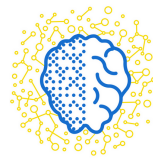
Style Transfer

(in images and text)

Ivan Skorokhodov

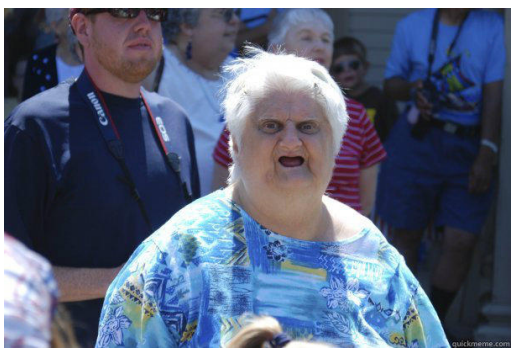
December 20, 2018



Neural Style[1]

Overview

1. Take three image:
   - Image for content x_c (e.g., your grandma picture)
   - Image for style x_s (e.g., "La Muse" by Picasso)
   - Image of random noise x_r, which we'll transform into the result
2. Pass them in VGG-19 and extract features
3. Run optimization to make features of x_r to be close to x_c and be correlated like features of x_s



(a) Content x_c

(b) Style x_s

(c) Result x_r

[1] Gatys, Ecker, and Bethge, "A Neural Algorithm of Artistic Style".

Neural Style

Computing loss

To transfer style we should find such $\boldsymbol{x}_r$, which minimizes:

$$\mathcal{L}(\boldsymbol{x}_r) = \alpha \mathcal{L}_{\text{content}}(\boldsymbol{x}_r) + \beta \mathcal{L}_{\text{style}}(\boldsymbol{x}_r),$$

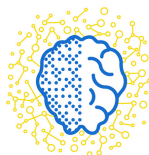
where:

$$\mathcal{L}_{\text{content}}(\boldsymbol{x}_r) = \frac{1}{2} \| F_c^k - F_r^k \|^2$$

$$\mathcal{L}_{\text{style}}(\boldsymbol{x}_r) = \frac{1}{k} \sum_k \frac{1}{4 n_k^2 m_k^2} \| F_s^k (F_s^k)^\top - F_r^k (F_r^k)^\top \|^2$$

Here F_c^k, F_s^k, F_r^k — activations in k-th layer for $\boldsymbol{x}_c, \boldsymbol{x}_s, \boldsymbol{x}_r$; n_k, m_k — dimension and amount of features in k-th layer.

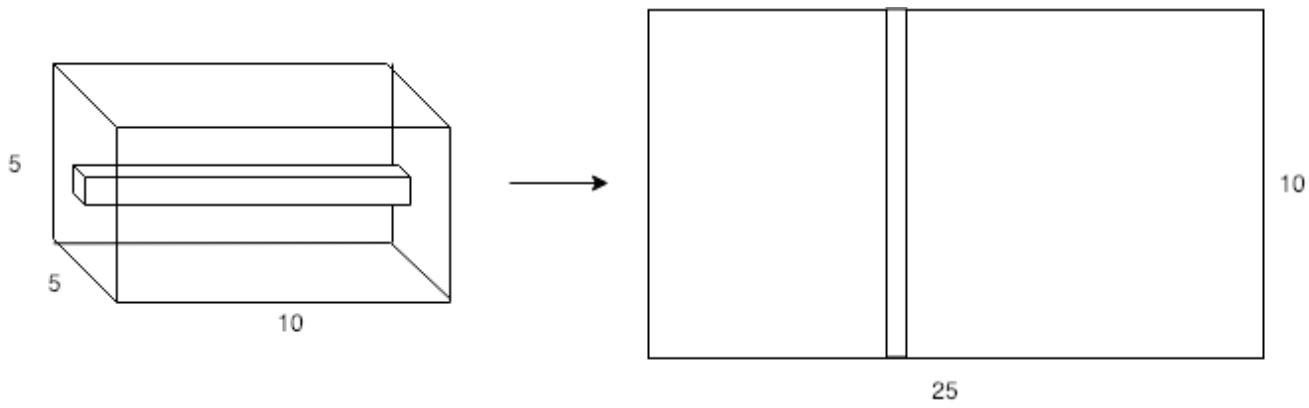
- For $\mathcal{L}_{\text{content}}(\boldsymbol{x}_r)$ we use layer #10.
- For $\mathcal{L}_{\text{style}}(\boldsymbol{x}_r)$ we use layers #1, 3, 5, 9, 13!



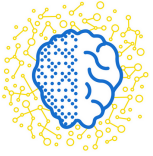
Neural style

Extracting features

To generate F^k, just pass $\boldsymbol{x}$ through VGG, obtain 3D matrix of activations (i.e. after ReLU was applied) and lay it out into 2D matrix:



- $\mathcal{L}_{\text{content}}(\boldsymbol{x}_r)$ uses feature matrix F^k as it is
- $\mathcal{L}_{\text{style}}(\boldsymbol{x}_r)$ uses $F^k \cdot (F^k)^\top$, i.e. a *Gram matrix* of matrix F^k.



Neural Style

Gram matrix?

Q. Why minimizing difference between Gram matrices transfers style?
A. It's equivalent to matching the distributions of features of $\boldsymbol{x}_r$ and $\boldsymbol{x}_s$.[2]

Proof 1 (intuitive). Let $F = [f_1, ..., f_n]$, where $\bar{f} = 0$. Then Gram matrix $G = FF^\top$ is an empirical covariance (up to $\frac{1}{n-1}$):

$$\hat{\text{Cov}}[f] = \frac{1}{n-1}\sum_{i=1}^{n} f_i f_i^\top = \frac{1}{n-1} FF^\top$$

- Matching covariances of zero-mean distributions is like matching their first two moments
- Usually, matching first two moments is enough (e.g. for normal distribution they are sufficient statistics)

[2] And why matching distributions transfers style? — hren ego znaet

Maximum Mean Discrepancy (MMD)[4]

Gram matrix?

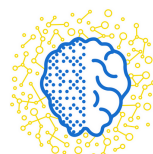
Explanation 2 (strict). Let $F = [f_1, ..., f_n]$ and $D = [d_1, ..., d_m]$ (i.e. stack features as columns). Then minimizing $\|FF^\top - DD^\top\|^2$ is equivalent[3] to minimizing squared MMD between f and d:

$$\text{MMD}^2[f, d] = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i}^{n} k(f_i, f_j) + \frac{1}{m(m-1)} \sum_{i=1}^{m} \sum_{j \neq i}^{m} k(d_i, d_j) - \frac{2}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} k(f_i, d_j)$$

with polynomial kernel of the kind:

$$k(f, d) = (f^\top d)^2$$

Other kernels transfer style too (authors tried linear, polynomial and rbf).



[3] Li et al., "Demystifying Neural Style Transfer".

[4] Gretton et al., "A Kernel Two-sample Test".

Adaptive Instance Normalization (AdaIN)[5]

Any other ways to match feature distributions?

Why another approach?

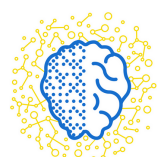
- Style transfer via optimization procedure is slow
- But we can train a CNN autoencoder for each style!
- But then we'll be limited to these styles only

AdaIN to the rescue!

- Pass $\boldsymbol{x}_c$ and $\boldsymbol{x}_s$ through VGG
- Extract activations $\boldsymbol{f}_c$ and $\boldsymbol{f}_s$ from layer #11; compute their means and stds.
- Apply AdaIN for $\boldsymbol{f}_c$ by using statistics of $\boldsymbol{f}_s$:

$$\hat{\boldsymbol{f}}_r = \text{AdaIN}(\boldsymbol{f}_c, \boldsymbol{f}_c) = \sigma(\boldsymbol{f}_s)\frac{\boldsymbol{f}_c - \mu(\boldsymbol{f}_c)}{\sigma(\boldsymbol{f}_c)} + \mu(\boldsymbol{f}_s)$$

- Decode $\hat{\boldsymbol{f}}_r$ into $\boldsymbol{x}_r$ with CNN decoder
- Extract activations of $\boldsymbol{x}_r$ from VGG layers and compute losses



[5] Huang and Belongie, "Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization".

Adaptive Instance Normalization (AdaIN)

How to compute statistics and losses?

Instance Normalization is like BatchNorm, but without batch dimension:

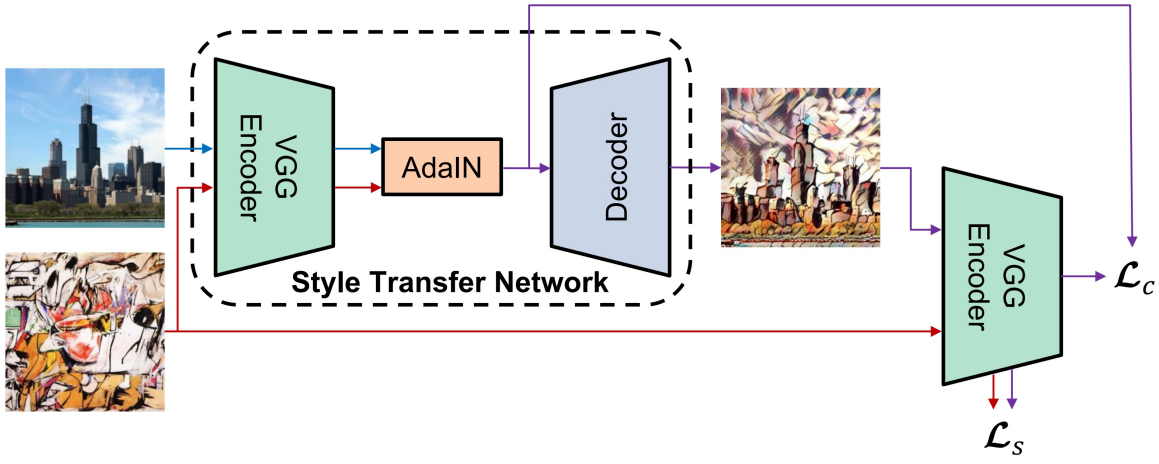
$$\mu_{nc}(x) = \frac{1}{HW}\sum_{h,w} x_{nhwc} \qquad \sigma^2_{nc}(x) = \frac{1}{HW}\sum_{h,w}\left(x_{nhwc} - \mu_{nc}(x)\right)^2$$

Loss function looks like:

$$\mathcal{L} = \mathcal{L}_{\text{content}} + \gamma \sum_k \mathcal{L}^k_{\text{style}}$$

$$\mathcal{L}^k_{\text{style}} = \|\mu(\boldsymbol{f}^k_r) - \mu(\boldsymbol{f}^k_s)\| + \|\sigma(\boldsymbol{f}^k_r) - \sigma(\boldsymbol{f}^k_s)\|$$

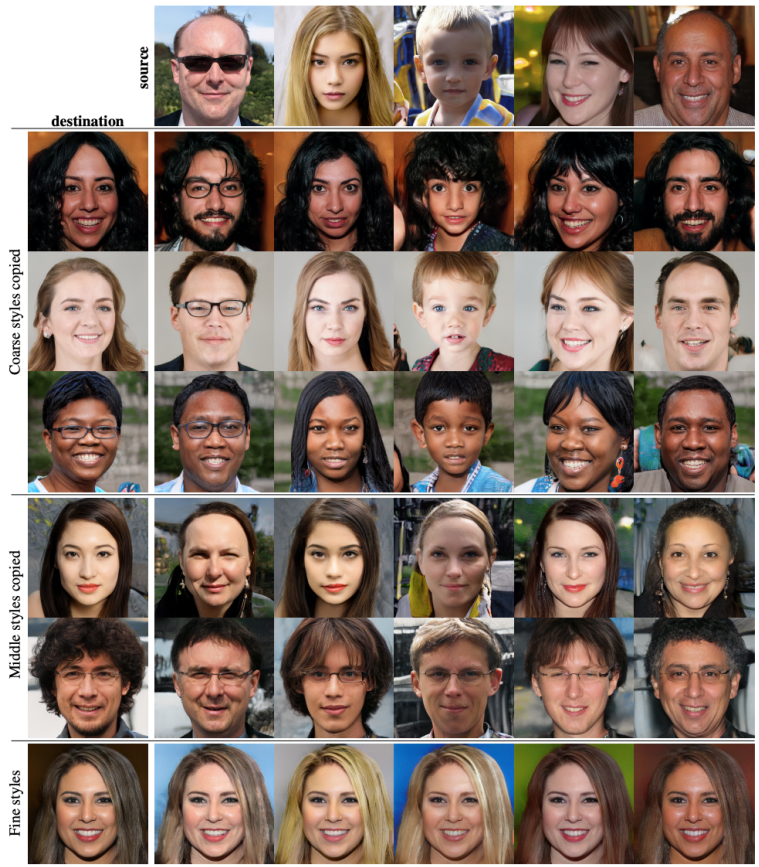
$$\mathcal{L}_{\text{content}} = \|\boldsymbol{f}_r - \hat{\boldsymbol{f}}_r\|$$



StyleGAN[6]

New GAN SotA!

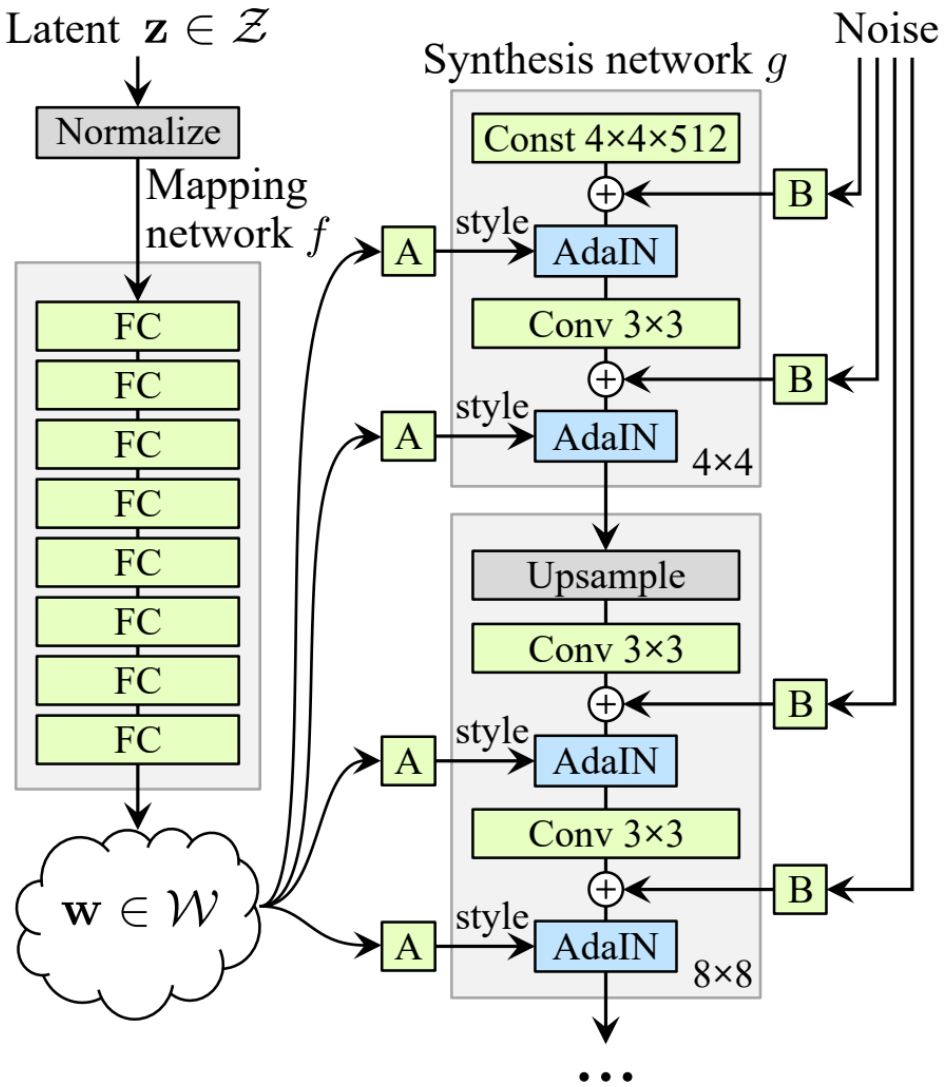
- Based on ProGAN
- Uses two sources of input: latent code and noise
- Inputs are passed at every layer
- Uses AdaIN to transfer style
- Uses old CV tricks to make produced image look nicer



[6]Karras, Laine, and Aila, *A Style-Based Generator Architecture for Generative Adversarial Networks*.

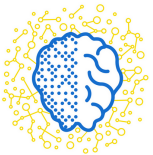
Style-based generator for GANs

How it works



1. Embed latent code z into w via deep FFN (*mapping network*)
2. Initialize Generator with constant input (which is learnt)
3. Transform w via affine mapping before passing to G
4. Noise has size $H \times W \times 1$: it is broadcasted and scaled for each feature map
5. New mean and std for AdaIN comes from $\boldsymbol{y} = A(w)$:

$$\text{AdaIN}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{y}_\sigma \frac{\boldsymbol{x} - \mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})} + \boldsymbol{y}_\mu$$



StyleGAN

Some (not strict) architectural implications and other interesting things

- Passing noise at every layer allows styles to localize in the network

(a) Generated image (b) Stochastic variation (c) Standard deviation

- Mapping Network eases disentanglement:
  - If $z \sim \mathcal{N}(0, 1)$ is disentangled, then mapping $z \mapsto \textit{features}$ is warped, because G should remove unexisting combinations of features
  - Mapping Network takes this part, freeing generator's capacity
- Passing latents and noise at every layer allows generator not to carry all the information through the whole forward pass

- They used non-saturating loss with R_1 regularization for FFHQ!

$$\mathop{\mathbb{E}}_{z} \left[\log D_\theta(G_\phi(z))\right] \to \max_{\phi} \qquad R_1(\theta) = \frac{\gamma}{2} \mathop{\mathbb{E}}_{p_{data}} \left[\|D_\theta(x)\|^2\right]$$

StyleGAN

Scores

FID (Frechet Inception Distance) is calcluated like:

- Generate a mini-dataset from generator (5k-50k samples)
- Pass real and generated data through Inception-v3
- Calculate empirical means and covariances of activations in some middle layer
- Calculate FID as:

$$\text{FID} = \|\mu_r - \mu_g\|^2 + \text{tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2}),$$

Something like ablation study:

	Method	CelebA-HQ	FFHQ
A	Baseline Progressive GAN [26]	7.79	8.04
B	+ Tuning (incl. bilinear up/down)	6.11	5.25
C	+ Add mapping and styles	5.34	4.85
D	+ Remove traditional input	5.07	4.88
E	+ Add noise inputs	**5.06**	4.42
F	+ Mixing regularization	5.17	**4.40**

CycleGAN[7]

Overview

We have a dataset of zebras and a dataset of horses (unparallel!), and want to transform zerbras to horses and horses to zebras

- Train two GANs: a GAN per each dataset
- Feed images of opposite domain to generators (instead of noise)
- Add *cycle consistency* loss

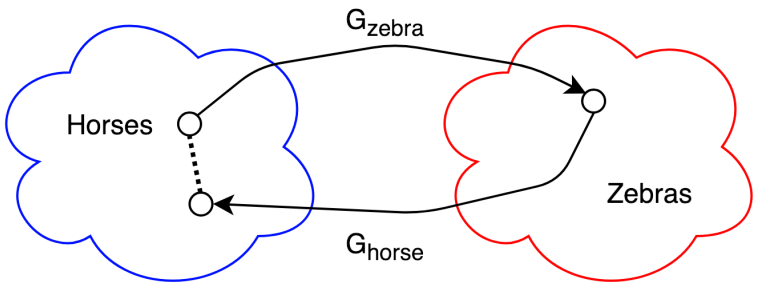


Figure: Cycle consistency

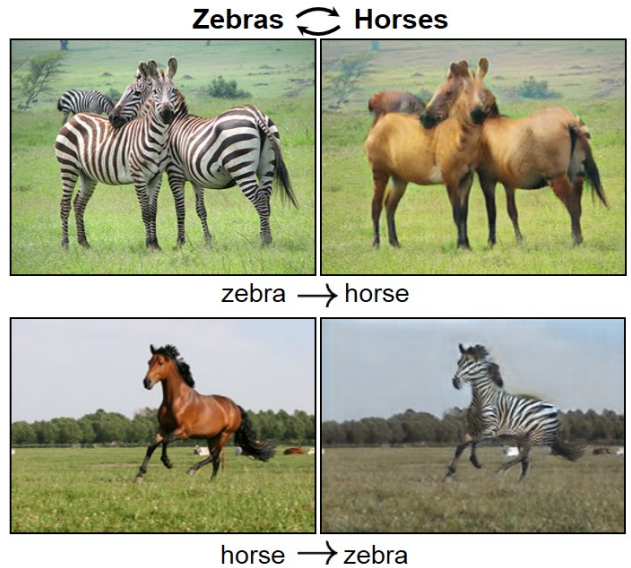


Figure: Samples from CycleGAN, trained on horses and zebras

[7] Zhu et al., "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks".

CycleGAN

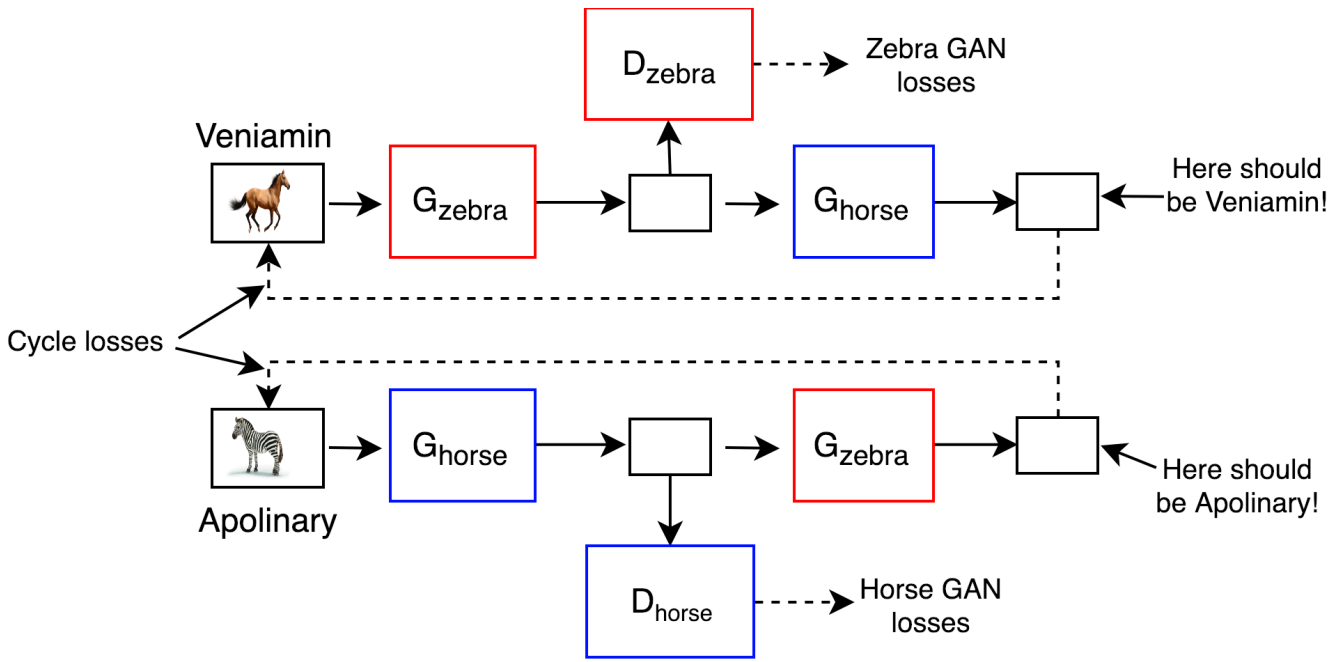
How to compute losses

$$\mathcal{L} = \mathcal{L}_{\text{Zebra GAN}} + \mathcal{L}_{\text{Horse GAN}} + \alpha \mathcal{L}_{\text{cycle}}$$

$$\mathcal{L}_{\text{Zebra GAN}} = \mathop{\mathbb{E}}_{z \sim p_z} [\log D_z(z)] + \mathop{\mathbb{E}}_{h \sim p_h} [\log(1 - D_z(G_z(h)))]$$

$$\mathcal{L}_{\text{Horse GAN}} = \mathop{\mathbb{E}}_{h \sim p_h} [\log D_h(h)] + \mathop{\mathbb{E}}_{h \sim p_h} [\log(1 - D_h(G_h(z)))]$$

$$\mathcal{L}_{\text{cycle}} = \mathop{\mathbb{E}}_{z \sim p_z} [\| G_z(G_h(z)) - z \|_1] + \mathop{\mathbb{E}}_{h \sim p_h} [\| G_h(G_z(h)) - h \|_1]$$



Unsupervised Neural Machine Translation[8][9]

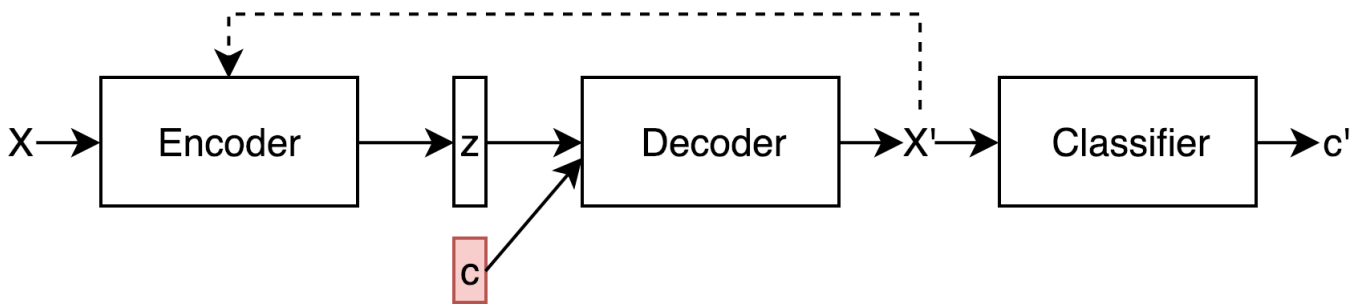
How to do unsupervised NMT in 2k18:

- Learn to translate in both directions
- Use shared encoder, initialized with cross-aligned word embeddings
- Use shared decoder, which recieves language-dependent BOS token
- Pretrain model with denoising autoencoding (words are dropped and swapped)
- Use back-translation (a lot)!
- And do not forget about fastText BPE (+3 BLEU)
- Switch to phrase-based statistical MT, because it works better (for $\sim$ 3 BLEU)

[8] Artetxe et al., "Unsupervised Neural Machine Translation".
[9] Lample et al., "Phrase-Based & Neural Unsupervised Machine Translation".

First works on Style Transfer[10]

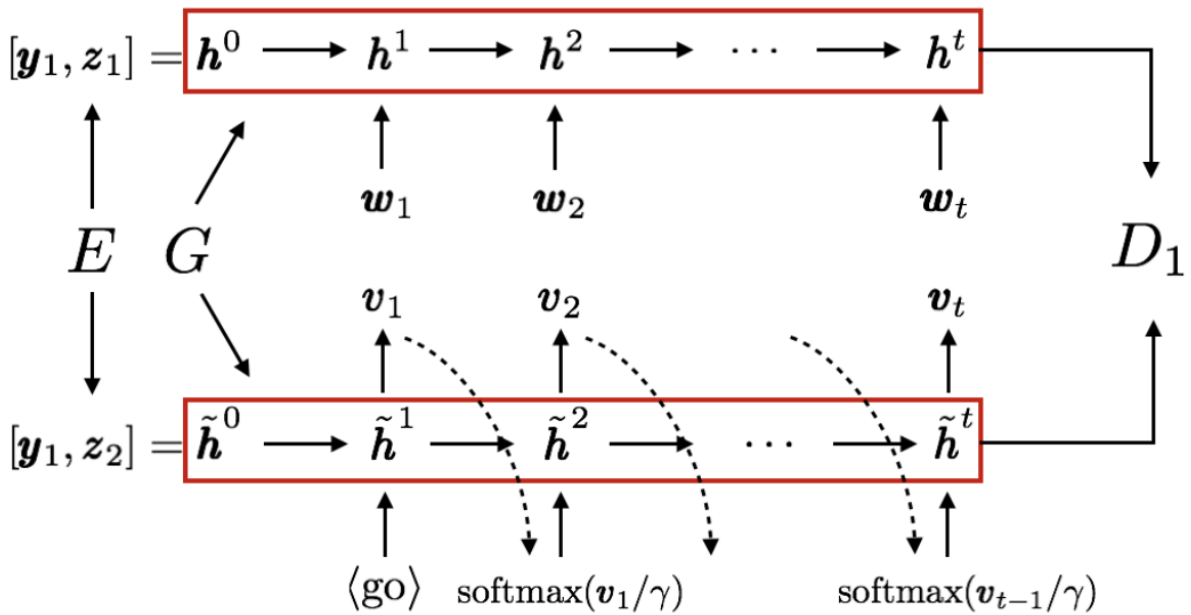
- Train VAE on text
- Concatenate sentiment flag c to z
- Penalize decoder for bad classifications of decoded sentences
- Some other training details:
  - Pretrain classifier on normal dataset
  - Minimize entropy of classifier as additional loss
  - Encode decoded sentence and penalize decoder if original z has low probability
  - Soft embeddings from decoder are passed to classifier



[10] Hu et al., "Controllable Text Generation".

Style Transfer via Professor-Forcing[11]

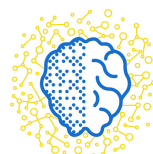
- Both encoder and decoder are shared and trained to autoencode
- Both encoder and decoder receive sentiment flag $\boldsymbol{y}_c$
- Train 2 discriminators D_1 and D_2 for each domain (positive and negative sentiment)
- Discriminators receive hidden states instead of word embeddings (i.e. Professor-Forcing)



[11] Shen et al., "Style Transfer from Non-Parallel Text by Cross-Alignment".

How to measure style transfer?[12]

- *Content preservation*:
  1. Embed words of transfered and original sentence via GloVe
  2. Transform into a single vector from avg/min/max pooling
  3. Compute cosine distance with original sentence
- *Transfer strength*: train classifier and measure its accuracy
- BLEU (or any other MT metric)
- Human Evaluations

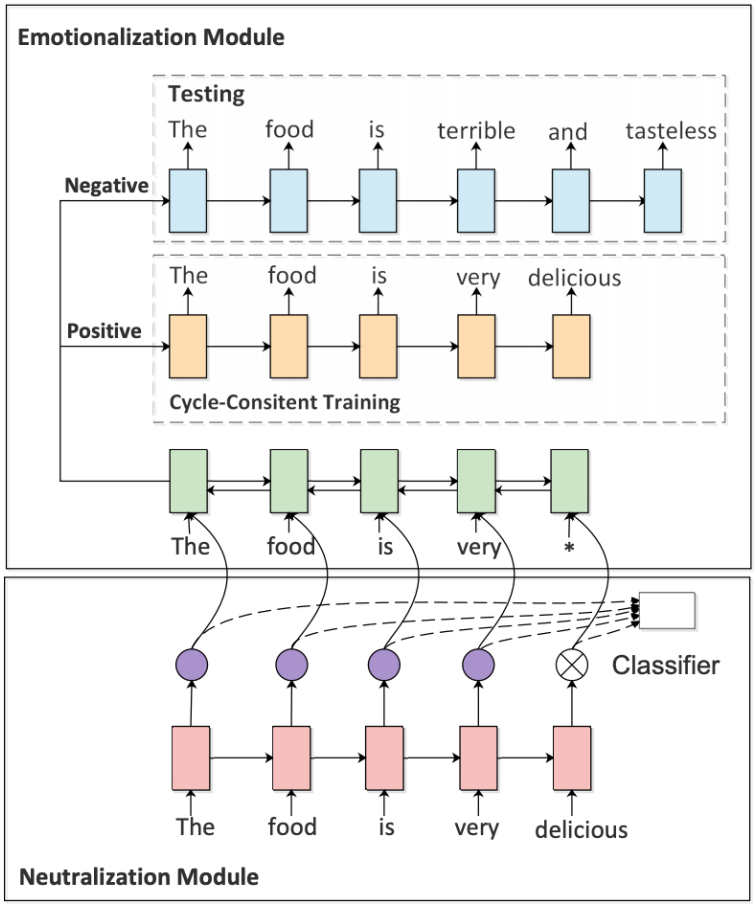


[12] Fu et al., "Style Transfer in Text: Exploration and Evaluation".

What if care about sentiment transfer only?[13]

Let's train the model just to replace positive words with negative

- Train two networks: neutralization net N and emotionalization net E
- N is LSTM net, which detects and removes emotional words
- E is encoder with two decoders, which receives neutralized sentence and inserts emotional words
- Pretrain N on classification, E on autoencoding
- After pretraining, we can train with policy gradients



[13] Xu et al., "Unpaired Sentiment-to-Sentiment Translation: A Cycled Reinforcement Learning Approach".

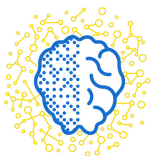
What if care about sentiment transfer only?

Some details

- How to pretrain neutralization module N:
  - Let h_i is the i-th output of N and we have T words
  - Compute attention weights α_i of h_i with h_T
  - Predict sentiment class as $y = \sigma(Wc)$ where $c = \sum_i \alpha_i h_i$
  - Discretize α_i and generate synthetic training dataset for N
- How to rl:
  - E is trained without RL, just train him to reconstruct neutralized sentence ("cycle approach"?)
  - Train module N with policy gradient:

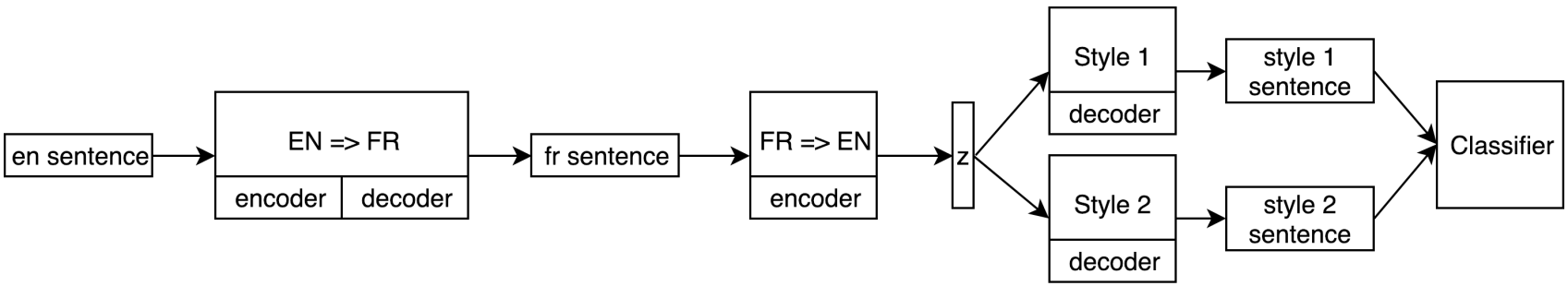
$$\nabla_\theta J = \mathbb{E}\left[R \cdot \nabla_\theta \log P_N(\hat{\alpha}|x)\right]$$

$$\text{with reward: } R = (1+\beta^2) \cdot \frac{\text{BLEU} \cdot \text{Confidence}}{\beta^2 \cdot \text{BLEU} + \text{Confidence}}$$



Style Transfer via Machine Translation[14]

- Train EN→FR and FR→EN NMT models
- Translate with EN→FR, encode with FR→EN model
- Train to style decoders for each style
- Train classifier to "fine-tune" decoders
- Datasets used: male↔female Yelp reviews and democratic↔republican facebook posts



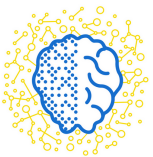
[14] Prabhumoye et al., "Style Transfer Through Back-Translation".

Style Transfer via Machine Translation

Some samples

Some samples of male↔female transfers

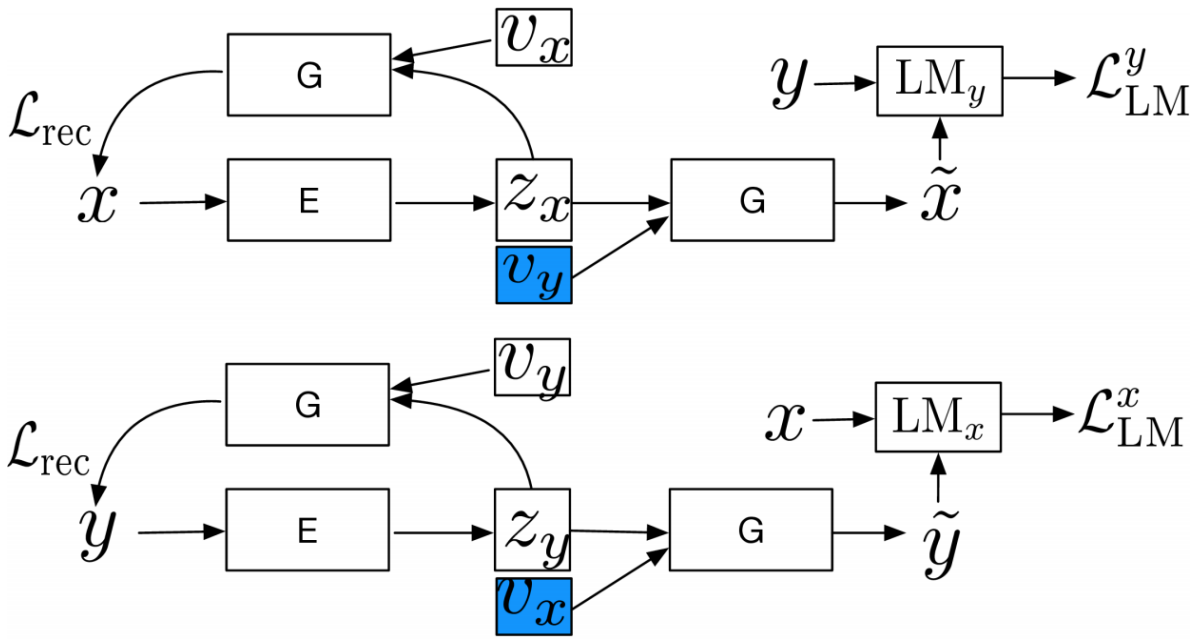
Input	Result
(M) my wife ordered country fried steak and eggs	(F) my husband ordered the chicken steak and eggs. salad and the fries
(M) the place is small but cosy and very clean	(F) the place is great but very clean and very friendly
(F) save yourself the huge headaches my husband ordered the salad	(M) you are going to be disappointed
(F) and the dressing - lrb - blue cheese - rrb - was watered down	(M) my wife ordered the mac-ncheese and the salad - lrb - $ 00 minutes - rrb - was cooked



Style Transfer with LM discriminators[15]

Overview

- Train encoder-decoder architecture as usual
- Train LM for each domain
- Let x, y be positive and negative sentences. Then decoder G_y is trained to spoof LM_x



[15] Yang et al., "Unsupervised Text Style Transfer using Language Models as Discriminators".

Style Transfer with LM discriminators

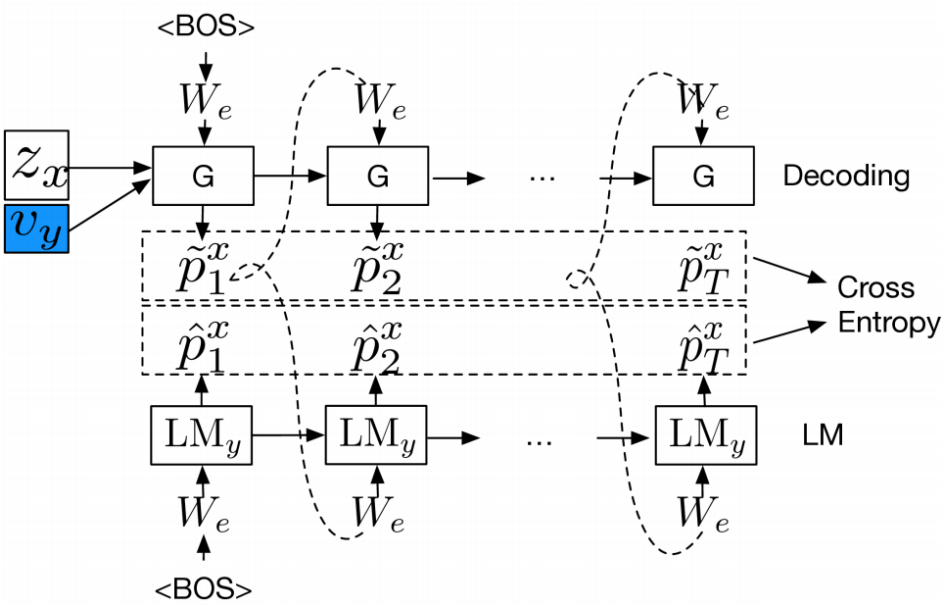
How we compute losses

- Normal autoencoding CE loss for encoder-decoder
- Loss to train LM (here $\tilde{y}$ is transfered sentence):

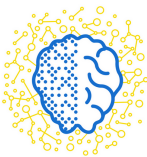
$$\mathcal{L}(\mathrm{LM}_x) = -\mathbb{E}\left[\log p_{\mathrm{LM}_x}(x)\right] + \gamma \mathbb{E}\left[\log p_{\mathrm{LM}_x}(\tilde{y})\right]$$

- Adversarial loss to train decoder

$$\mathcal{L}_{adv}(G_y) \approx \mathbb{E}\left[\sum_{t=1}^{T} \langle \tilde{p}_t, \log \hat{p}_t \rangle\right]$$ where $\tilde{p}_t, \hat{p}_t$ are G_y and LM_x probabilities

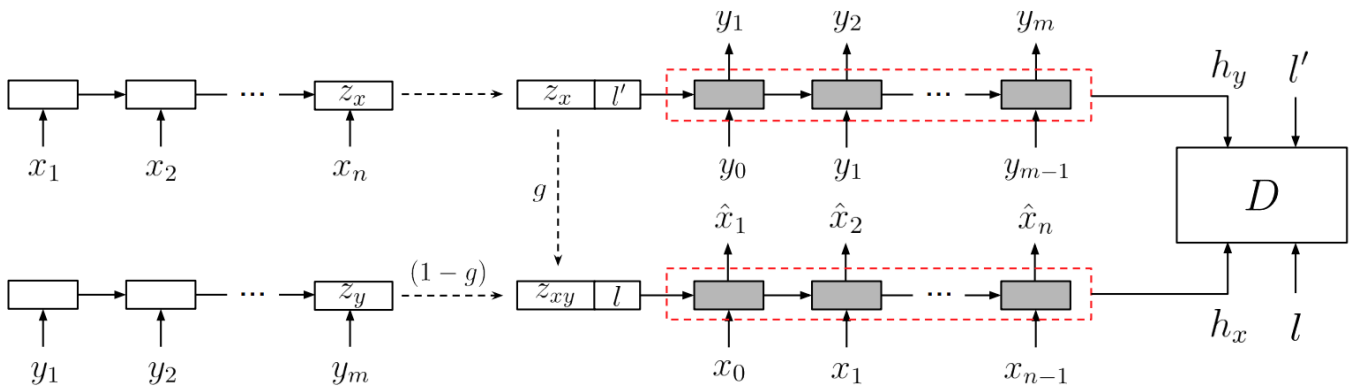


- Decoded sentence is passed to LM as soft Gumbel
- Additional tricks: normalize LM loss by length (LM prefers short sentences) and make $\tilde{x}$ to have the same length as x



Back-translation strikes again[16]

- Usual encoder-decoder architecture, but now we have *sentence attributes* instead of style flags
- Train as a general auto-encoder
- Train with general back-translation:
  1. Transfer x with attributes l into y with attributes l'
  2. Transfer y with attributes l' into x with attributes l again
- Add *interpolated reconstruction*:
  1. Transfer x with attributes l into y with attributes l'
  2. Encode x into z_x and encode y into z_y
  3. Generate z_{xy}: with some probability p_i replace i-th coordinate in z_y with value from z_x
  4. Decode z_{xy} into $\hat{x}$ with labels l



[16] Logeswaran, Lee, and Bengio, "Content preserving text generation with attribute controls".

Back-translation strikes again

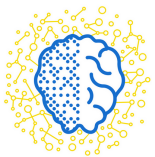
Adversarial loss details

Let's also add discriminator loss:

- Discriminator D recieves hidden states from decoder and sentence attributes
- Adversarial loss includes three kind of pairs:
  - Auto-encoded sentence x, real attributes (as true sample)
  - Auto-encoded sentence x, mistmatched attributes (as fake sample)
  - Transfered sentence y, "real" attributes (as fake sample)
- Given attributes l', adversarial loss is:

$$\mathcal{L}_{\text{adv}} = \mathbb{E}\left[2 \log D(h_x, l) + \log(1 - D(h_y, l')) + \log(1 - D(h_x, l'))\right]$$

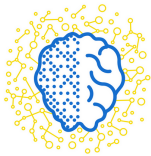
(expectation is taken over $(x, l) \sim p_{\text{data}}, y \sim p_G(\cdot|z_x, l')$)



Back-translation strikes again

Some samples from old↔new Shakespeare:

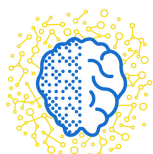
Mood	Tense	Voice	Neg.	john was born in the camp
Indicative	Past	Passive	No	john was born in the camp .
Indicative	Past	Passive	Yes	john wasn't born in the camp .
Indicative	Past	Active	No	john had lived in the camp .
Indicative	Past	Active	Yes	john didn't live in the camp .
Indicative	Present	Passive	No	john is born in the camp .
Indicative	Present	Passive	Yes	john isn't born in the camp .
Indicative	Present	Active	No	john has lived in the camp .
Indicative	Present	Active	Yes	john doesn't live in the camp .
Indicative	Future	Passive	No	john will be born in the camp .
Indicative	Future	Passive	Yes	john will not be born in the camp .
Indicative	Future	Active	No	john will live in the camp .
Indicative	Future	Active	Yes	john will not survive in the camp .
Subjunctive	Cond	Passive	No	john could be born in the camp .
Subjunctive	Cond	Passive	Yes	john couldn't live in the camp .
Subjunctive	Cond	Active	No	john could live in the camp .
Subjunctive	Cond	Active	Yes	john couldn't live in the camp .



Style Transfer without disentagled things[17]

Or a little bit more of back-translation

- Train usual Unsupervised NMT system (back-translation is crucial)
- Do not use discriminator to remove style information from encodings, because it does not work anyway: if one trains a classifier after training is done, then it will still be able to guess the style attributes in content vector with 90% accuracy
- Pass attribute embedding as BOS token to decoder
- Add attention to decoder on max-pooled parts of encoder states: run non-overlapping window of size 5 over encoder hidden states and max-pool everything inside a window
- Also learn biases in decoder output softmax for each attribute characteristic (and average them later)

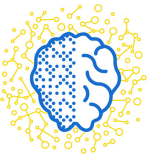


[17] Subramanian et al., "Multiple-Attribute Text Style Transfer".

Style Transfer without disentagled things

Samples from the model:

Relaxed ↔ Annoyed	
Relaxed	Sitting by the Christmas tree and watching Star Wars after cooking dinner. What a nice night 💗🎄✨
Annoyed	Sitting by the computer and watching The Voice for the second time tonight. What a horrible way to start the weekend 😡😡😡
Annoyed	Getting a speeding ticket 50 feet in front of work is not how I wanted to start this month 😒
Relaxed	Getting a haircut followed by a cold foot massage in the morning is how I wanted to start this month 😊
Male ↔ Female	
Male	Gotta say that beard makes you look like a Viking...
Female	Gotta say that hair makes you look like a Mermaid...
Female	Awww he's so gorgeous 😍 can't wait for a cuddle. Well done 😘 xxx
Male	Bro he's so f***ing dope can't wait for a cuddle. Well done bro
Age 18-24 ↔ 65+	
18-24	You cheated on me but now I know nothing about loyalty 😂 ok
65+	You cheated on America but now I know nothing about patriotism. So ok.
65+	Ah! Sweet photo of the sisters. So happy to see them together today .
18-24	Ah 😂 Thankyou 💞 #sisters ❤️ happy to see them together today

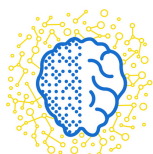


Adversarial AutoEncoders (AAE)[18]

VAE loss can be decomposed as:

$$\mathcal{L} = \mathop{\mathbb{E}}_{x}\left[\mathop{\mathbb{E}}_{q(z|x)}\left[\log p(x|z)\right]\right] + \mathop{\mathbb{E}}_{x}\left[H(q(z|x))\right] - CE(q(z)|p(z))$$

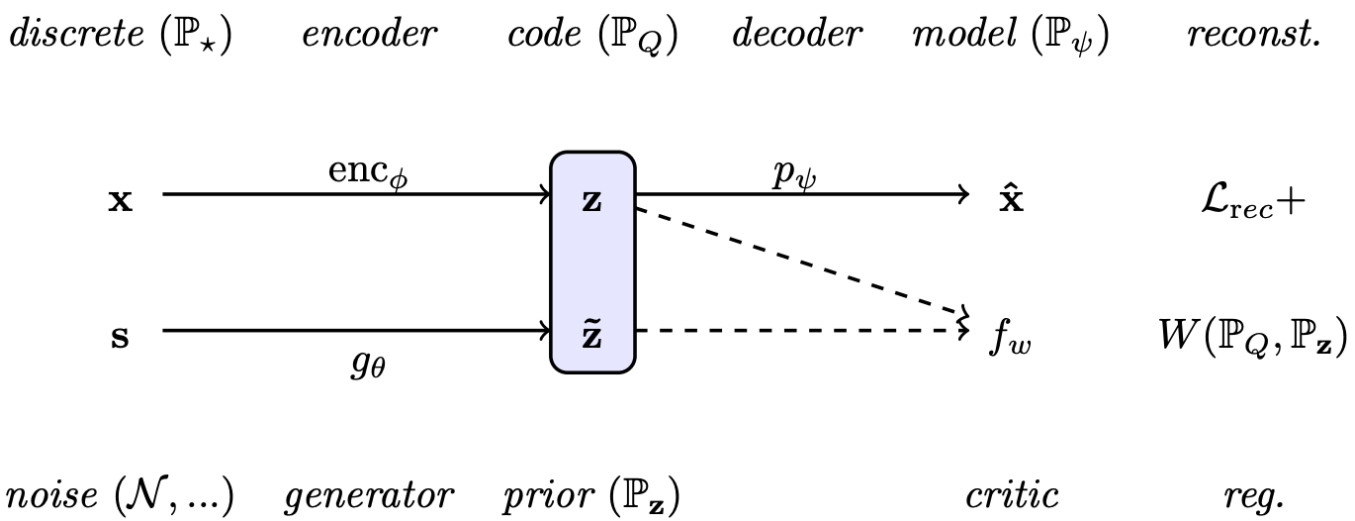
- First term is a reconstruction loss, other two terms are regularization
- AAE replaces this regularization by training a GAN to distinguish samples from $q(z)$ and $p(z)$
- Encoder is deterministic (because quality is the same)
- To sample from $q(z)$ we sample $x \sim p_d(x)$ and then compute $z = E(x)$



[18] Makhzani et al., "Adversarial Autoencoders".

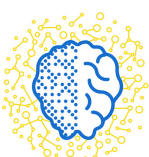
Adversarially Regularized Autoencoders (ARAE)[20]

- ARAE is like AAE but now $p(z)$ is a generator (flexible prior!)
- How to transfer style with this thing? By disentangling:
  - Train classifier to guess attributes in latent code
  - Train encoder to spoof it (so, it's a GAN without real samples)
- Flaws: sensitive to hyperparameters and have bad samples from prior[19]



[19] Cífka et al., "Eval all, trust a few, do wrong to none: Comparing sentence generation models".

[20] Zhao et al., "Adversarially Regularized Autoencoders for Generating Discrete Structures".



Adversarially Regularized Autoencoders (ARAE)

Samples from posterior:

Music	do you know a website that you can find people who want to join bands ?
⇒ Science	do you know a website that can help me with science ?
⇒ Politics	do you think that you can find a person who is in prison ?
Music	all three are fabulous artists , with just incredible talent ! !
⇒ Science	all three are genetically bonded with water , but just as many substances , are capable of producing a special case .
⇒ Politics	all three are competing with the government , just as far as i can .
Music	but there are so many more i can 't think of !
⇒ Science	but there are so many more of the number of questions .
⇒ Politics	but there are so many more of the can i think of today .

Samples from prior (without style transfer):

Marvin Armstrong has been impressed.
The running in Angelina walked control up, the life.
irburg spoke to over a 6 An state is showing Kanover mayor.
A Wayne James man, accusing The Pittsburgh sent no fund at his time to reang 2014.
Craig said being is planning the United States, admitted she are, placed CBo Sangan.
Hosting Goldman Micherro set team at school home.
The A woman 24, is gun with.