Achieving Robustness in the Wild via Adversarial Mixing with Disentangled Representations

Sven Gowal*, Chongli Qin*, Po-Sen Huang, Taylan Cemgil,
Krishnamurthy (Dj) Dvijotham, Timothy Mann, Pushmeet Kohli

CVPR’20

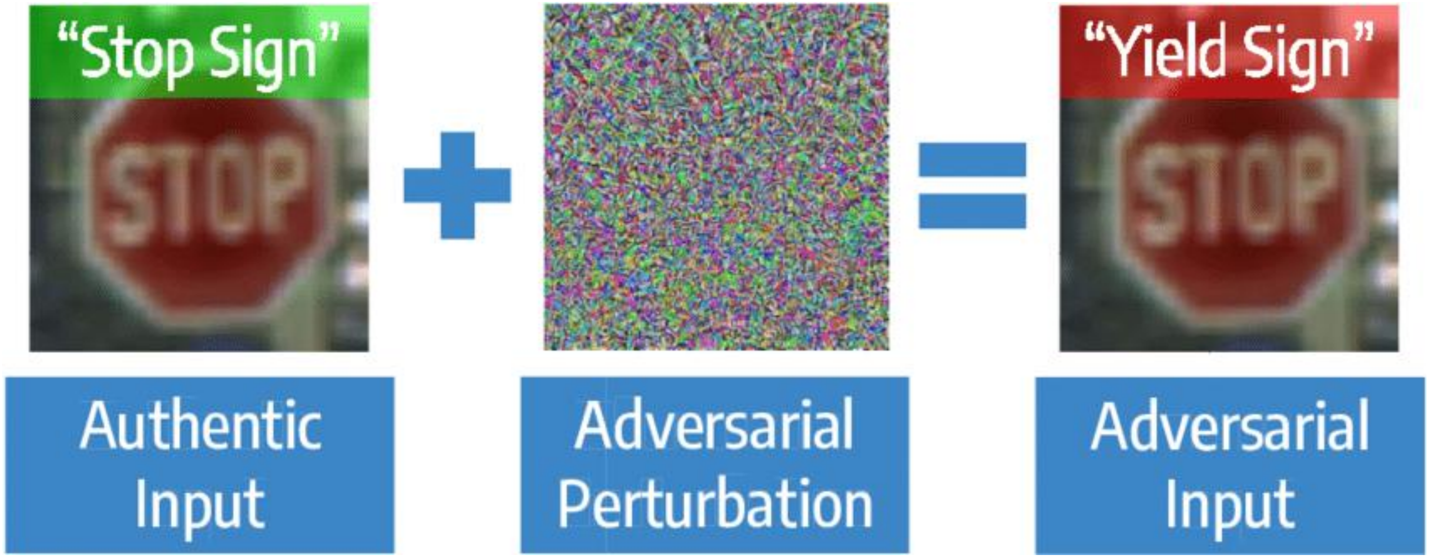
Presented by Eungyeup Kim

Vision Study
22 FEB 2021

Backgrounds

Perturbation-based Adversarial Vulnerability

- ϵ-perturbation leads our model to misclassify while maintaining its class-specific concept.
- Neural network is *too sensitive* to task-irrelevant *small* changes of their input.



https://medium.com/@sach.jbp/a-review-on-adversarial-attacks-on-machine-learning-algorithms-8eab0d3734ce

Backgrounds

Plausible Real-world Adversarial Vulnerability

- Plausible real-world perturbation leads our model to misclassify while preserving its semantic content.
- Neural network is *too sensitive* to task-irrelevant *less-semantic* changes of their input.

Change
skin-tone

"Smiling" : 98%

"Smiling" : 56%

Introduction

AdvMix

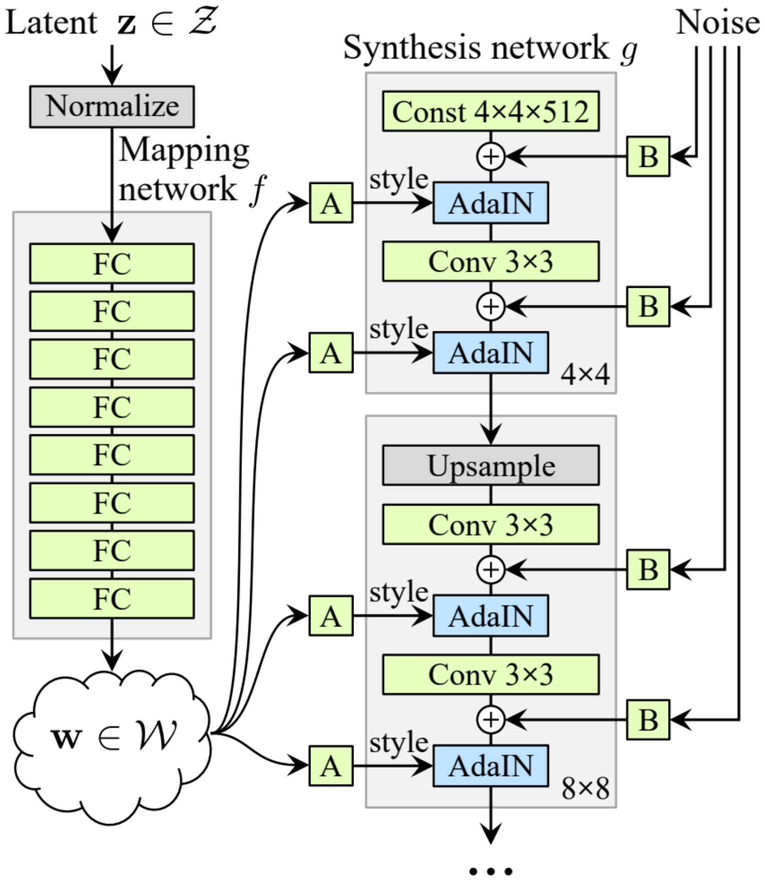
- This paper focuses on training model robust to plausible real-world perturbations that preserve semantic contents.

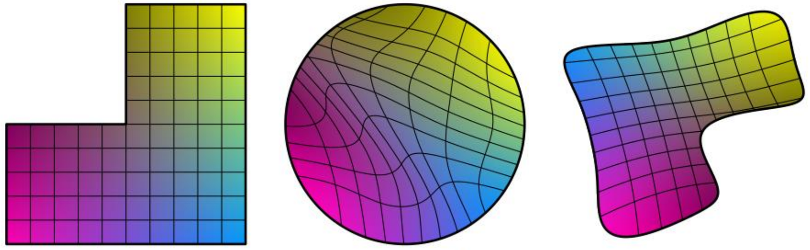
- Leveraging *StyleGAN* can enable us to conduct data augmentation beyond l_p norm bounded perturbation.

- The authors propose a framework dubbed *Adversarial Mixing with Disentangled Representations* (AdvMix), which systematically transfers non-robust attributes via StyleGAN's mixing property.

Backgrounds

StyleGAN





(a) Distribution of features in training set

(b) Mapping from $\mathcal{Z}$ to features

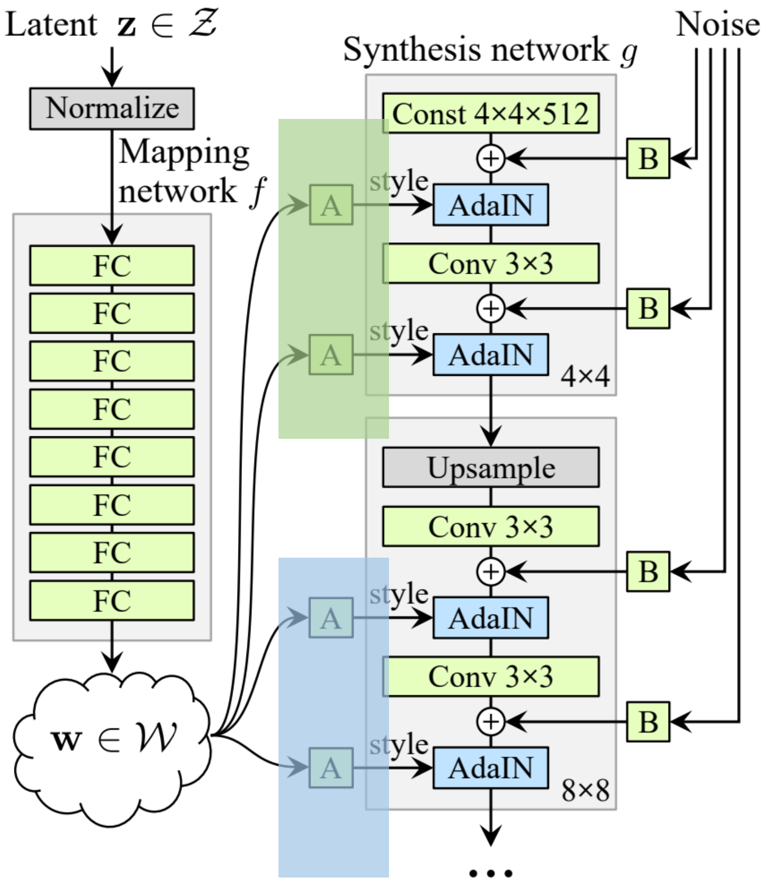
(c) Mapping from $\mathcal{W}$ to features

Figure 6. Illustrative example with two factors of variation (image features, e.g., masculinity and hair length). (a) An example training set where some combination (e.g., long haired males) is missing. (b) This forces the mapping from $\mathcal{Z}$ to image features to become curved so that the forbidden combination disappears in $\mathcal{Z}$ to prevent the sampling of invalid combinations. (c) The learned mapping from $\mathcal{Z}$ to $\mathcal{W}$ is able to "undo" much of the warping.

- Mapping network enables the sampling of latent from more linear embedding space, rather than fixed distribution.
- Generator encourages this as it should be easier to generate realistic images based on a disentangled representation, rather than entangled one.

Backgrounds

StyleGAN



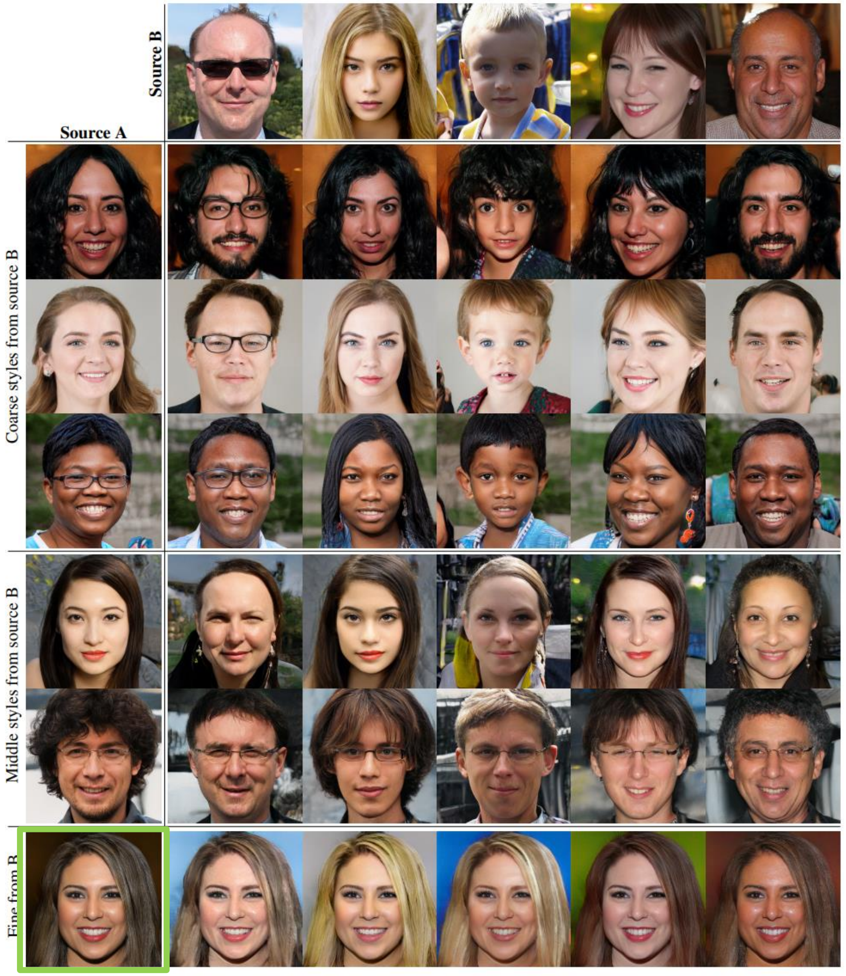


Figure 3. Two sets of images were generated from their respective latent codes (sources A and B); the rest of the images were generated b copying a specified subset of styles from source B and taking the rest from source A. Copying the styles corresponding to coarse spatia resolutions ($4^2 - 8^2$) brings high-level aspects such as pose, general hair style, face shape, and eyeglasses from source B, while all colo (eyes, hair, lighting) and finer facial features resemble A. If we instead copy the styles of middle resolutions ($16^2 - 32^2$) from B, we inher smaller scale facial features, hair style, eyes open/closed from B, while the pose, general face shape, and eyeglasses from A are preserve Finally, copying the fine styles ($64^2 - 1024^2$) from B brings mainly the color scheme and microstructure.

Introduction

Label-invariant Latent Factor

- We assume that we have an ideal generator (or decoder), $dec: Z \mapsto X$, where the latent space Z is a product space of the form $Z = Z_{||} \times Z_{\perp}$.
- For a given classification task that predicts the label y, only the coordinates corresponding to $Z_{||}$ are relevant, while $Z_{\perp}$ is irrelevant:

$$\mathbb{P}(y|z_{||}, z_{\perp}) = \mathbb{P}(y|z_{||})$$

- Hence, the ideal invariant classifier f^* should be consistent with the invariance assumption:

$$f^\star(\mathtt{dec}(z_{||}, z_{\perp})) = f^\star(\mathtt{dec}(z_{||}, \tilde{z}_{\perp}))$$

$$\underset{y' \in \mathcal{Y}}{\operatorname{argmax}}\, f^\star(\mathtt{dec}(z_{||}, z_{\perp})) = y$$

Introduction

Adversarial Training with Semantically Irrelevant Perturbations

- We define the set of transformations T that induce semantically irrelevant perturbations as:

$$\mathcal{T} = \{t \mid t(x) = \mathtt{dec}(z_{\|}, \tilde{z}_{\perp}) \text{ with } \tilde{z}_{\perp} \in \mathcal{Z}_{\perp} \text{ s.t. } \exists z_{\perp} x = \mathtt{dec}(z_{\|}, z_{\perp})\}.$$

- Our goal is to find the model parameters θ that minimize the *semantic adversarial risk*

$$\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \left[\max_{t \in \mathcal{T}} L(f_\theta(t(x)), y) \right]$$

where D is a data distribution and L is a suitable loss function.

- Therefore,

$$\theta^\star = \operatorname*{argmin}_{\theta} \mathop{\mathbb{E}}_{\substack{(x,y)\sim\mathcal{D} \\ x=\mathtt{dec}(z_{\|}, z_{\perp})}} \left[\max_{\tilde{z}_{\perp} \in \mathcal{Z}_{\perp}} L(f_\theta(\mathtt{dec}(z_{\|}, \tilde{z}_{\perp})), y) \right]$$

Introduction

Adversarial Training with Semantically Irrelevant Perturbations

- Solving the saddle point problem requires solving the corresponding inner-maximization problem:

$$\tilde{z}_\perp^\star = \operatorname*{argmax}_{\tilde{z}_\perp \in \mathcal{Z}_\perp} L(f_\theta(\mathtt{dec}(z_\parallel, \tilde{z}_\perp)), y)$$

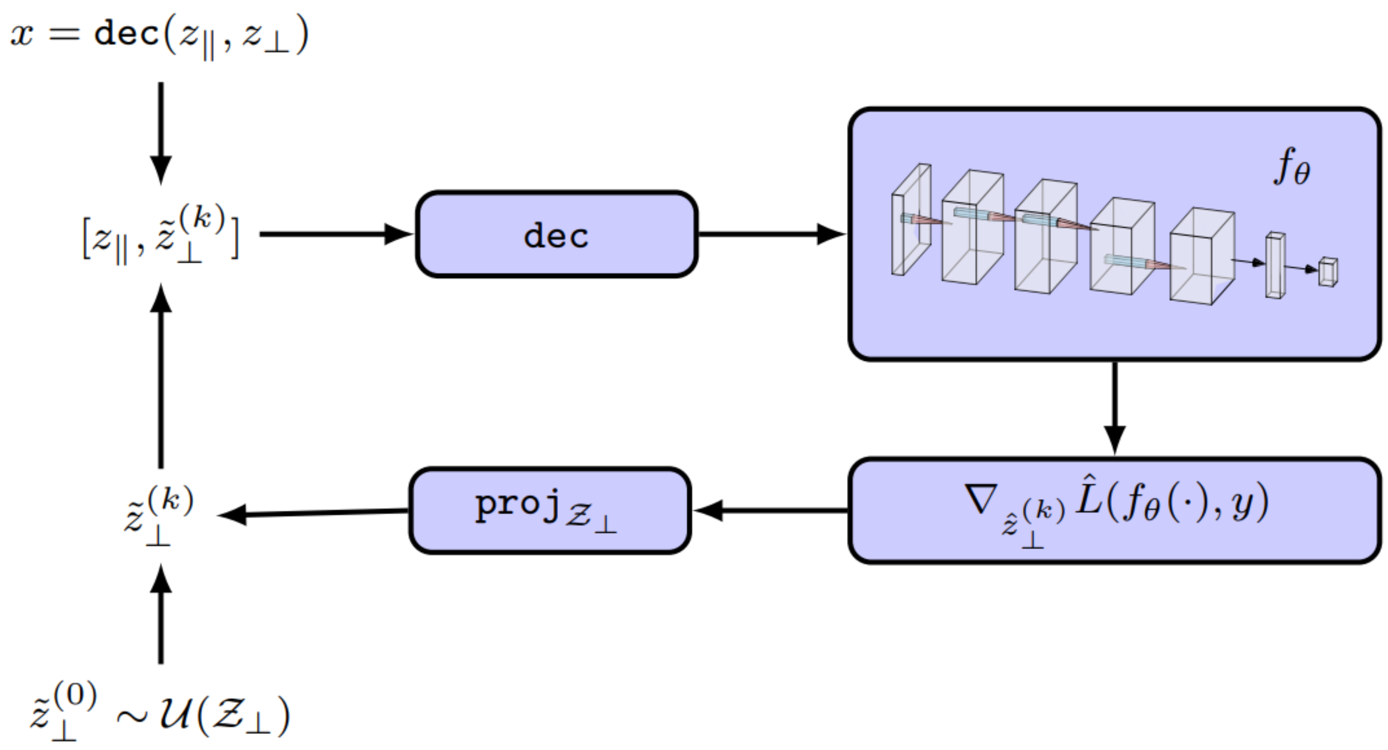
- Rather than enumerating all possible latent $\tilde{z}_\perp \in Z_\perp$, this paper utilizes projected gradient ascent on a cross-entropy loss.

$$\hat{L}(f_\theta(x), y) = -\log([f_\theta(x)]_y)$$

$$\tilde{z}_\perp^{(k+1)} = \mathtt{proj}_{\mathcal{Z}_\perp}\left(\tilde{z}_\perp^{(k)} + \alpha \nabla_{\tilde{z}_\perp^{(k)}} \hat{L}(f_\theta(\mathtt{dec}(z_\parallel, \tilde{z}_\perp^{(k)})), y)\right)$$

Introduction

Adversarial Training with Semantically Irrelevant Perturbations



Introduction

Why StyleGAN?

- As *AdvMix* need disentangled latents z, this paper heavily relies on *StyleGAN*'s *mixing property* to enforce a partitioning of the latents.

- Style mixing of *StyleGAN* can be applied via coarse spatial resolutions corresponding to the high-level features, and finer resolutions corresponding to the low-level features, such as color scheme.

- In this paper, the authors assume that fine attribute $z_\perp$ corresponds to the label-independent style.

Introduction

Construction of a dataset of disentangled latents

- As we want to obtain disentangled latents $z = [z_{||}, z_{\perp}]$ from image x, this paper construct the dataset D using algorithm 1 as below:

Algorithm 1 Encoder `enc`

Input: Target image x, trained *StyleGAN* model $\texttt{dec} \circ \texttt{map}$, and trained VGG network `vgg`. α_i and β_i are hyperparameters all set to 1 and 1/5 respectively. $\gamma^{(k)}$ is a step-size schedule.

Output: Disentangled latents $\hat{z}$ such that $\texttt{dec}(\hat{z}) \approx x$

$\hat{z} \leftarrow \frac{1}{M} \sum_{i=1}^{M} \texttt{map}(\mathbf{z}^{(i)})$ with $\mathbf{z}^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$ ▷ Average latents

for $k \in \{1, \ldots, N\}$ **do** ▷ N is the number of iterations

    $\hat{x} = \texttt{dec}(\hat{z})$

    $\hat{\mathcal{A}} = \texttt{vgg}(\hat{x})$ ▷ $\hat{\mathcal{A}}$ is a list of activations (after the 2nd convolution of 1st, 2nd and 3rd blocks)

    $\mathcal{A} = \texttt{vgg}(x)$

    $\mathcal{A}_{\text{mix}} = \texttt{vgg}(\texttt{dec}(\hat{z}_{||}, \texttt{map}(\mathbf{z})_{\perp})))$ with $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$

    $L_{\text{reconstruct}} = \alpha_0 \|\hat{x} - x\|_2^2 + \sum_{i=1}^{|\mathcal{A}|} \alpha_i \|\hat{\mathcal{A}}_i - \mathcal{A}_i\|_2^2$ ▷ Reconstruction loss

    $L_{\text{mix}} = \sum_{i=1}^{|\mathcal{A}|} \beta_i \|\mathcal{A}_{\text{mix},i} - \mathcal{A}_i\|_2^2$ ▷ Mixing loss

    $\hat{z} \leftarrow \hat{z} - \gamma^{(k)} \nabla_{\hat{z}} (L_{\text{reconstruct}} + L_{\text{mix}})$

end for

Randomly sampled latent $\hat{z}$

Disentangled latents $\hat{z}$ such that $dec(\hat{z}) \approx x$

Introduction

Generating worst-case examples to train robust models

- We want to minimize the *semantic adversarial risk* by relying on projected gradient ascent as below:

Algorithm 2 Solution to Equation (7)

Input: A nominal input x and label y, a model f_θ, a *StyleGAN* model $\texttt{dec} \circ \texttt{map}$ and an encoder $\texttt{enc}$. L is the $0-1$ loss and $\hat{L}$ is the cross-entropy loss.

Output: Possible misclassified example $\tilde{x}$

$\tilde{x} \leftarrow x$

$[z_{||}, z_{\perp}] = \texttt{enc}(x)$ ▷ See Algorithm 1

for $r \in \{1, \ldots, N_r\}$ **do** ▷ Repeat N_r times

  $\tilde{z}_{\perp}^{(0)} \leftarrow \texttt{map}(\mathbf{z})_{\perp}$ with $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$ ▷ Initial latents

  $\tilde{x}^{(0)} = \texttt{dec}(z_{||}, \tilde{z}_{\perp}^{(0)})$

  for $k \in \{1, \ldots, K\}$ **do** ▷ K is the number of optimization steps

    $\tilde{z}_{\perp}^{(k)} \leftarrow \texttt{proj}\left(\tilde{z}_{\perp}^{(k-1)} + \alpha \nabla_{\tilde{z}_{\perp}^{(k-1)}} \hat{L}(f_\theta(\tilde{x}^{(0)}), y)\right)$

    $\tilde{x}^{(k)} = \texttt{dec}(z_{||}, \tilde{z}_{\perp}^{(k)})$

    if $L(f_\theta(\tilde{x}^{(k)}), y) > L(f_\theta(\tilde{x}, y)$ **then**

      $\tilde{x} \leftarrow \tilde{x}^{(k)}$

      return ▷ Since L is the $0-1$ loss, the procedure can terminate early

    end if

  end for

end for

Label — Classical training	Not smiling (99.36%)	Not smiling (57.88%)
Label — AdvMix (our) training	Not smiling (95.82%)	Smiling (54.90%)

Images under transformation

Label — Classical training	Smiling (99.98%)	Smiling (100%)
Label — AdvMix (our) training	Not smiling (97.26%)	Smiling (61.34%)

Experiments

Baselines

- Standard training
- Adversarial training (*AT)*
- Random Mixing with Disentangled Representations *(RandMix):* randomly sample $z_\perp$ from $Z_\perp$, rather than systematically finding the worst-case variations.

$$\tilde{x} = \mathtt{dec}(\mathtt{enc}(x)_{||}, \mathtt{map}(\mathrm{z})_{\perp}) \text{ with } \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$$

Datasets

- Color-MNIST
- CelebA

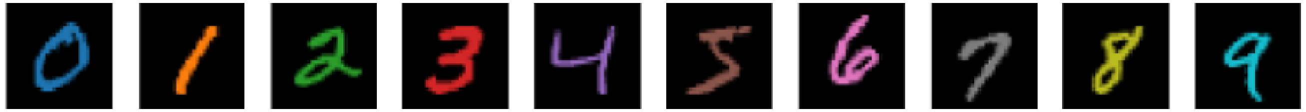
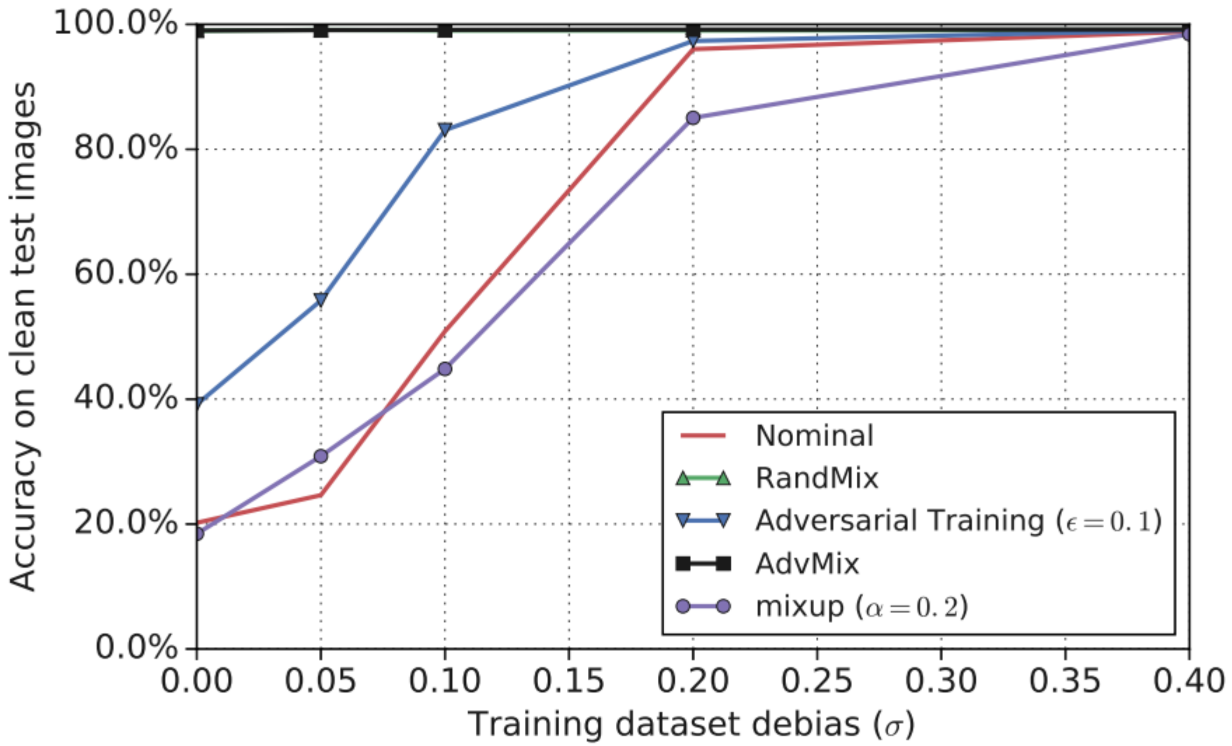
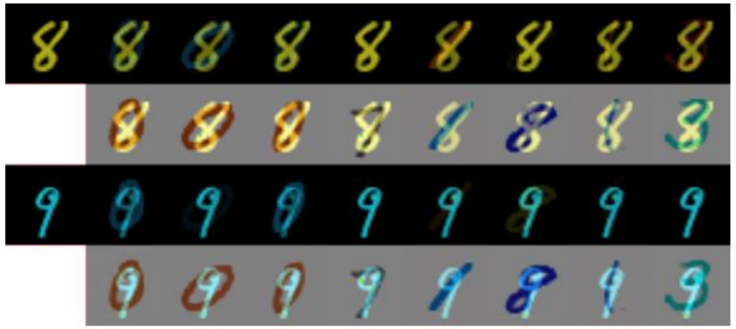


Figure 6. Mean colors given to each digit in the training set of our Color-MNIST case-study.

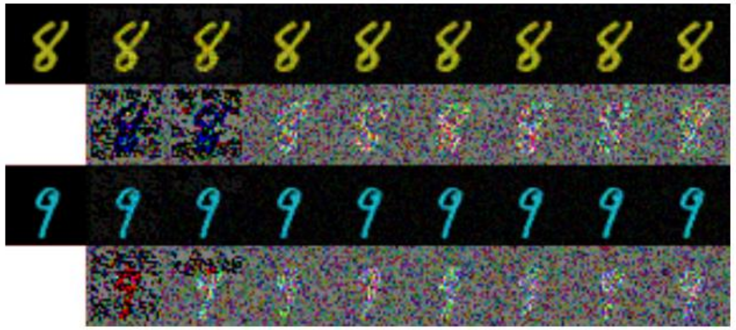
Experiments

Color-MNIST

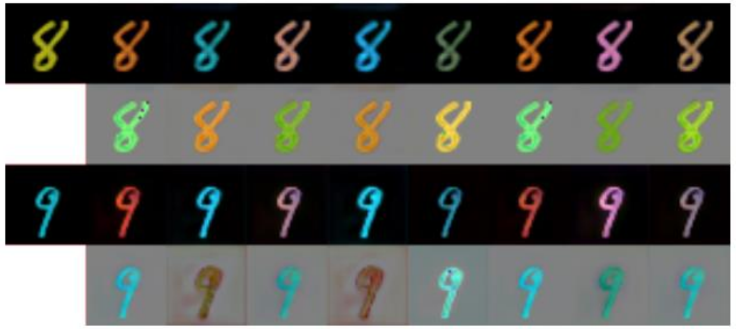




(a) *mixup*



(b) Adversarial Training ($\epsilon = 0.1$)

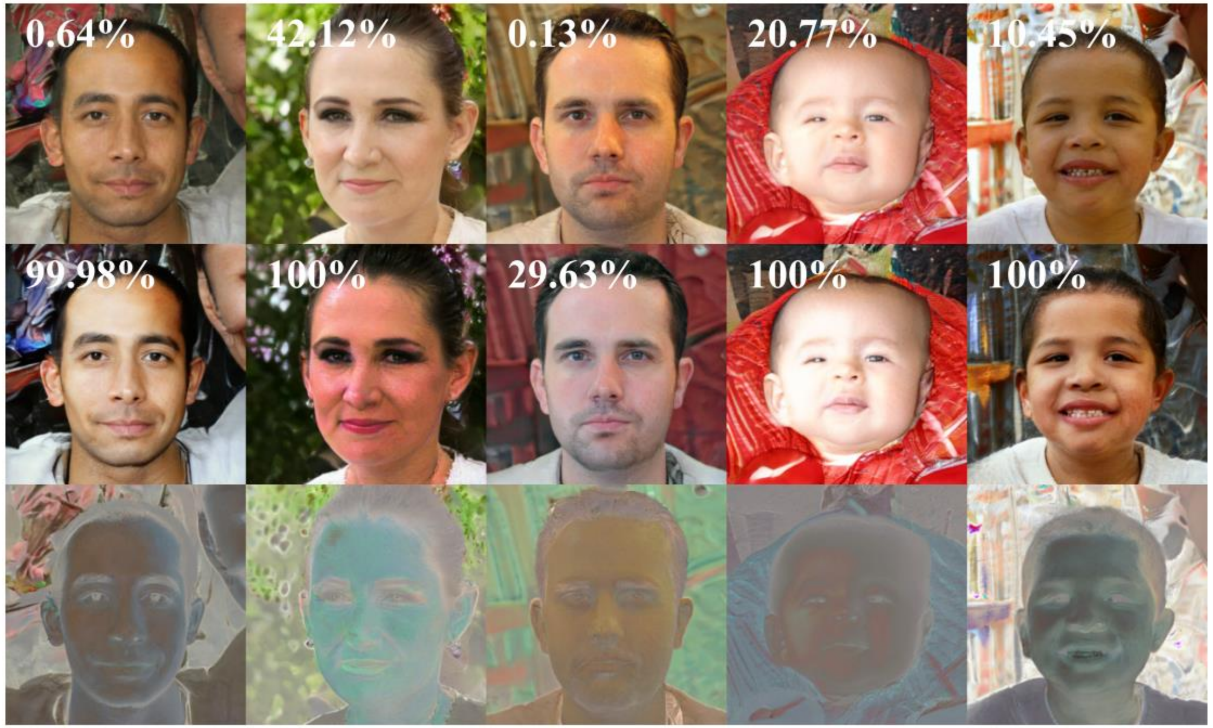


(c) *AdvMix* or *RandMix*

Experiments

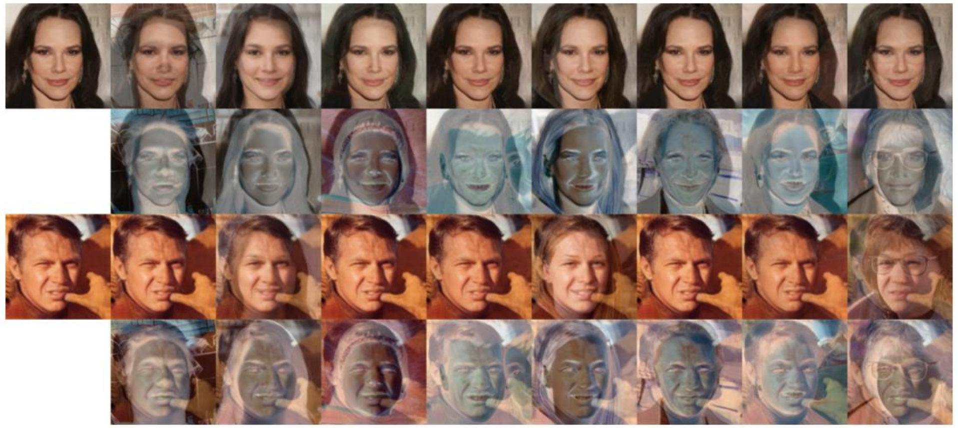
CelebA

Method	Test accuracy on attribute #1	#2 (smiling)	#3	#4
Nominal	96.49%	90.22%	83.52%	78.05%
AT ($\epsilon = 4/255$)	95.34%	91.11%	81.43%	76.61%
AT ($\epsilon = 8/255$)	95.22%	89.29%	79.46%	74.39%
RandMix	96.70%	90.36%	84.49%	76.41%
AdvMix	**97.56%**	**92.29%**	**85.65%**	**79.47%**

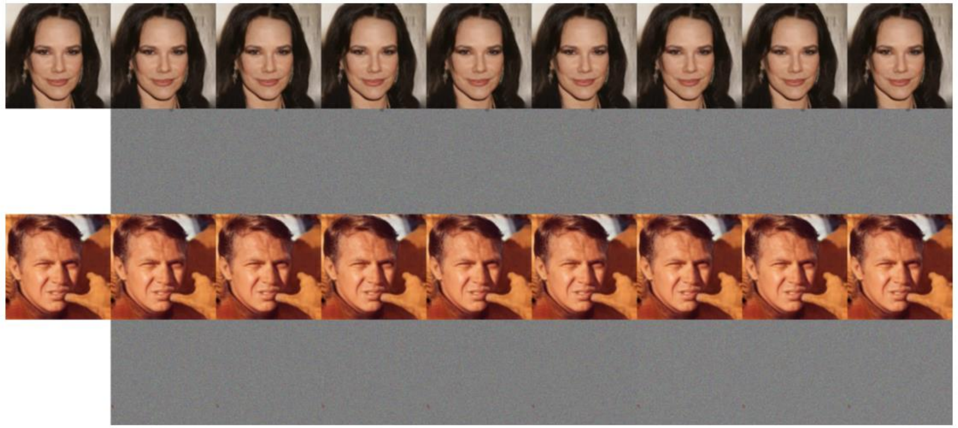


Experiments

CelebA



(a) *mixup*



(b) Adversarial Training ($\epsilon = 8/255$)

(c) *AdvMix* or *RandMix*