The Hessian Penalty: A Weak Prior for Unsupervised Disentanglement

Marin Nikita 171

Метод. Формулировка

$$G : \mathbb{R}^{|z|} \to \mathbb{R}$$

$$H_{ij} = \frac{\partial^2 G}{\partial z_i \partial z_j} = \frac{\partial}{\partial z_j}\left(\frac{\partial G}{\partial z_i}\right) = 0.$$

$$\mathcal{L}_H(G) = \sum_{i=1}^{|z|} \sum_{j \neq i}^{|z|} H_{ij}^2.$$

G - функция, возвращающая скаляр

z - входной вектор

H - Гессиан функции G по входу z.

L_H - Штраф Гессиана.

Метод. Обобщение

$$\mathcal{L}_{\mathbf{H}}(G) = \max_i \mathcal{L}_{\mathbf{H}_i}(G)$$

H_i - Гессиан функции G по i-ой координате выходного вектора x.

H - множество Гессианов G для каждого элемента x_i.

Метод на практике

$$\mathcal{L}_H(G) = \mathrm{Var}_v\left(v^T H v\right)$$

$$v^T H v \approx \frac{1}{\epsilon^2}\left[G(z+\epsilon v) - 2G(z) + G(z-\epsilon v)\right]$$

v - вектор Радемахера (каждая компонента +-1)

На практике для подсчета несмещенной выборочной дисперсии берут небольшое количество векторов v.

Для ускорения подсчетов используется аппроксимация второй производной по направлению с параметром e=0.1.

Теорема

$$Var_v\left(v^T H v\right) = 2\sum_{i=1}^{|z|}\sum_{j\neq i}^{|z|} H_{ij}^2$$

$$\operatorname{Var}(x^\intercal H x) = \operatorname{Var}\left(\sum_{i,j} H_{ij}x_i x_j\right) = \operatorname{Var}\left(\sum_i H_{ii}x_i^2 + \sum_{i\neq j} H_{ij}x_i x_j\right) = \operatorname{Var}\left(\underbrace{\sum_i H_{ii}}_{\text{constant}} + \sum_{i\neq j} H_{ij}x_i x_j\right)$$

$$= \operatorname{Var}\left(\sum_{i\neq j} H_{ij}x_i x_j\right) = \operatorname{Var}\left(2\sum_{i<j} H_{ij}x_i x_j\right) = 4\operatorname{Cov}\left(\sum_{i<j} H_{ij}x_i x_j, \sum_{i<j} H_{ij}x_i x_j\right)$$

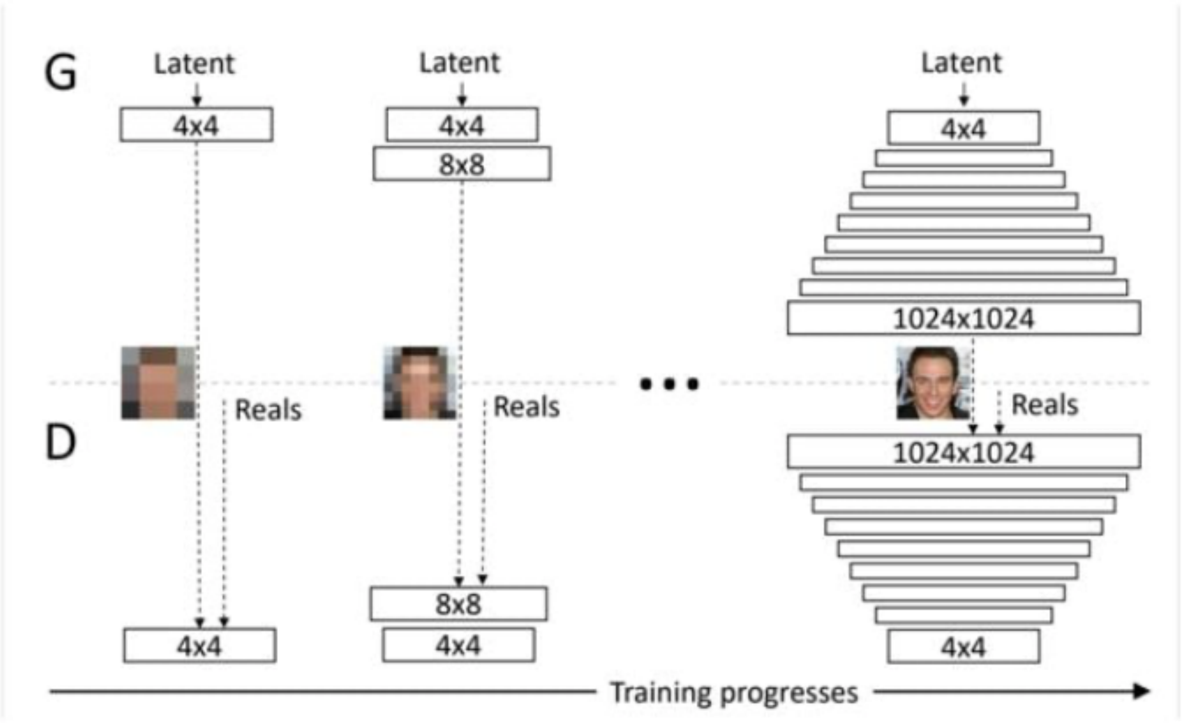
$$= 4\sum_{i<j}\operatorname{Cov}\left(H_{ij}x_i x_j, H_{ij}x_i x_j\right) = 4\sum_{i<j}\operatorname{Var}\left(H_{ij}x_i x_j\right) = 4\sum_{i<j} H_{ij}^2 \cdot 1 = 2\sum_{i\neq j} H_{ij}^2$$

Использование в GANs

$$\mathcal{L}_{\text{adv}} = \mathop{\mathbb{E}}_{x \sim p_{\text{data}}(x)} [f(D(x))] + \mathop{\mathbb{E}}_{z \sim p_z(z)} [f(1 - D(G(z)))]$$

$$\mathcal{L}_{\text{G}} = \mathop{\mathbb{E}}_{z \sim p_z(z)} [f\,(1 - D(G(z)))] + \lambda \mathop{\mathbb{E}}_{z \sim p_z(z)} [\mathcal{L}_{\mathbf{H}}(G)]$$

Эксперименты. ProGAN (Progressive growing)

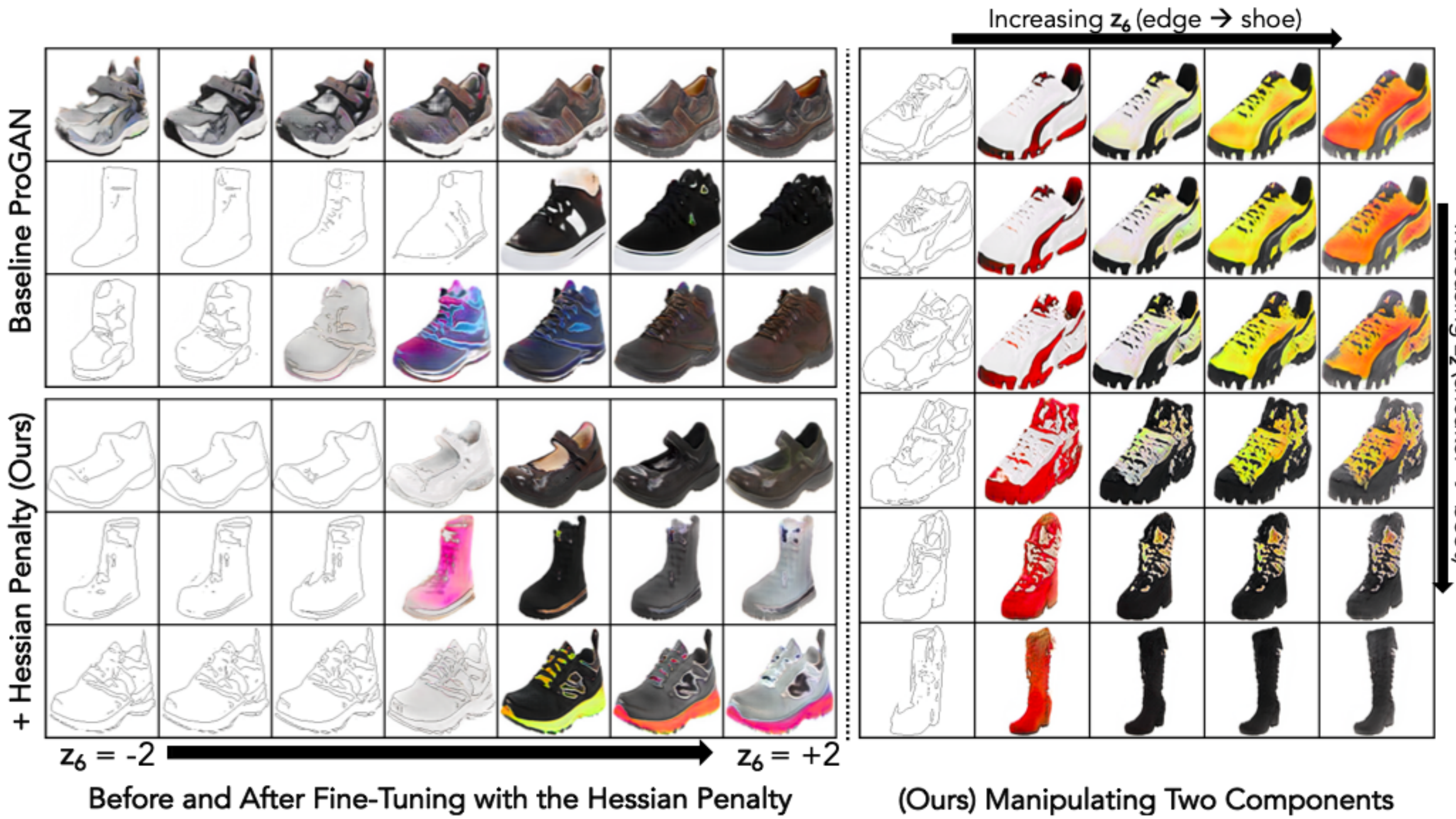


Обучение проходит следующим образом, на каждом временном шаге добавляется один слой генератора и один слой дискриминатора.

Во время одного временного шага идет обучение до почти сходимости.

! В наших примерах |z| = 12, а размер сгенерированного изображения 128 * 128.

Эксперименты. ProGAN. Edge2shoes

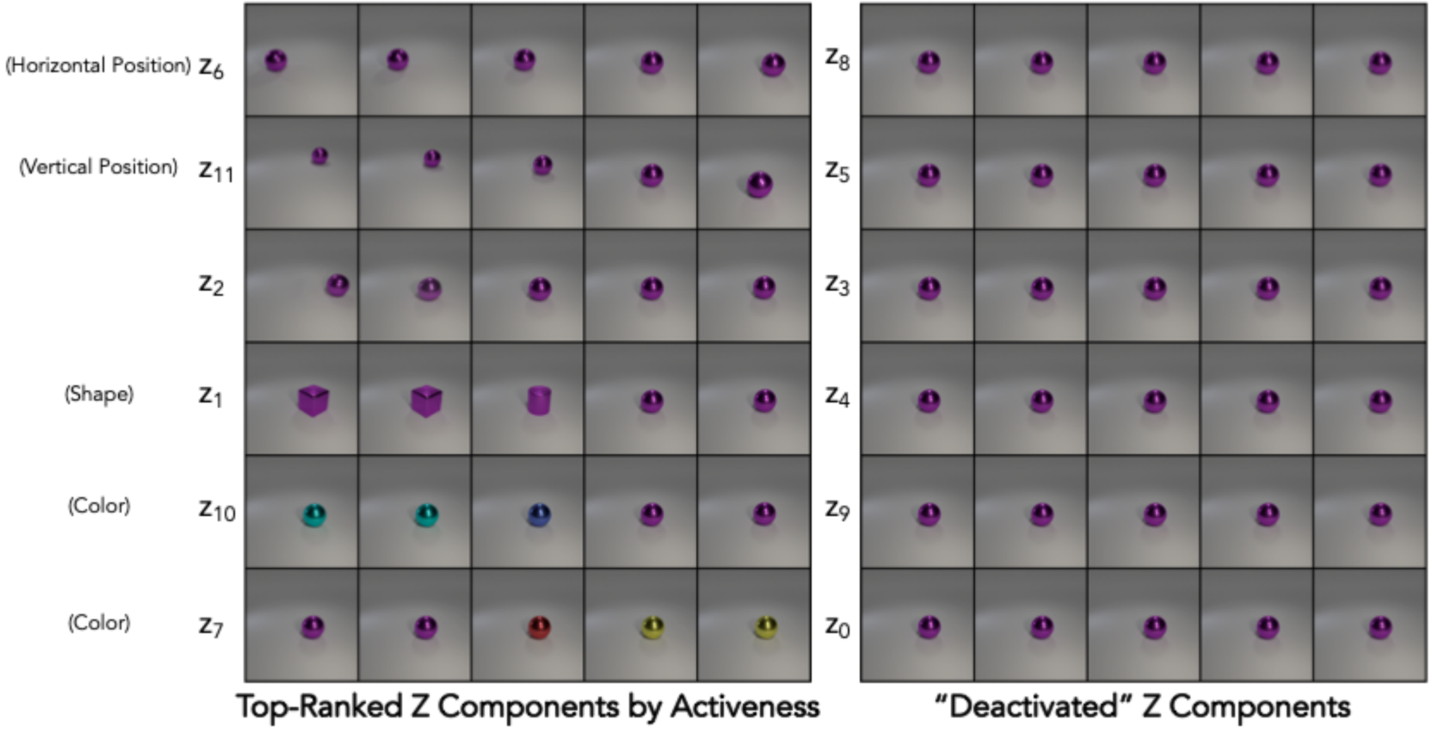


Сгенерировали три вектора z из нормального распределения (вектор = строка),

Сверху слева fine-tune без HP, снизу слева fine-tune с HP.

Справа fine-tune с HP, манипуляция сразу по двум координатам z.

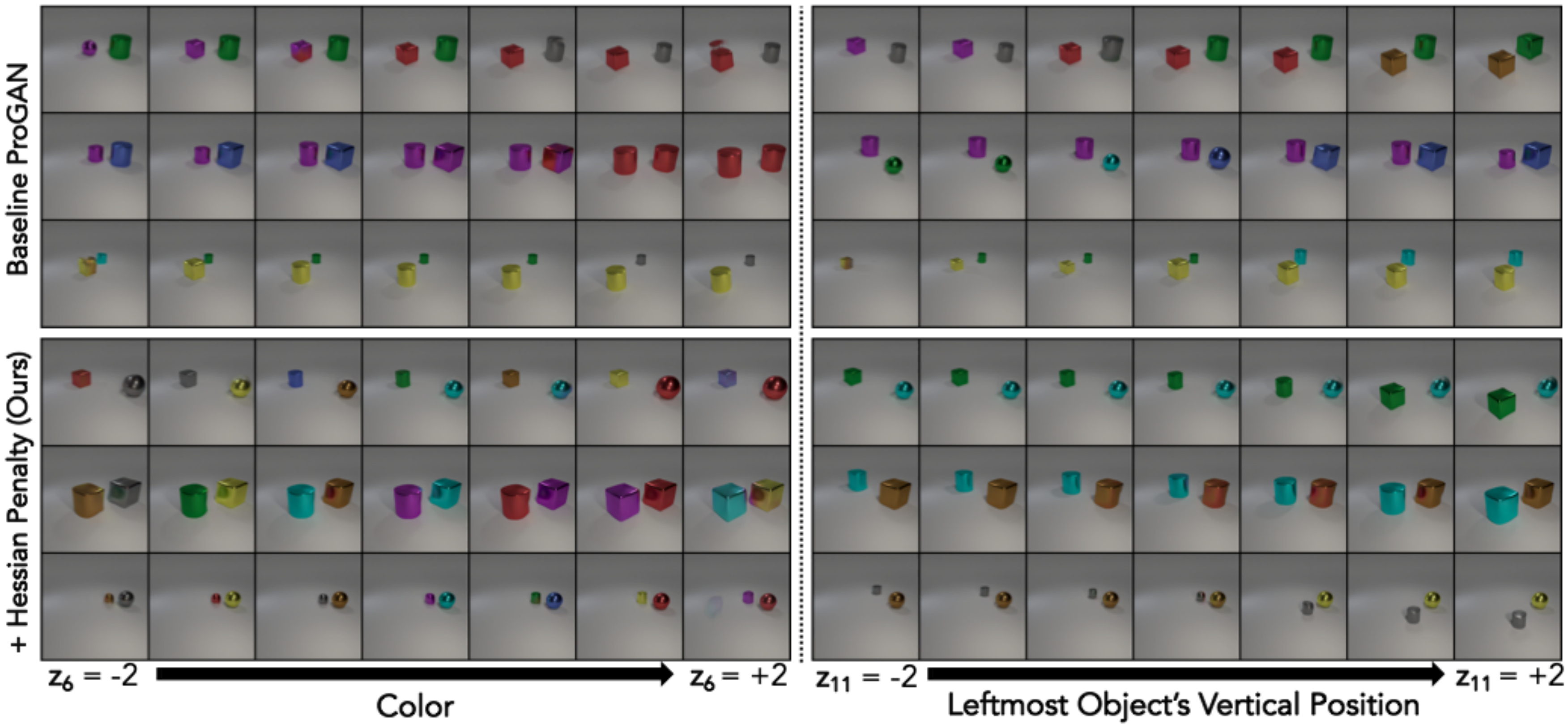
Эксперименты. ProGAN. CLEVR-Simple



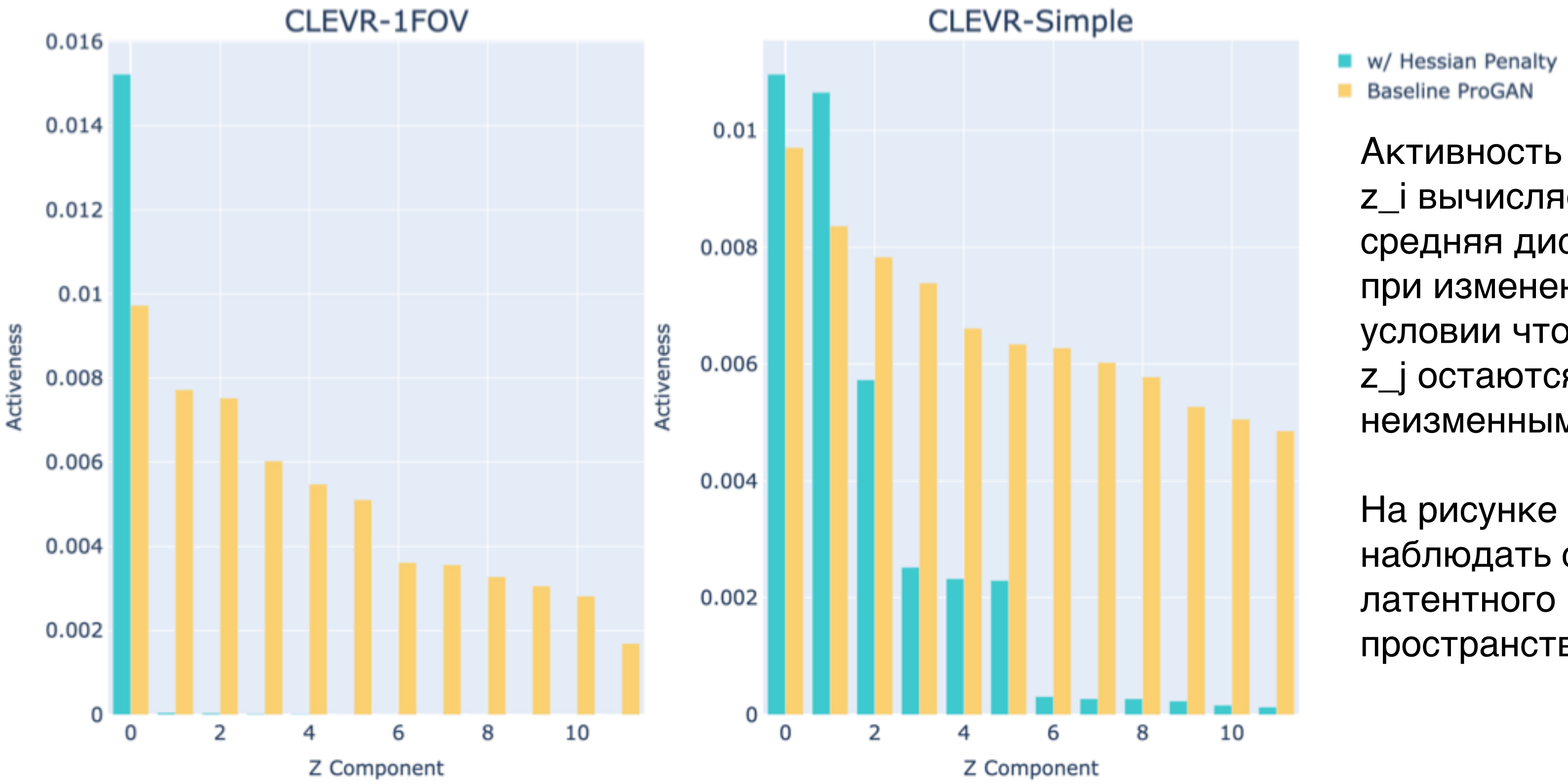
Датасет содержит один элемент на изображении, у него есть следующие свойства : цвет, форма, расположение верт. и гор.

Fine-tune с HP, получилось распутать пространство довольно наглядно.

Эксперименты. ProGAN. CLEVR-Complex



Эксперименты. Активность компонент.



Активность компоненты z_i вычисляется как средняя дисперсия G(z) при изменении z_i и при условии что остальные z_j остаются неизменными.

На рисунке можно наблюдать сжатие латентного пространства.

Количественная оценка распутывания. Метрики

$$\mathrm{PPL}(G) = \underset{z^{(1)}, z^{(2)} \sim p_z(z)}{\mathbb{E}} \left[\frac{1}{\alpha^2} d\left(G(z^{(1)}), G(\mathrm{slerp}(z^{(1)}, z^{(2)}; \alpha)) \right) \right]$$

$$\mathrm{FID} = |\mu - \mu_w|^2 + \mathrm{tr}(\Sigma + \Sigma_w - 2(\Sigma\Sigma_w)^{1/2})$$

α - параметр интерполяции. slerp() - сферическая интерполяция. d() - некоторая метрика.

μ - вектор признаковых средних исходного изображения, μ_w - вектор признаковых средних сгенерированного изображения. Σ - матрица ковариации исходного изображения, Σ_w - матрица ковариации сгенерированного изображения.

Количественная оценка распутывания.

Method	Edges+Shoes		CLEVR-Simple		CLEVR-Complex		CLEVR-U		CLEVR-1FOV	
	PPL	FID	PPL	FID	PPL	FID	PPL	FID	PPL	FID
InfoGAN	2952.2	**10.4**	56.2	**2.9**	83.9	**4.2**	766.7	3.6	22.1	6.2
ProGAN+	3154.1	10.8	64.5	3.8	84.4	5.5	697.7	3.4	30.3	9.0
ProGAN	1809.7	14.0	61.5	3.5	92.8	5.8	720.2	**3.2**	35.5	11.5
w/ HP	1301.3	21.2	45.7	25.0	**73.1**	21.1	68.7	26.6	20.8	**2.3**
w/ HP FT	**554.1**	17.3	**39.7**	6.1	74.7	7.1	**61.6**	26.8	**10.0**	4.5

Обучение направлений в BigGAN.

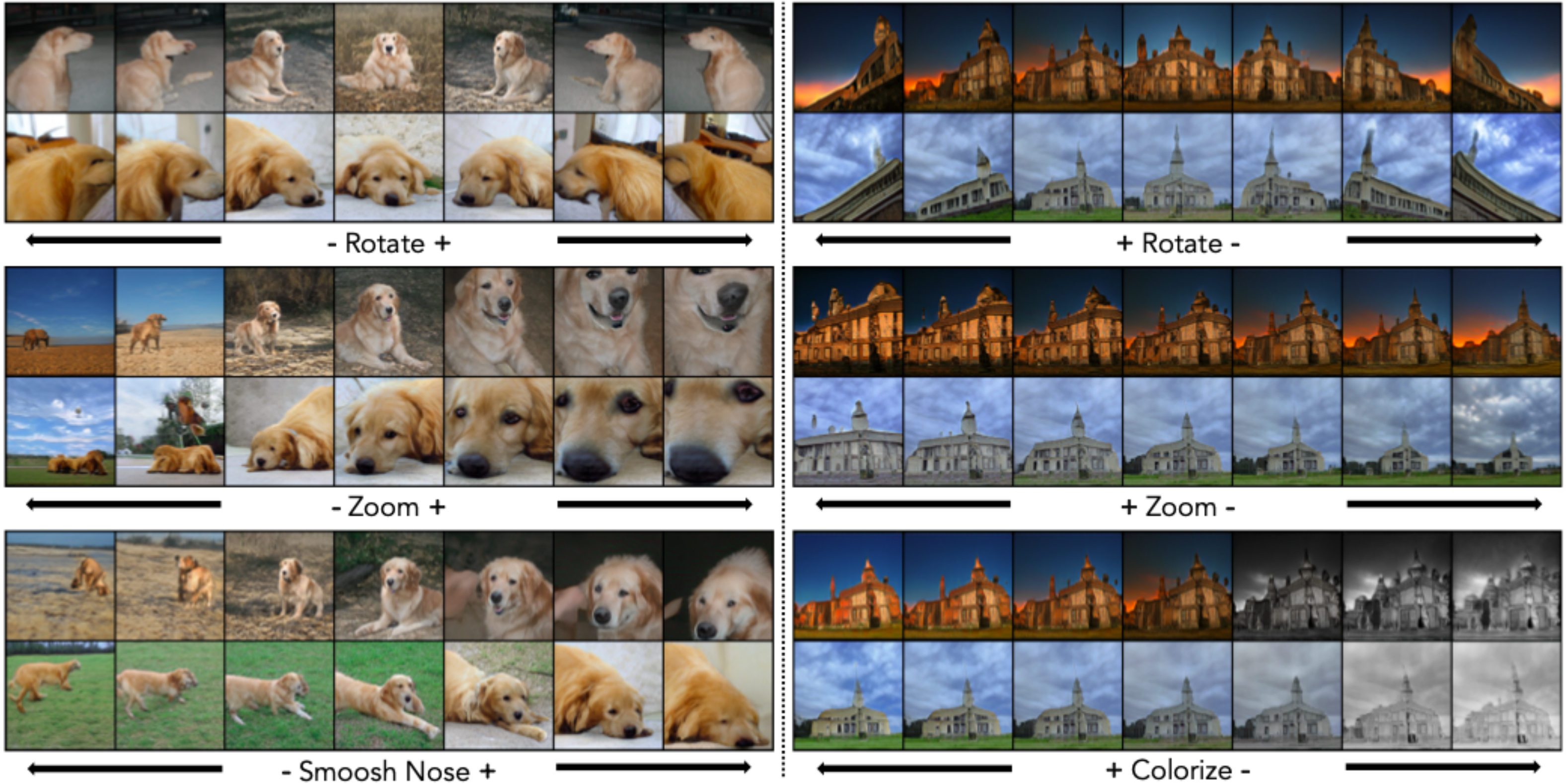
$A \in \mathbb{R}^{|z| \times N}$ $\eta \in$ U[-5;5]

$$G(z + \eta A w_i), \text{ where } w_i \in \{0,1\}^N$$

$$A^* = \arg\min_A \mathbb{E}_{z,w_i,\eta} \mathcal{L}_{\mathbf{H}}(G(z + \eta A w_i))$$

На каждом forward шаге следим за тем, чтобы A была ортогональна с помощью ортогонализации Грама-Шмидта

Результаты обучения направлений



Вопросы

1) Как авторы предложили расширить применение штрафа Гессиана на функции, которые возвращают вектор? Как вычисляется штраф на практике (Напишите эффективную формулу через аппроксимацию) ?
2) Как авторы решили находить матрицу A для поиска направлений в задаче BigGAN (напишите формулу) ? Какими свойствами обладает эта матрица?
3) В чем суть распутывания латентного пространства с помощью штрафа Гессиана? Приведите наглядный пример распутывания на задачах edges2shoes и любого из датасетов CLEVR.

Источники

1) https://arxiv.org/pdf/2008.10599.pdf
2) https://arxiv.org/pdf/1710.10196.pdf
3) https://machinelearningmastery.com/how-to-implement-the-frechet-inception-distance-fid-from-scratch/