.

R2 The Cross Entropy $\mathcal{H}(\Psi,\Pi)$ between the chosen prior Ψ, and Π on $\mathcal{Z}$ is minimal.

Conditions to Satisfy R1 and R2

The conditions required to ensure R1 and R2 are met with assumed data generation process are:

Theorem
With the assumption of data generating hypothesis, requirements R1 and R2, can be satisfied iff assumed latent dimension m is equal to true latent dimension n.

- For $m < n$, A2 will be violated since range of $f \subset$ range of g'.
- For $m > n$, the range of $\circ f$ will have 0 Lebesgue measure leading to arbitrarily large $\mathcal{H}$.

The solution: MaskAAE

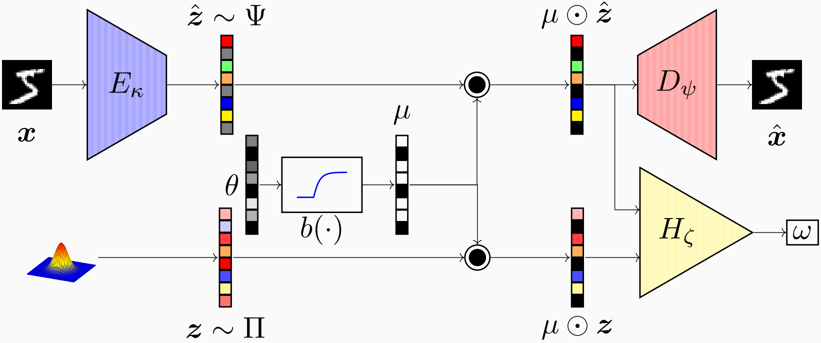
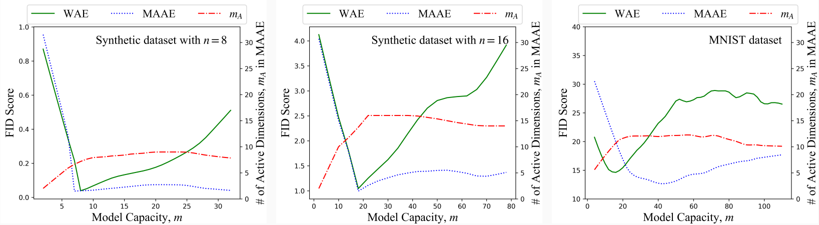


Figure 4: Block Diagram of MaskAAE. It consists of an encoder, E_κ, a decoder, D_ψ, and a discriminator H_ζ as in AAE/WAE. A new layer called mask, μ is introduced at the end of the encoder to suppress spurious latent dimensions. The prior also gets multiplied with the same mask before going into the Discriminator to ensure prior matching (R2).

Experimental Results: Effect of Latent Dimension on Generation



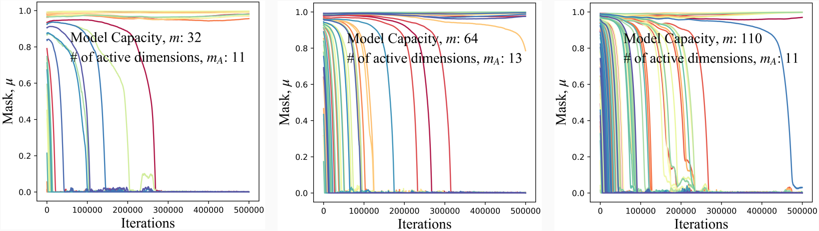
(a)

(b)

(c)

Figure 5: (a) and (b) shows FID score for WAE and MAAE and active dimension in a trained MAAE model with varying model capacity, m for synthetic dataset of true latent dimensions, $n = 8$ and $n = 16$, m_A represents the number of unmasked latent dimensions in the trained model and (c) shows the same plots for MNIST dataset.

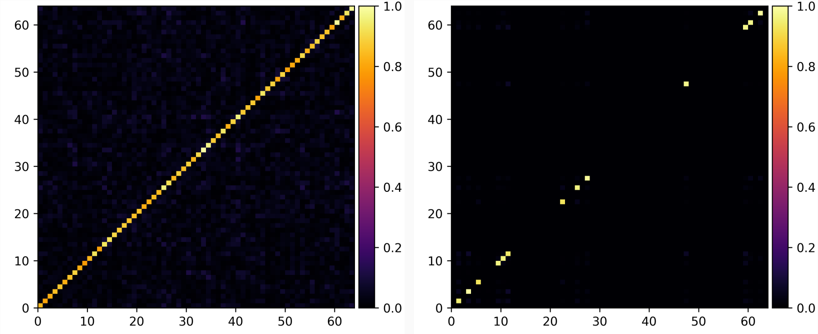
Experimental Results: Behaviour of Mask



(a) (b) (c)

Figure 6: Behaviour of mask in MAAE models with different m for the MNIST dataset. Model capacity, m, in figure (a), (b), and (c) are 32, 64, and 110, respectively. The active dimensions after training are m_A are 11, 13, and 11 respectively.

Experimental Results: Covariance Matrix of the Latent Vectors for MNIST



(a)

(b)

Figure 7: Co-variance Matrix of (a) WAE (b) MAAE latent representation for MNIST dataset.

Experimental Results: FID

Table 1: FID scores for generated images from different AE-based generative models (Lower is better).

	MNIST	Fashion	CIFAR-10	CelebA
VAE (cross-entr.)	16.6	43.6	106.0	53.3
VAE (fixed variance)	52.0	84.6	160.5	55.9
VAE (learned variance)	54.5	60.0	76.7	60.5
VAE + Flow	54.8	62.1	81.2	65.7
WAE-MMD	115.0	101.7	80.9	62.9
WAE-GAN	12.4	31.5	93.1	66.5
2-Stage VAE	12.6	29.3	72.9	44.4
MAAE	**10.5**	**28.4**	**71.9**	**40.5**

Experimental Results: Normalized Absolute Correlation

Table 2: Average off-diagonal covariance NAC for both WAE and MAAE. m_A represents the number of unmasked latent dimensions in the trained model. It is seen that MAAE has lower NAC values indicating lesser deviation of $\Psi(z)$ from $\Pi(z)$ as compared to a WAE.

Dataset	Model Capacity	WAE		MAAE	
		m_A	NAC	m_A	NAC
Synthetic$_8$	16	16	0.040	9	**0.030**
Synthetic$_{16}$	32	32	0.031	16	**0.013**
MNIST	64	64	0.027	13	**0.020**
FMNIST	128	128	0.025	40	**0.019**
CIFAR-10	256	256	0.017	120	**0.013**
CelebA	256	256	0.046	77	**0.039**

To regularize or not - Effect of prior in AE (FlexAE)

Conditions for optimality (II)

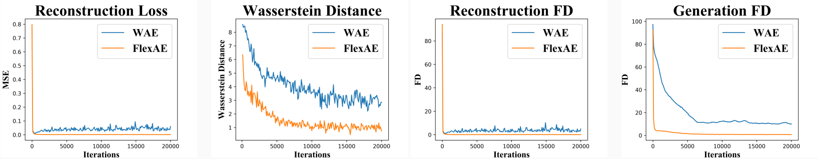
Theorem

If $m > n$, then the divergence term in the WAE objective $D_Z(Q_\phi(\mathbf{z}), P(\mathbf{z})) > 0,\ \forall\phi$ and for any distributional divergence D_Z when $p_z \sim \mathcal{N}(0, I_{m\times m})$.

Corollary

When $m > n$, if $P_Z \notin Q_m^n$ then $D_Z\big(Q_\phi(\mathbf{z}), P(\mathbf{z})\big) > 0,\ \forall\phi$ and for any distributional divergence D_Z. WAE objective has a feasible solution iff $P(\mathbf{z}) \in Q_m^n$.

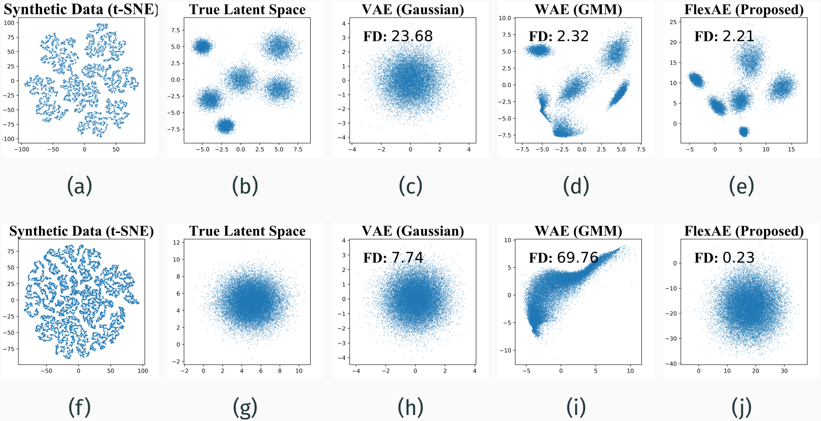
Motivation



(a) (b) (c) (d)

Figure 8: Comparison of RAEs with fixed and learnable latent priors.

Motivation



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Figure 9: Visualization of data (t-SNE) and the learnt latent space of different AE-based generative models for the synthetic data.

Choosing the “Right” Prior: The Bias-variance Trade-off

- Question - Can we do away with the prior on latent space?
- Amortized sampling via post-hoc samplers on latent space.
- Answer: No. There exists a bias-variance trade-off in practice.
- The generalized objective:

$$D_{FlexAE}(P_X, P_\theta) =$$

$$\inf_{\phi,\theta,\psi} \left(\underbrace{\mathop{\mathbb{E}}_{P(x)} \mathop{\mathbb{E}}_{Q(z|x)} \Big[c(x, D_\theta(z)) \Big]}_{\text{a}} + \lambda \cdot \underbrace{D_Z(q_\phi(z) || p_\psi(z))}_{\text{b}} \right)$$

FlexAE

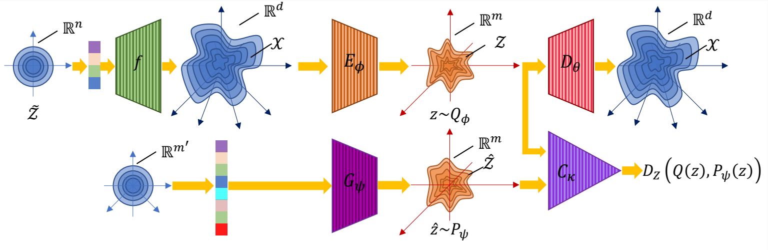


Figure 10: Nature first samples a n-dimensional latent code from the true latent space, $\tilde{\mathcal{Z}}$. Next, the latent code is mapped to a n-dimensional manifold, $\mathcal{X}$ in a d-dimensional ambient space. The observed variables are encoded using deterministic encoder, E_ϕ. The m-dimensional encoded representations lie in a n-dimensional manifold $\mathcal{Z}$. The decoder network, D_θ, learns an inverse projection from the learnt latent space, $\mathcal{Z}$ to the dataspace, $\mathcal{X}$. The generator netowrk, G_ψ parameterizes the learnable prior distribution. Dimensionality of the latent space of the prior generator, $m^{'} \geq m$. The critic network, C_κ measures the distributional divergence between Q_ϕ and P_ψ.

Experimental Results: FID

Table 3: Comparison of FID scores [18] on real datasets. Lower is better.

	MNIST		CIFAR10		CELEBA	
	Rec.	Gen.	Rec.	Gen.	Rec.	Gen.
VAE [1]	65.10	57.04	176.5	169.1	62.36	72.48
β-VAE [4]	7.91	24.31	43.86	83.59	30.06	50.66
VAE-Vamprior [12]	11.01	49.75	107.33	161.02	49.71	64.26
VAE-IOP [17]	8.01	32.61	92.17	141.92	41.52	57.30
WAE-GAN [3]	8.06	13.30	42.39	72.90	29.34	39.58
AE + GMM (L2) [16]	8.69	12.14	41.45	70.97	30.16	43.89
RAE + GMM (L2) [16]	6.15	7.30	40.48	69.24	29.05	35.30
VAE + FLOW [8]	8.62	20.17	43.87	73.28	36.31	42.39
InjFlow^{ln} [14]	7.40	35.96	40.11	78.78	27.93	47.70
InjFlow^{ln} + GMM [14]	7.40	9.93	40.11	68.26	27.93	40.23
2-S VAE [19]	6.38	7.41	47.03	86.15	29.38	37.85
MaskAAE [20]	8.46	10.52	58.40	71.90	35.75	40.49
FlexAE (Proposed)	**4.33**	**4.69**	**39.91**	**62.66**	**20.47**	**24.72**

Experimental Results: Precision/Recall Scores

Table 4: Comparison of Precision/Recall scores [21] on real datasets. Higher is better.

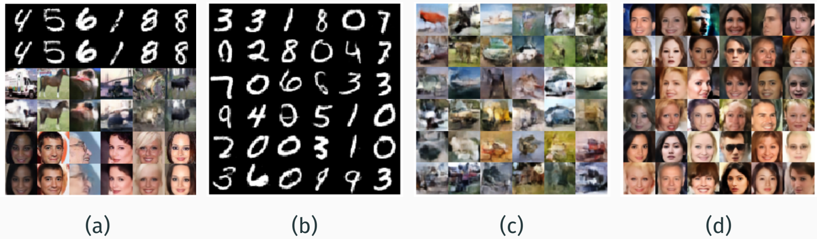
	MNIST	CIFAR10	CELEBA
VAE [1]	0.69/0.76	0.23/0.47	0.47/0.58
2S-VAE [19]	0.97/0.98	0.47/0.76	0.75/0.72
RAE + GMM (L2) [16]	0.98/0.98	0.61/0.87	0.74/0.75
MaskAAE [20]	0.94/0.96	0.58/0.83	0.59/0.68
FlexAE (Proposed)	**0.99/0.99**	**0.68/0.85**	**0.89/0.88**

Experiment: Bias-Variance Trade-off

Table 5: Variation of reconstruction and generation FID scores on limited training datasets with varying P-GEN capacity, demonstrating bias-variance trade-off. Models (1-6) are presented in increasing order of capacity.

	Model 1		Model 2		Model 3		Model 4		Model 5		Model 6	
	Rec.	Gen.	Rec.	Gen.	Rec.	Gen.	Rec.	Gen.	Rec.	Gen.	Rec.	Gen.
MNIST	60.51	55.49	21.00	53.93	13.41	42.14	14.40	**31.00**	8.11	63.64	8.94	62.43
CIFAR-10	154.17	135.32	91.85	104.06	82.95	108.63	83.88	**108.46**	94.2	120.64	94.54	121.96
CELEBA	79.04	66.84	42.77	56.16	47.02	54.32	42.75	**54.14**	44.02	59.3	39.1	58.49

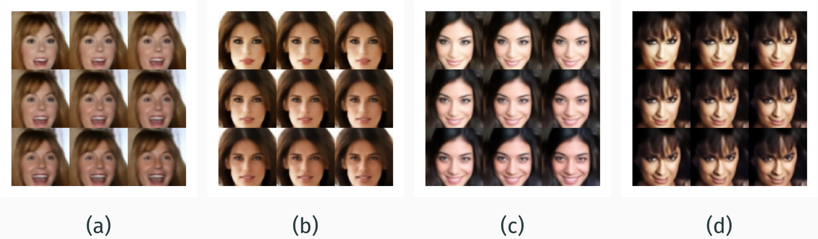
Experimental Results: Reconstructed and Generated Samples



(a) (b) (c) (d)

Figure 11: (a) Visualization of reconstruction quality of FlexAE model on randomly selected data from the test split of MNIST (first and second rows), CIFAR-10 (third and fourth rows) and CELEBA (fifth and sixth rows). The odd rows represent the real data and the even rows represent reconstructed data. Randomly generated samples from (b) MNIST, (c) CIFAR-10, and (d) CELEBA datasets using FlexAE model.

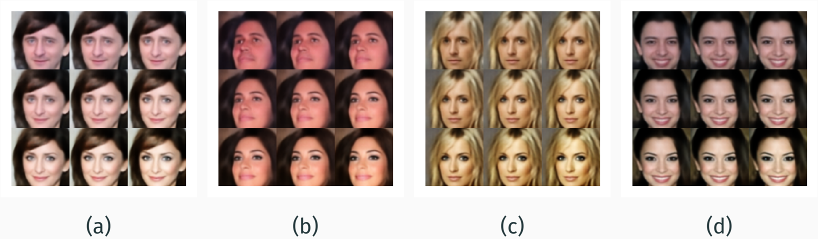
Experimental Results: Attribute Manipulation Via Interpolation in Latent Space i



(a) (b) (c) (d)

Figure 12: Interpolations in the latent space of FlexAE on CelebA. Each row in (a) and (b) presents manipulation of the attribute “Big Nose”. The central image of each grid in (a), and (b) is a true image from the test split without the attribute. Whereas, the central image of each grid in (c) and (d) is a true image from the test split with the attribute.

Experimental Results: Attribute Manipulation Via Interpolation in Latent Space ii



(a)

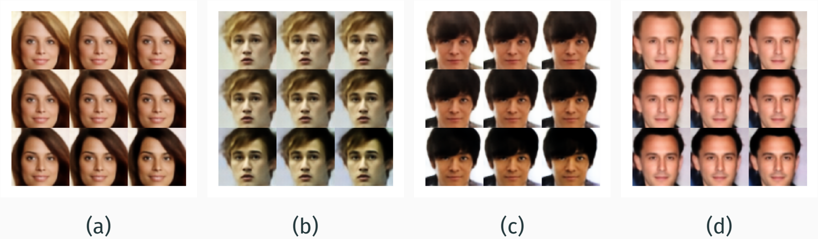
(b)

(c)

(d)

Figure 13: Interpolations in the latent space of FlexAE on CelebA. Each row in (a) and (b) presents manipulation of the attribute “Heavy Makeup”. The central image of each grid in (a), and (b) is a true image from the test split without the attribute. Whereas, the central image of each grid in (c) and (d) is a true image from the test split with the attribute.

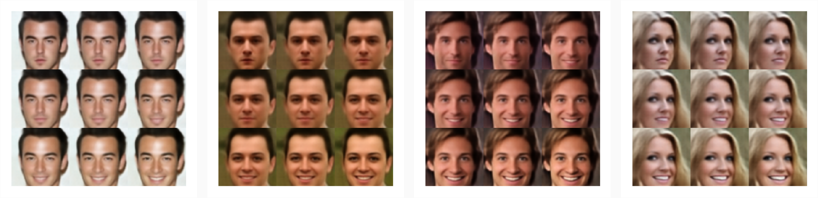
Experimental Results: Attribute Manipulation Via Interpolation in Latent Space iii



(a) (b) (c) (d)

Figure 14: Interpolations in the latent space of FlexAE on CelebA. Each row in (a) and (b) presents manipulation of the attribute “Black Hair”. The central image of each grid in (a), and (b) is a true image from the test split without the attribute. Whereas, the central image of each grid in (c) and (d) is a true image from the test split with the attribute.

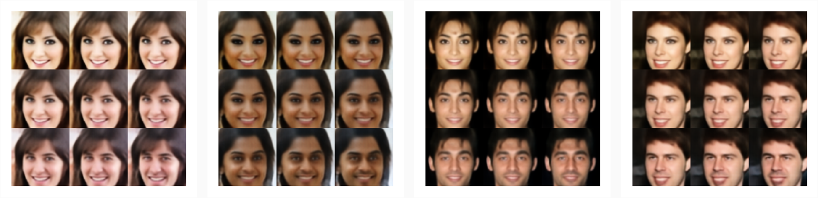
Experimental Results: Attribute Manipulation Via Interpolation in Latent Space iv



(a) (b) (c) (d)

Figure 15: Interpolations in the latent space of FlexAE on CelebA. Each row in (a) and (b) presents manipulation of the attribute “Smiling”. The central image of each grid in (a), and (b) is a true image from the test split without the attribute. Whereas, the central image of each grid in (c) and (d) is a true image from the test split with the attribute.

Experimental Results: Attribute Manipulation Via Interpolation in Latent Space v



(a) (b) (c) (d)

Figure 16: Interpolations in the latent space of FlexAE on CelebA. Each row in (a) and (b) presents manipulation of the attribute “Male”. The central image of each grid in (a), and (b) is a true image from the test split without the attribute. Whereas, the central image of each grid in (c) and (d) is a true image from the test split with the attribute.

Experimental Results: k-Nearest Training Split Neighbours of Generated Images

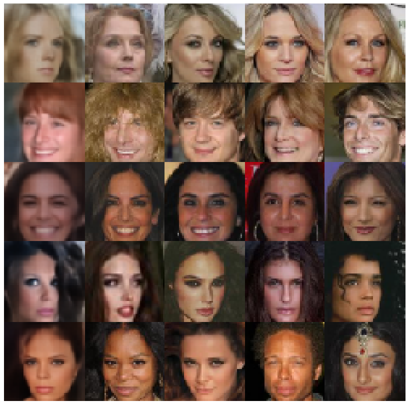


Figure 17: The first entry in each row represents a randomly generated face using FlexAE. The remaining entries in each row represents 4 nearest neighbours (in terms of Euclidean distance) from the train split of CELEBA dataset.

Conclusion

- RAEs are a powerful alternatives to GANs for generative modeling.
- Dimensionality mismatch between the true and assumed latent is a major concern.
- Described two methods to alleviate them.
- Next important question - Identifiability of RAEs.

References i

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

[2] A. Makhzani, J. Shlens, N. Jaitly, and I. Goodfellow, "Adversarial autoencoders," in *Proc. of ICLR*, 2016.

[3] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Scholkopf, "Wasserstein auto-encoders," in *Proc. of ICLR*, 2018.

[4] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "β-VAE: Learning basic visual concepts with a constrained variational framework," in *Proc. of ICLR*, 2017.

[5] S. Zhao, J. Song, and S. Ermon, "InfoVAE: Balancing learning and inference in variational autoencoders," in *Proc. of AAAI*, 2019.

References ii

[6] H. Kim and A. Mnih, “Disentangling by factorising,” in *Proc. of ICML*, 2018.

[7] E. Nalisnick and P. Smyth, “Stick-breaking variational autoencoders,” in *Proc. of ICLR*, 2017.

[8] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, “Improved variational inference with inverse autoregressive flow,” in *Proc. of NeuRIPS*, 2016.

[9] D. J. Rezende and S. Mohamed, “Variational inference with normalizing flows,” in *Proc. of ICML*, 2015.

[10] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, “Autoencoding beyond pixels using a learned similarity metric,” in *Proc. of ICML*, 2016.

References iii

[11] M. Rosca, B. Lakshminarayanan, and S. Mohamed, “Distribution matching in variational inference,” *arXiv preprint arXiv:1802.06847*, 2018.

[12] J. M. Tomczak and M. Welling, “VAE with a vampprior,” in *Proc. of AISTATS*, 2018.

[13] A. Klushyn, N. Chen, R. Kurle, B. Cseke, and P. van der Smagt, “Learning hierarchical priors in VAEs,” in *Proc. of NeuRIPS*, 2019.

[14] A. Kumar, B. Poole, and K. Murphy, “Regularized autoencoders via relaxed injective probability flow,” *arXiv preprint arXiv:2002.08927*, 2020.

[15] M. Bauer and A. Mnih, “Resampled priors for variational autoencoders,” in *Proc. of AISTATS*, 2019.

References iv

[16] P. Ghosh, M. S. M. Sajjadi, A. Vergari, M. Black, and B. Scholkopf, “From variational to deterministic autoencoders,” in *Proc. of ICLR*, 2020.

[17] H. Takahashi, T. Iwata, Y. Yamanaka, M. Yamada, and S. Yagi, “Variational autoencoder with implicit optimal priors,” in *Proc. of AAAI*, 2019.

[18] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “Gans trained by a two time-scale update rule converge to a local nash equilibrium,” in *Proc. of NeuRIPS*, 2017.

[19] B. Dai and D. Wipf, “Diagnosing and enhancing vae models,” in *Proc. of ICLR*, 2019.

References v

[20] A. K. Mondal, S. P. Chowdhury, A. Jayendran, P. Singla, H. Asnani, and A. Prathosh, “MaskAAE: Latent space optimization for adversarial auto-encoders,” in *Proc. of UAI*, 2020.

[21] M. S. M. Sajjadi, O. Bachem, M. Lucic, O. Bousquet, and S. Gelly, “Assessing generative models via precision and recall,” in *Proc. of NeuRIPS*, 2018.