

A PROOF OF LEMMA 2

Lemma 2. *Let \mathcal{G} be the space of smooth invertable functions with smooth inverse (i.e., a diffeomorphism) that map \mathcal{Z} to \mathcal{X} , and $h : \mathcal{Z} \rightarrow \mathcal{Z}$ is a smooth invertable function. Then, any function $g \in \mathcal{G}$ can be represented as $g = g^* \circ h$, where $g^* : \mathcal{Z} \rightarrow \mathcal{X}$ is the ground-truth disentangled generative model in the function space \mathcal{G} . Formally, we have $g \sim_h g^*, \forall g \in \mathcal{G}$ and the model g is \sim_h -identifiable.*

Proof. By definition, $g : \mathcal{Z} \rightarrow \mathcal{X}$ and $g^* \circ h : \mathcal{Z} \rightarrow \mathcal{X}$ are equal if their domain and codomain are the same and $g(z) = g^*(h(z))$ for any $z \in \mathcal{Z}$. For $g \in \mathcal{G}$ and $g^* \circ h$, their domain and codomain are defined to be the same. Then, we let h to be a permutation function, which is invertable, that satisfies $g(z) = g^*(h(z)), \forall z \in \mathcal{Z}$. First, we choose a $z \in \mathcal{Z}$ and let $x = g(z)$. Then, we can construct $h(z) = z^*$ such that $g^*(z^*) = g^*(h(z)) = x$. We can apply this procedure to every $z \in \mathcal{Z}$. This is because we assume the mapping between \mathcal{Z} and \mathcal{X} are bijective, which means there is no conflict. As we assume h to be smooth, which is not implicitly satisfied by the permutation function, we show the smooth assumption holds by showing the limit of h exists everywhere on \mathcal{Z} :

$$\begin{aligned} \lim_{\Delta z \rightarrow 0} \frac{h(z + \Delta z) - h(z)}{\Delta z} &= \lim_{\Delta z \rightarrow 0} \frac{g^{*-1}(g(z + \Delta z)) - g^{*-1}(g(z))}{\Delta z} \\ &= g^{*-1'}(g(z)) \times g'(z) \end{aligned} \quad (5)$$

As we know that both g and g^{*-1} are smooth, their composition, $g^{*-1} \circ g$, are also smooth. Thus, the limit above exist for any $z \in \mathcal{Z}$. By definition, h is a smooth function. Thus, we can always find a smooth invertable permutation function h for any $g \in \mathcal{G}$ such that $g = g^* \circ h, \forall g \in \mathcal{G}$. As the valid output x for g and $g^* \circ h$ identical, we have $g \sim_h g^*$ by definition. As the equivalence relation holds between any $g \in \mathcal{G}$ and g^* that admits the same marginal distribution, g is \sim_h -identifiable. \square

B PROOF OF THEOREM 3

Theorem 3. *Let $p(z)$ denote the unconventional mixture prior. We use \mathcal{M} to denote the manifold where the ground-truth latent variable z lies on. Let $k \in \{1, 2, \dots, N\}$ be the index of generative mechanisms M . We assume each z_{M_k} lies on \mathcal{M}_{M_k} and z_S lies on \mathcal{M}_S . We let $\hat{z} = h(z)$ which lies on $\hat{\mathcal{M}}$. g^* is the ground-truth disentangled model. Then, if there exists an smooth invertable function $h : \mathcal{Z} \rightarrow \mathcal{Z}$ such that $g = g^* \circ h$ maps \mathcal{Z} to \mathcal{X} , we have h maps each \mathcal{M}_{M_k} to disjoint sub-manifold $\hat{\mathcal{M}}_{M_k}$ and maps \mathcal{M}_S to $\hat{\mathcal{M}}_S$, which is disjoint from all $\hat{\mathcal{M}}_{M_k}$.*

Proof. By the definition of unconventional mixture prior, we have: if $\hat{z}_{M_i} \neq \mathbf{0}$, $\hat{z}_{M_j} = \mathbf{0}$ for all $j \neq i$. We name this condition as the structure constraint. As we know that $h : \mathcal{Z} \rightarrow \mathcal{Z}$ and $g^* : \mathcal{Z} \rightarrow \mathcal{X}$, we have $\mathcal{M} = \hat{\mathcal{M}}$ because the valid inputs for g^* and $g = g^* \circ h$ are identical. The structure constraint is enforced on both \mathcal{M} and $\hat{\mathcal{M}}$. If h entangles $\hat{\mathcal{M}}_{M_i}$ and $\hat{\mathcal{M}}_{M_j}$ for any $i \neq j$, which means the change in one ground-truth z_{M_k} will be reflected in two \hat{z}_{M_i} and \hat{z}_{M_j} where $i \neq j$. There exist a z such that $h(z)_{M_i} = \hat{z}_{M_i} \neq \mathbf{0}$ and $h(z)_{M_j} = \hat{z}_{M_j} \neq \mathbf{0}$ holds simultaneously. For example, we can let $[z_{M_i}, z_{M_j}] = [1.0, 0.0, 0.0, 0.0]$ and construct $[\hat{z}_{M_i}, \hat{z}_{M_j}] = h([z_{M_i}, z_{M_j}]) = [0.7, 0.0, 0.5, 0.0]$. We can also let $[z_{M_i}, z_{M_j}] = [1.0, 1.0, 0.0, 0.0]$ and construct $[\hat{z}_{M_i}, \hat{z}_{M_j}] = h([z_{M_i}, z_{M_j}]) = [0.0, 1.0, 1.0, 0.0]$. The existence of such \hat{z} violates the structure constraint. Thus, h need to map each \mathcal{M}_{M_k} to disjoint $\hat{\mathcal{M}}_{M_k}$ to make the structure constraint hold.

Additionally, if h entangles \mathcal{M}_S and \mathcal{M}_{M_i} for any valid i , there exist two cases: 1) h maps \mathcal{M}_S to $\hat{\mathcal{M}}_S$ and $\hat{\mathcal{M}}_{M_i}$ or 2) h maps \mathcal{M}_{M_i} to $\hat{\mathcal{M}}_S$ and $\hat{\mathcal{M}}_{M_i}$. For the first case, there exist a $j \neq i$ such that $z_{M_i} = \mathbf{0}$ and $z_{M_j} \neq \mathbf{0}$ but $h(z)_{M_i} = \hat{z}_{M_i} \neq \mathbf{0}$ and $h(z)_{M_j} = \hat{z}_{M_j} \neq \mathbf{0}$. This is because z_{M_S} is independent from all the z_{M_k} and can push \hat{z}_{M_i} to arbitrary value when z_{M_i} is fixed. Thus, the structure constraint is violated. For the second case, we can apply the proof of the first case in the opposite direction using the assumption that h is smooth and invertable. we can find a \hat{z}_{M_S} such that $h^{-1}(\hat{z})_{M_i} = z_{M_i} \neq \mathbf{0}$. In the meanwhile, we can again find a \hat{z}_{M_j} which makes

$h^{-1}(\hat{z})_{M_j} = z_{M_j} \neq 0$. This contradicts the structure constraint. Thus, h needs to map \mathcal{M}_{M_S} to $\hat{\mathcal{M}}_{M_S}$, which is disjoint from all the $\hat{\mathcal{M}}_{M_k}$, to keep the structure constraint hold.

Furthermore, if h entangles multiple mechanisms, we can find two mechanisms among the entangled mechanisms to show that h violates the structure constraint. \square

C IMPLEMENTATION DETAILS

Network Architecture We develop the architectures base on ClusterGAN (Mukherjee et al., 2019). We report the network architectures for 1x28x28, 3x32x32, and 3x64x64 images in Table 6, 7, and 8, respectively. We use the same generator/decoder and encoder (with modified output layer) for VAEs. For the colored dataset like SVHN, we use a separate encoder, which takes the gray-scale image is input and predicts \hat{z}_C and \hat{z}_M . Such configuration would avoid clustering by color.

Tuning Hyper-parameters For the ICM model, we use the same hyper-parameters as the ClusterGAN (Mukherjee et al., 2019) except for β_C and β_r , which are set to 30 and 0.05, respectively, on MNIST and FashionMNIST. We train our ICM model for 200 epochs, which is sufficient for MNIST and FashionMNIST datasets. For each β -VAE and Ada-GVAE model, we train the model for 200 epochs with $\beta = [1, 2, 4, 6, 16]$. We train each generative model three times and report the best result. We use Adam optimizer with learning rate = 0.001 and batch size = 64 across the experiments. We set $(\beta_1, \beta_2) = (0.5, 0.9)$ for ICM and set $(\beta_1, \beta_2) = (0.9, 0.999)$ for VAEs. We train the VAEs with dimension [10, 20, 30, 50]. We find the VAEs with 20 dimensions have the best robustness when β is optimal. For the ICM model, we set the dimension of each ICM-conditioned subspace to 1 and 2 on MNIST and FashionMNIST, respectively. We set the dimension of the shared subspace to 4 and 5 on MNIST and FashionMNIST, respectively. Additionally, we apply the historically averaging (Salimans et al., 2016) with image buffer size = $50 \times$ batch size as the CycleGAN (Zhu et al., 2017) does.

D MORE EXPERIMENTAL RESULTS

D.1 INTERVENTIONAL ROBUSTNESS AND LATENT SPACE TRAVERSAL

MNIST We visualize the shared mechanism subspace traversal in Figure 5.

FashionMNIST We visualize the traversal for each generative mechanism subspace and the shared mechanism subspace in Figure 6.

SVHN We visualize the traversal for the generative mechanism subspaces and two dimensions in the shared mechanisms subspace in Figure 7. We note that the digits in SVHN do not have as many intra-class variations as the MNIST. The possible reason is that the house numbers are produced with the same template. Interestingly, the ICM model treat the contrast of the image as cause for the generative mechanism.

CIFAR10 We visualize the traversal for the generative mechanism subspaces and one dimension in the shared mechanism subspace in Figure 8. The traversal may not show meaningful images on CIFAR10. A possible reason is that the data samples of CIFAR10 do not lie on a continuous manifold. Another possible reason is that the data are not sampled in such a way that we can observe continuous variations from them. Despite such an issue, we can still tell the type of variations from the traversal.

Disentangled Datasets The disentangled datasets, such as 3DShapes, assume the latent explanatory factors are independent of each other. Such an assumption is not compatible with our unconventional mixture prior. In the experiment, we find our method may treat the entangled height and shape as cause for the generative mechanisms. We tried to create a dataset that correlates one type of variation (e.g. floor hue) with one shape. However, such a configuration will make the data too small to use.

Table 6: Generator, Discriminator, and Encoder Architectures for $1 \times 28 \times 28$ Inputs

Generator	Discriminator	Encoder
Input: $\mathbb{R}^{\dim(\mathbf{z})}$	Input: $\mathbb{R}^{1 \times 28 \times 28}$	Input: $\mathbb{R}^{1 \times 28 \times 28}$
FC 1024 BN ReLU	4×4 conv, 64 LReLU, stride 2	4×4 conv, 64 LReLU, stride 2
FC 128×7×7 BN ReLU	4×4 conv, 128 LReLU, stride 2	4×4 conv, 128 LReLU, stride 2
4×4 upconv, stride 2, 64 BN ReLU	FC 1024 LReLU	FC 1024 LReLU
4×4 upconv, stride 2, 1 Sigmoid	FC 1	FC $\dim(\mathbf{z})$

Table 7: Generator, Discriminator, and Encoder Architectures for $3 \times 32 \times 32$ Inputs

Generator	Discriminator	Encoder
Input: $\mathbb{R}^{\dim(\mathbf{z})}$	Input: $\mathbb{R}^{3 \times 32 \times 32}$	Input: $\mathbb{R}^{3 \times 32 \times 32}$
FC 1024 BN ReLU	4×4 conv, 64 LReLU, stride 2	4×4 conv, 64 LReLU, stride 2
FC 128×8×8 BN ReLU	4×4 conv, 128 LReLU, stride 2	4×4 conv, 128 LReLU, stride 2
4×4 upconv, stride 2, 64 BN ReLU	FC 1024 LReLU	FC 1024 LReLU
4×4 upconv, stride 2, 3 Sigmoid	FC 1	FC $\dim(\mathbf{z})$

Table 8: Generator, Discriminator, and Encoder Architectures for $3 \times 64 \times 64$ Inputs

Generator	Discriminator	Encoder
Input: $\mathbb{R}^{\dim(\mathbf{z})}$	Input: $\mathbb{R}^{3 \times 64 \times 64}$	Input: $\mathbb{R}^{3 \times 64 \times 64}$
FC 1024 BN ReLU	4×4 conv, 64 LReLU, stride 2	4×4 conv, 64 LReLU, stride 2
FC 128×8×8 BN ReLU	4×4 conv, 64 LReLU, stride 2	4×4 conv, 64 LReLU, stride 2
4×4 upconv, stride 2, 64 BN ReLU	4×4 conv, 128 LReLU, stride 2	4×4 conv, 128 LReLU, stride 2
4×4 upconv, stride 2, 64 BN ReLU	FC 1024 LReLU	FC 1024 LReLU
4×4 upconv, stride 2, 3 Sigmoid	FC 1	FC $\dim(\mathbf{z})$

Table 9: Shift Distance Needed for 20% Relative Accuracy Drop under covariant Shift

MODEL	MNIST (T)	MNIST (W&R)	MNIST (R)	FASHIONMNIST (D&W)
ICM	2.0	1.2	0.8	0.7
VAE	1.2	0.7	0.7	0.4
β -VAE	1.7	0.8	0.7	0.5
ADA-GVAE	1.6	0.9	0.7	0.5

D.2 ROBUSTNESS UNDER COVARIANT SHIFT

We report the accuracy change of all the evaluated models under covariant shift on MNIST, FashionMNIST and SVHN in Figures 9, 10, 11, 12, 13. We report the shift distance needed for 20% and 40% relative accuracy drop in Tables 9 and 10, respectively. On SVHN dataset, our ICM model is slightly worse than a few other methods even if the covariant shift is small. However, as the shift distance increases, our method starts to narrow the gap. The possible reason is that our encoder takes the output of the generator as the input. On a complex dataset, the generator may not approximate the data distribution perfectly. Thus, if we use the test data as the input, some of the input may become out-of-distribution for the encoder. This is why there seemingly exist negative offsets of the lines in Figure 13(b). Despite the issue, these results still suggest that our ICM model is more robust against the covariant shift before the test sets shift too far away.

Table 10: Shift Distance Needed for 40% Relative Accuracy Drop under covariant Shift

MODEL	MNIST (T)	MNIST (W&R)	MNIST (R)	FASHIONMNIST (D&W)
ICM	2.6	1.6	1.1	1.2
VAE	1.9	1.3	1.1	0.8
β -VAE	2.4	1.5	1.3	1.1
ADA-GVAE	2.2	1.5	1.2	1.0

Table 11: Accuracy of Downstream Classifiers under Noise Uniform($\mathbf{0}, \mathbf{I}$)

MODEL	MNIST (T)	MNIST (W&R)	MNIST (R)	FASHIONMNIST (D&W)
ICM	61.72% \pm 2.30%	33.14% \pm 4.65%	64.65% \pm 2.30%	66.13% \pm 1.29%
VAE	41.62% \pm 4.53%	34.66% \pm 3.20%	46.33% \pm 1.97%	19.32% \pm 2.13%
β -VAE	43.46% \pm 2.03%	37.30% \pm 1.23%	45.78% \pm 3.41%	24.88% \pm 2.68%
ADA-GVAE	51.27% \pm 3.75%	34.58% \pm 3.37%	43.51% \pm 4.39%	18.43% \pm 2.26%

D.3 ROBUSTNESS UNDER NOISE

We show the robustness under Uniform($\mathbf{0}, \mathbf{I}$) noise in Table 11. The result is slightly better compared to the Gaussian noise experiment as Section 5.3 shows. For the FashionMNIST (D&W) dataset, the advantage of our method goes beyond our expectations.

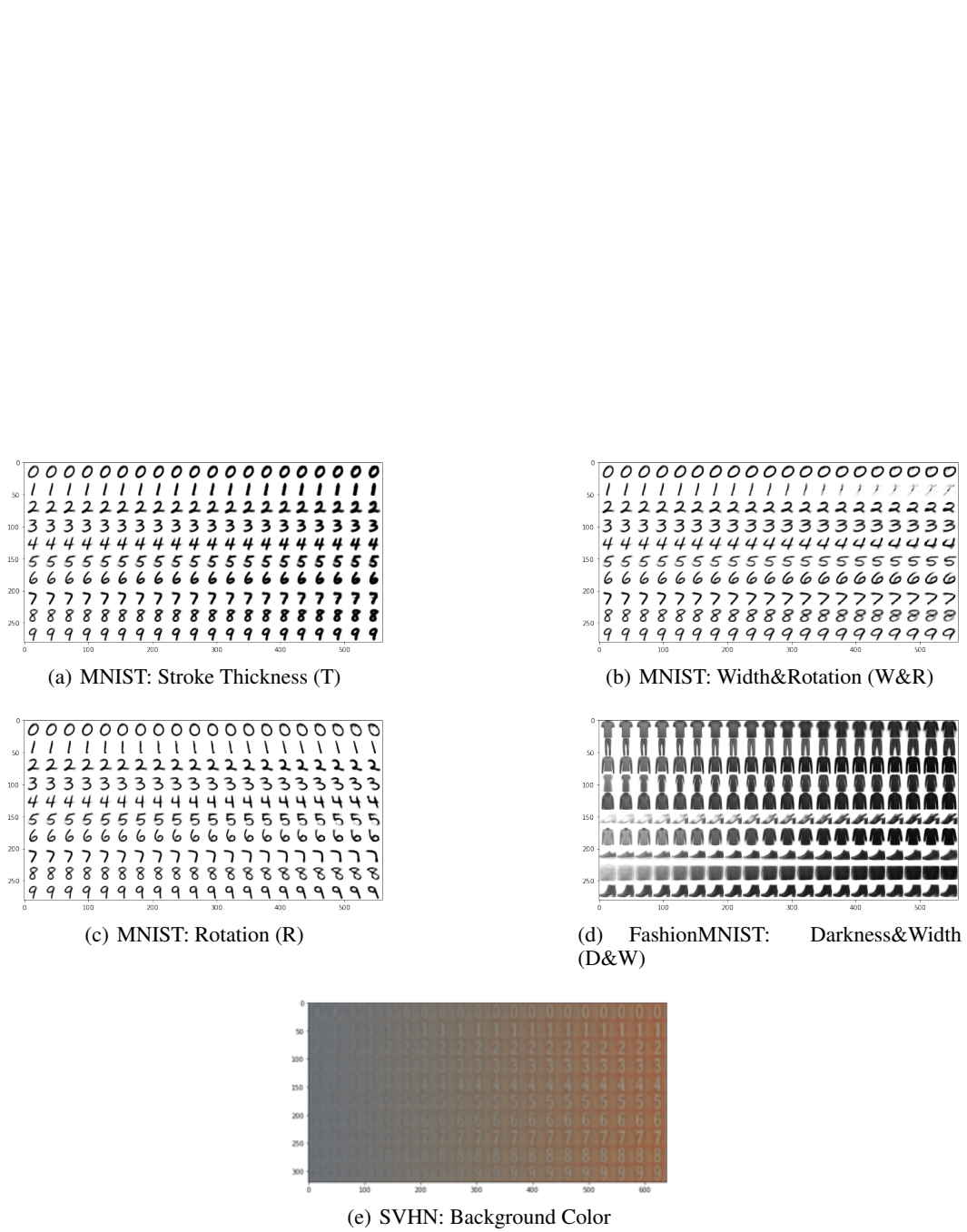
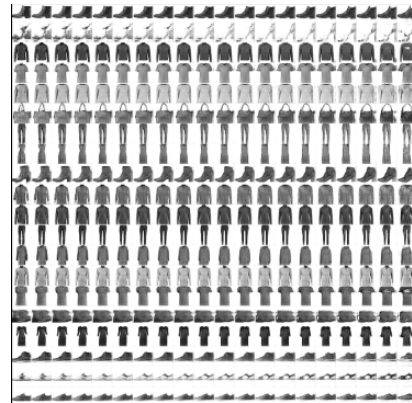


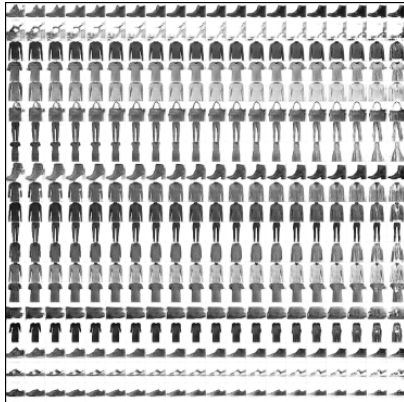
Figure 4: Covariate Used in the Covariate Shift Experiment



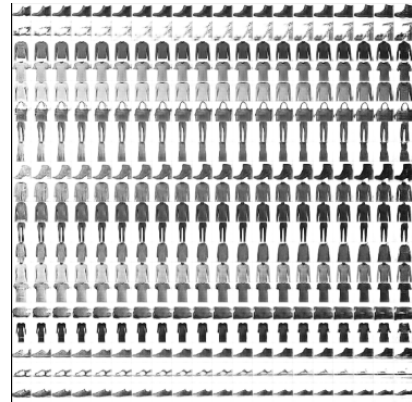
(a) Generative Mechanisms Subspaces



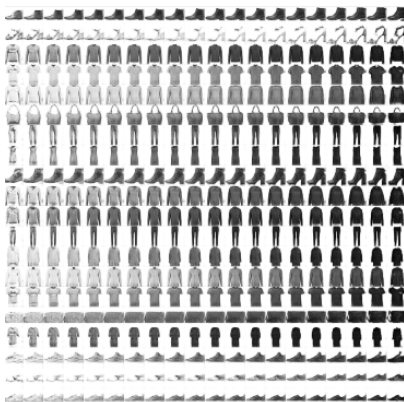
(b) Shared Mechanism Subspace, Dimension #0



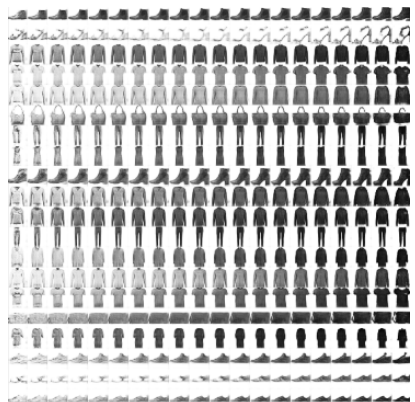
(c) Shared Mechanism Subspace, Dimension #1



(d) Shared Mechanism Subspace, Dimension #2



(e) Shared Mechanism Subspace, Dimension #3



(f) Shared Mechanism Subspace, Dimension #4

Figure 6: Latent Space Traversal of ICM model on FashionMNIST

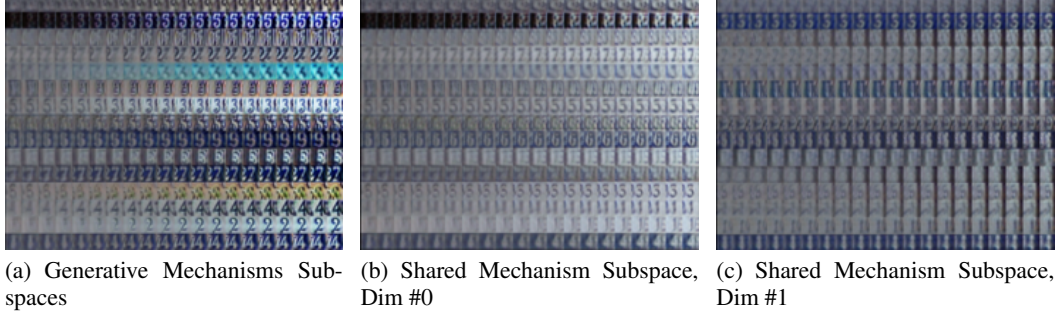


Figure 7: Latent Space Traversal of ICM model on SVHN

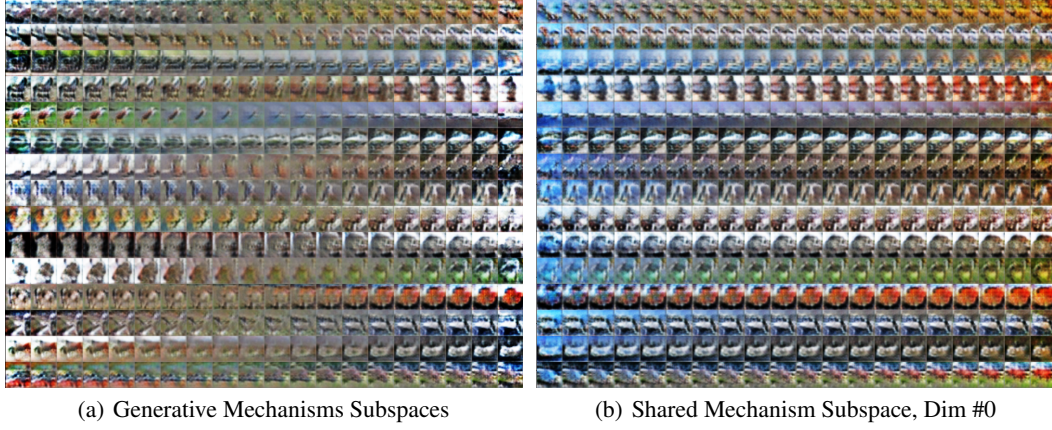


Figure 8: Latent Space Traversal of ICM model on CIFAR10

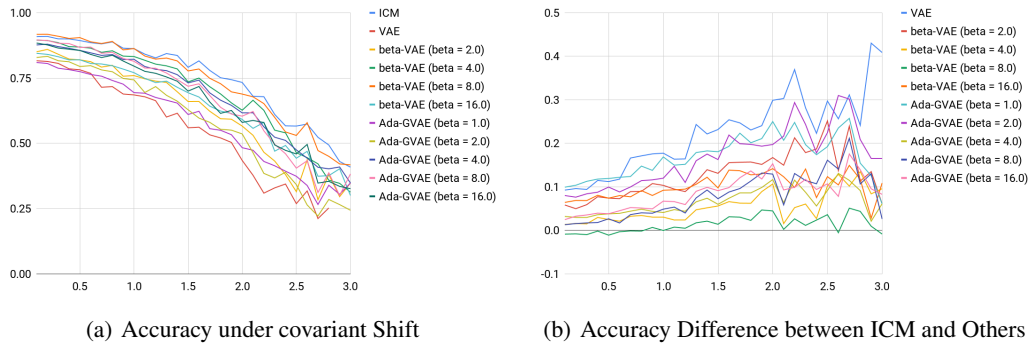


Figure 9: Covariate Shift (Stroke Thickness) on MNIST

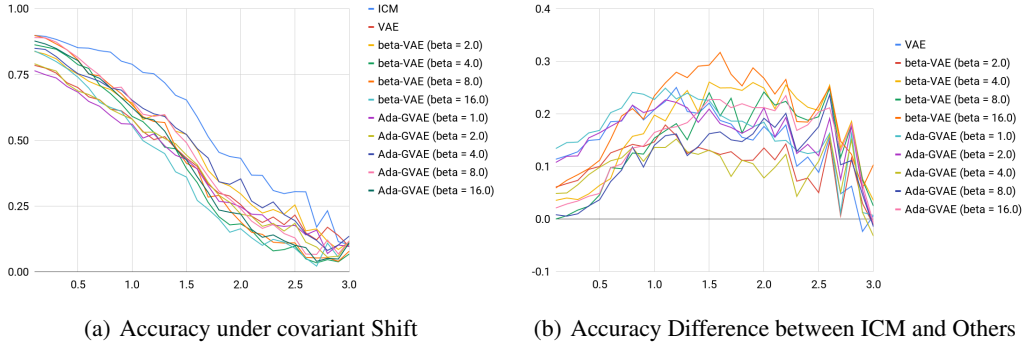


Figure 10: Covariate Shift (Width&Rotation) Experiment on MNIST

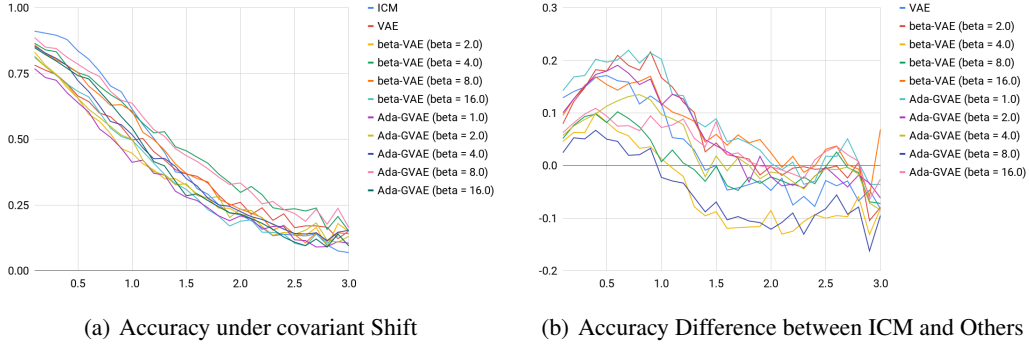


Figure 11: Covariate Shift (Rotation) Experiment on MNIST

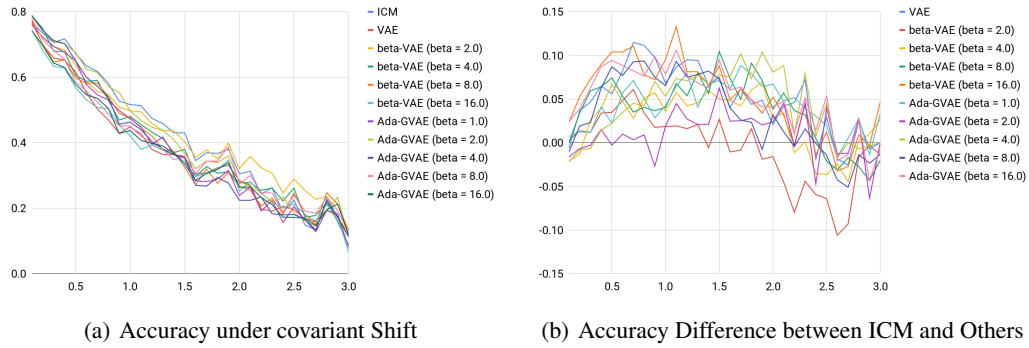


Figure 12: Covariate Shift (Darkness&Width) Experiment on FashionMNIST

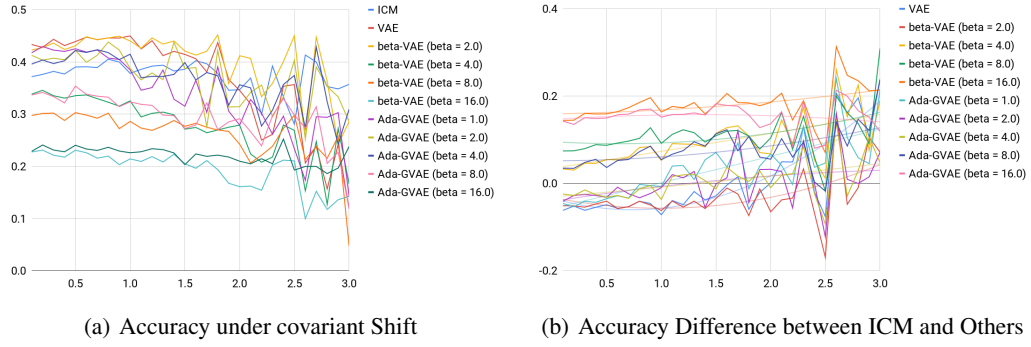


Figure 13: Covariate Shift (Background Color) Experiment on SVHN