Notes on Geometric Learning

Matthew Mo

2020 年 9 月 18 日

目录

1 NeVAE

Idea VAE on graph, node-wise repr, permutation invariant.

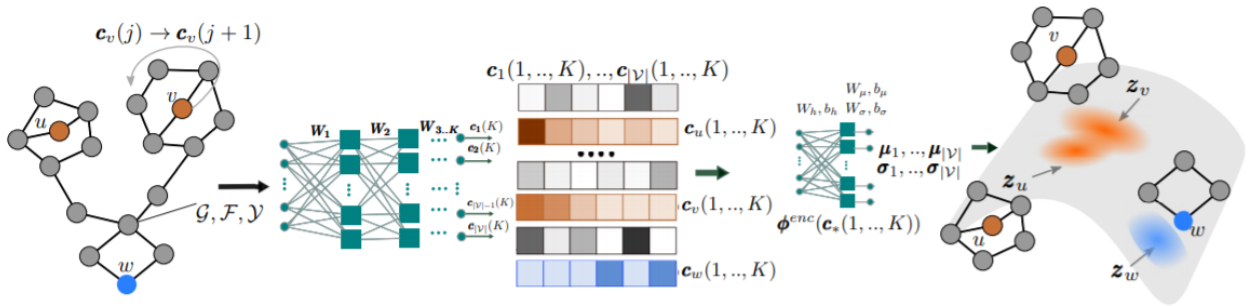


Figure 1: The encoder of our variational autoencoder for molecular graphs. From left to right, given a molecular graph $\mathcal{G}$ with a set of node features $\mathcal{F}$ and edge weights $\mathcal{Y}$, the encoder aggregates information from a different number of hops $j \leq K$ away for each node $v \in \mathcal{G}$ into an embedding vector $\mathbf{c}_v(j)$. These embeddings are fed into a differentiable function ϕ^{enc} which parameterizes the posterior distribution q_ϕ, from where the latent representation of each node in the input graph are sampled from.

Encoder: GNN-like message-passing on k-hops:

$$q_\phi(\boldsymbol{z}_u|\mathcal{V},\mathcal{E},\mathcal{F},\mathcal{Y}) \sim \mathcal{N}(\boldsymbol{\mu}_u, Diag(\boldsymbol{\sigma}_u)) \tag{1}$$

$$[\boldsymbol{\mu}_u, \boldsymbol{\sigma}_u] = \phi^{enc}(\boldsymbol{c}_u(k)_{k=1..K}) \tag{2}$$

$$\boldsymbol{c}_u(k) = \begin{cases} \boldsymbol{r}(\boldsymbol{W}_k^{\mathcal{T}}\boldsymbol{t}_u + \boldsymbol{W}_k^{\mathcal{X}}\boldsymbol{x}_u), k=1 \\ \boldsymbol{r}(\boldsymbol{W}_k^{\mathcal{T}}\boldsymbol{t}_u + \boldsymbol{W}_k^{\mathcal{X}}\boldsymbol{x}_u \odot \boldsymbol{\Lambda}(\{y_{uv}\boldsymbol{c}_v(k-1)\}_v \in N(u))), k>1 \end{cases} \tag{3}$$

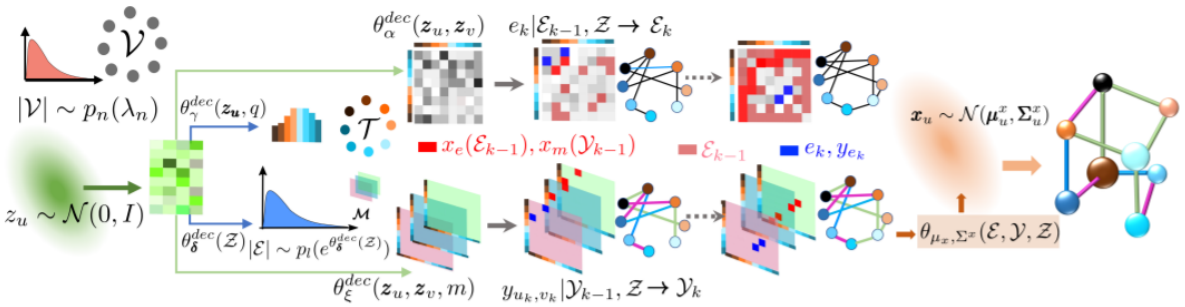


Figure 2: The decoder of our variational autoencoder for molecular graphs. From left to right, the decoder first samples the number of nodes $n = |\mathcal{V}|$ from a Poisson distribution $p_n(\lambda_n)$ and it samples a latent vector $\boldsymbol{z}_u$ per node $u \in \mathcal{V}$ from $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. Then, for each node u, it represents all potential node feature values as an unnormalized log probability vector (or 'logits'), where each entry is given by a nonlinearity θ_γ^{dec} of the corresponding latent representation $\boldsymbol{z}_u$, feeds this logit into a softmax distribution and samples the node features. Next, it feeds all latent vectors $\mathcal{Z}$ into a nonlinear log intensity function $\theta_{\boldsymbol{\delta}}^{dec}(\mathcal{Z})$ which is used to sample the number of edges. Thereafter, on the top row, it constructs a logit for all potential edges (u, v), where each entry is given by a nonlinearity θ_α^{dec} of the corresponding latent representations $(\boldsymbol{z}_u, \boldsymbol{z}_v)$. Then, it samples the edges one by one from a soft max distribution depending on the logit and a mask $\beta_e(\mathcal{E}_{k-1})$, which gets updated every time it samples a new edge e_k. On the bottom row, it constructs a logit per edge (u, v) for all potential edge weight values m, where each entry is given by a nonlinearity θ_ξ^{dec} of the latent representations of the edge and edge weight value $(\boldsymbol{z}_u, \boldsymbol{z}_v, m)$. Then, every time it samples an edge, it samples the edge weight value from a soft max distribution depending on the corresponding logit and mask $x_m(u, v)$, which gets updated every time it samples a new $y_{u_k v_k}$. Finally, for each atom u, it samples its coordinates $\boldsymbol{x}_u$ from a multidimensional Gaussian distribution whose mean $\mu_{\boldsymbol{x}}$ and variance $\Sigma_{\boldsymbol{x}}$ depends on the latent vectors of the corresponding atom and its neighbors and the underlying chemical bonds.

Decoder: genetate logits, softmax edges one-by-one, possible binary mask(for expert exp.).

$$\text{Nodes Count:} |\mathcal{V}| \sim p_l(\lambda_n) \tag{4}$$

$$\text{Latent Repr.:} \boldsymbol{z}_u \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}) \tag{5}$$

$$\text{Node Feat.:} \boldsymbol{f}_u = softmax_u(\theta_{\boldsymbol{\gamma}}^{dec}(\boldsymbol{z}_u, q)), q\text{is atom type} \tag{6}$$

$$\text{Edges Count:} |\mathcal{E}| \sim p_l(e^{\theta_{\boldsymbol{\delta}}^{dec}(\mathcal{Z})}) \tag{7}$$

$$\text{Edges Gen.:} p(e=(u,v)|\mathcal{E}_{k-1}, \mathcal{V}) = \frac{\beta_e e^{\theta_{\boldsymbol{\alpha}}^{dec}(\boldsymbol{z}_u, \boldsymbol{z}_v)}}{\sum_{e'=(u',v') \notin \mathcal{E}_{k-1}} \beta_{e'} e^{\theta_{\boldsymbol{\alpha}}^{dec}(\boldsymbol{z}'_u, \boldsymbol{z}'_v)}} \tag{8}$$

$$\text{E. Feat. Gen.:} p(y_{uv}=m|\mathcal{Y}_{k-1}, \mathcal{V}) = \frac{\beta_m(u,v) e^{\theta_{\boldsymbol{\xi}}^{dec}(\boldsymbol{z}_u, \boldsymbol{z}_v, m)}}{\sum_{m' \neq m \beta'_m(u,v) e^{\theta_{\boldsymbol{\xi}}^{dec}(\boldsymbol{z}_u, \boldsymbol{z}_v, m')}}}, \text{note: not normal softmax?} \tag{9}$$

$$\text{Pos. Gen.:} p(\boldsymbol{x}_u|\mathcal{E}, \mathcal{Y}, \mathcal{Z}) = \mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x) \tag{10}$$

$$[\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x] = [] \tag{11}$$

2 Seminar on Self/Un-Supervised Learning @ 2020/9/16

2.1 Self-Learning @ Video Learning

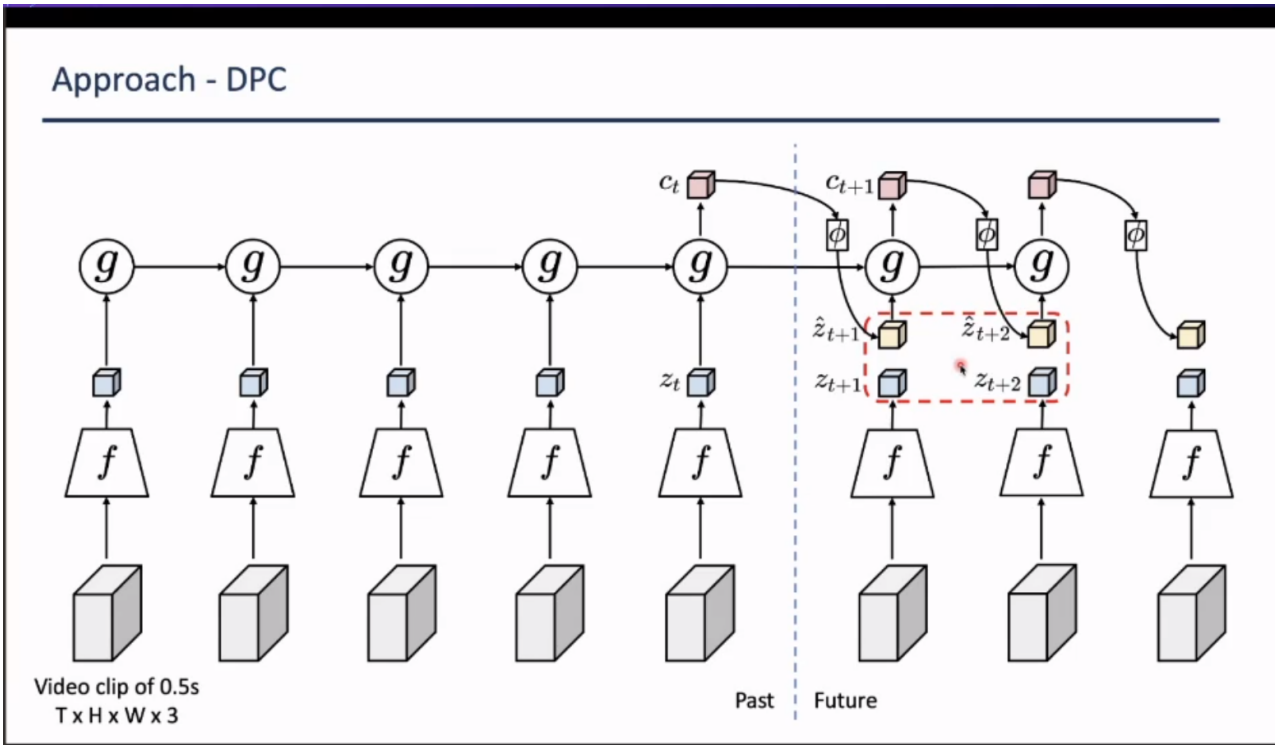
Supervised success: good & sufficient data, a way different from human! ⇒*Linda Smith, The Dev. of Embodied Cognition*

Paragidims:

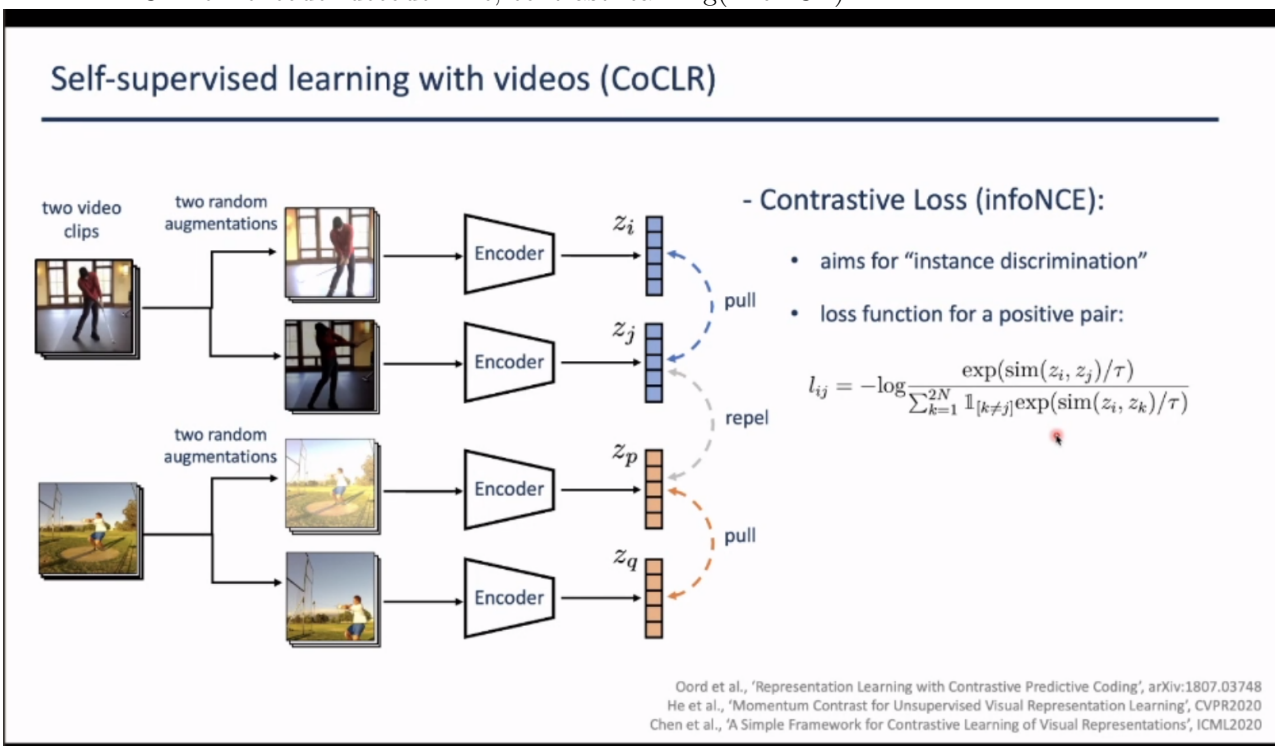
- Use proxy task(e.g. semantics repr.) for a repr., use linear probing for downstream task.
- Use proxy task(e.g. semantics repr.) for a repr., generalizable with *zero annotation*

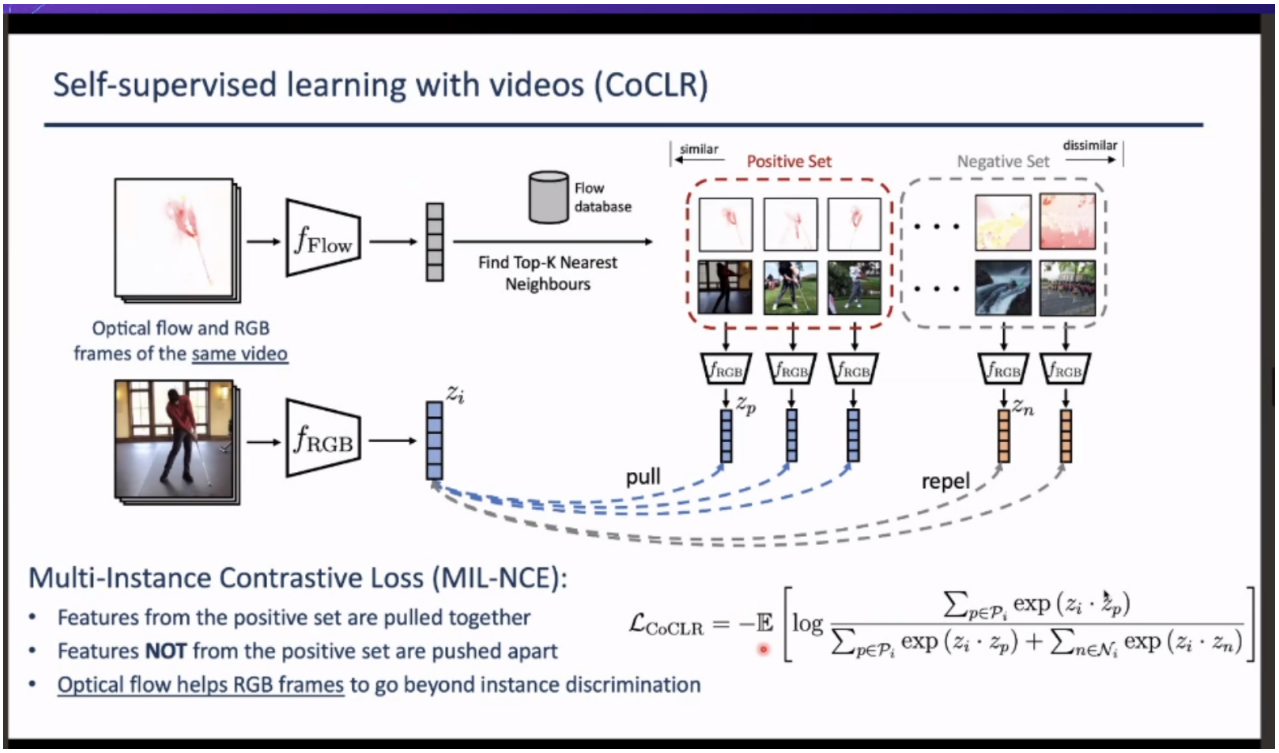
Why video-based self-supervised: like what human percepts, rich info; might with audio.
Proxy loss design: temporal info, spatial cohenrence, motions of obj., multimodal
Temporal:

- shuffle & learn
- forward or backward?(arrow of time)
- SpeedNet: which speed(frame-rate) is normal/speed-up
- ===Weak, irrelative with downstream tasks===
- DPC: *learn repr. in predicting future in video*



DPC Arch.:encoder-decoder like, contrast learning(infoNCE)





CoCLR: SimCLR like(infoNCE), colearning multimodally with motion flow(MIL-NCE, with noise added)

Audio-Video Co-learning: train a net to check if image/audio clip are same-sourced! Get both video/audio repr.

MAST: self-supervised tracking, give 1st frame mask(seg.), predict sequantial segmentations

Next:

- More efficient learning
- Scale up model to uncurrated data, like GPT-1/2/3
- Design proxy task for obj-centric learing
- Design and understand **effective memory!!!**(for video task especially)
- Hand-crafted proxy task ⇒Auto proxy task design?
- Theoratic: small or negative improvement, in upstream task to downstream task.
- Are there difficult task for supervised learning but easy for SSL(e.g. unable to label)?

2.2 Transformation Equivariance vs. Invariance @ Visial Repr. Learning

Contents:

- TER(Transformation Equivariance Repr.)

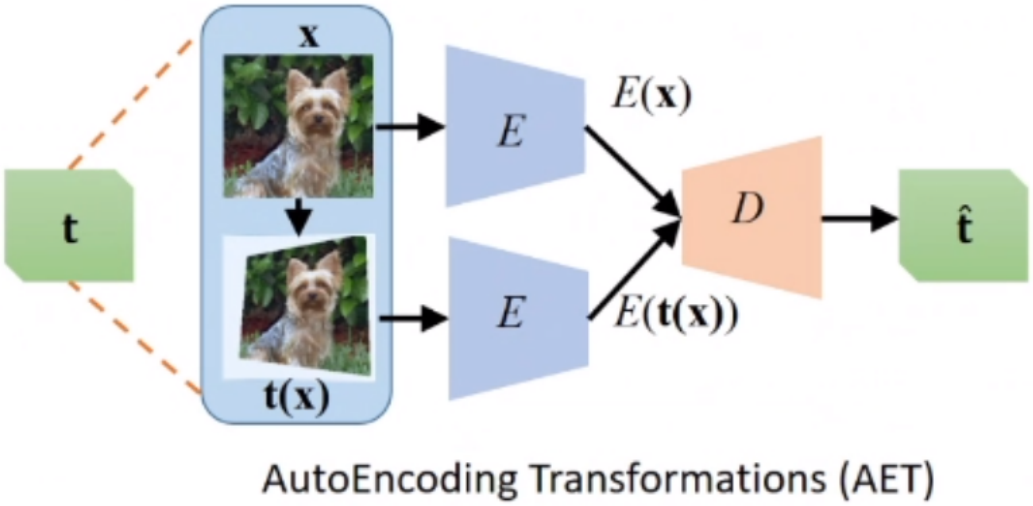
- AET(AutoEncoding Transformation)
- AVT(Autoencoding Variational Transformation)
- SAT(Semi-supervised Autoencoding Transformation)

CNN = Translation Equivariant Repr. + FC Classifier. Go beyond: Tranformation Equivariant Repr. + Tranformation Invariant Classifier

Trans. Equiv.: $E(\boldsymbol{t}(\boldsymbol{x})) = \boldsymbol{\rho}(\boldsymbol{t})[E(\boldsymbol{x})]$. Trans. Inv. is when $\boldsymbol{\rho} \equiv \mathbf{1}_E$.

Steerability: $\boldsymbol{\rho}$ is inpend. with sample $\boldsymbol{x}$.

Targets: Non-linear $\boldsymbol{\rho}$, General Transformation(e.g. recoloring)



AutoEncoding Transformations (AET)

AET loss for training

- Parameterized Transformations: $\mathcal{T} = \{t_\theta \mid \theta \sim \Theta\}$

  E.g. affine or projective: $l\left(\mathbf{t}_\theta, \mathbf{t}_{\hat{\theta}}\right) = \frac{1}{2}\,||M(\theta) - M(\hat{\theta})||_2^2$

- GAN-Induced Transformations: transformed image $\mathbf{G(x, z)}$

$$l(\mathbf{t_z}, \mathbf{t_{\hat{z}}}) = \frac{1}{2}\,||\mathbf{z} - \hat{\mathbf{z}}||_2^2$$

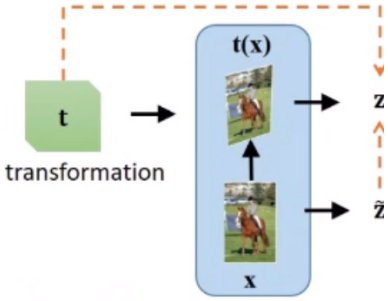
- Non-Parametric Transformations

$$l(\mathbf{t}, \hat{\mathbf{t}}) = \frac{1}{2}\mathbb{E}_{\mathbf{x}\sim X}\mathrm{dist}(\mathbf{t}(\mathbf{x}), \hat{\mathbf{t}}(\mathbf{x}))$$

AET:

- use autoencoders to learn **transformations**
- trans. generated randomly for self-supervised learing
- use Siamese net as encoder backbone
- AET loss: parameterized, non-parametric, GAN-induced

An Information-Theoretical Insight



- Train a TER model $\boldsymbol{\theta}$ by maximizing

$$\max_{\boldsymbol{\theta}} \; \boldsymbol{I_\theta}(\mathbf{z}; \tilde{\mathbf{z}}, \mathbf{t})$$

- By chain rule of mutual information, we have

$$\boldsymbol{I_\theta}(\mathbf{z}; \tilde{\mathbf{z}}, \mathbf{t}) = \boldsymbol{I_\theta}(\mathbf{z}; \tilde{\mathbf{z}}, \mathbf{t}, \mathbf{x}) - \boldsymbol{I_\theta}(\mathbf{z}; \mathbf{x}|\tilde{\mathbf{z}}, \mathbf{t}) \leq \boldsymbol{I_\theta}(\mathbf{z}; \tilde{\mathbf{z}}, \mathbf{t}, \mathbf{x})$$

  - $\boldsymbol{I_\theta}(\mathbf{z}; \tilde{\mathbf{z}}, \mathbf{t})$ attains its maximum value $\boldsymbol{I_\theta}(\mathbf{z}; \tilde{\mathbf{z}}, \mathbf{t}, \mathbf{x})$ (the upper bound) when

$$\boldsymbol{I_\theta}(\mathbf{z}; \mathbf{x}|\tilde{\mathbf{z}}, \mathbf{t}) = \mathbf{0}$$

Steerability: Given $(\tilde{\mathbf{z}}, \mathbf{t})$, $\mathbf{x}$ contains no more information about $\mathbf{z}$.

- **Nonlinearity** of transformation $\boldsymbol{\rho}(\mathbf{t})$ in representations.

AVT: Autoencoding Variational Transformations

- Unable to maximize the mutual information directly
  - Intractable to evaluate the posterior $p_\theta(\mathbf{t}|\mathbf{z}, \mathbf{x})$
- Deriving a lower bound by introducing a transformation decoder $q_{\boldsymbol{\phi}}$

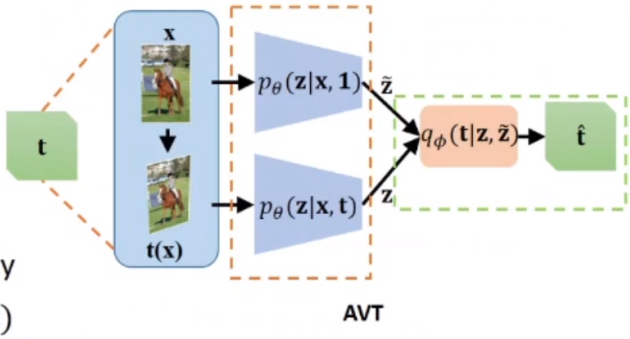
$$\boldsymbol{I_\theta}(\mathbf{z}; \tilde{\mathbf{z}}, \mathbf{t}) \geq H(\mathbf{t}|\tilde{\mathbf{z}}) + \mathbb{E}_{\boldsymbol{p_\theta}(\boldsymbol{t},\boldsymbol{z},\tilde{\boldsymbol{z}})} \log q_{\boldsymbol{\phi}}(\boldsymbol{t}|\boldsymbol{z}, \tilde{\boldsymbol{z}})$$

- Unsupervised loss to learn AVT

$$\max_{\boldsymbol{\theta},\boldsymbol{\phi}} \mathbb{E}_{\boldsymbol{p_\theta}(\mathbf{t},\mathbf{z},\tilde{\mathbf{z}})} \log q_{\boldsymbol{\phi}}(\boldsymbol{t}|\boldsymbol{z}, \tilde{\boldsymbol{z}})$$

AVT: Autoencoding Variational Transformations

- Generative process
  - Given an image $\mathbf{x}$ sampled from $p(\mathbf{x})$
  - Sample a transformation $\mathbf{t}$ from $p(\mathbf{t})$
  - Apply $\mathbf{t}$ to $\mathbf{x}$, resulting in $\mathbf{t}(\mathbf{x})$
  - Sample a representation $\mathbf{z}$ of $\mathbf{t}(\mathbf{x})$ from $p_\theta(\mathbf{z}|\mathbf{x}, \mathbf{t})$
    - $\tilde{\mathbf{z}}$ is sampled by setting $\mathbf{t}$ to an identity
  - Decode transformations $\hat{\mathbf{t}}$ from $q_\phi(\mathbf{t}|\mathbf{z}, \tilde{\mathbf{z}})$

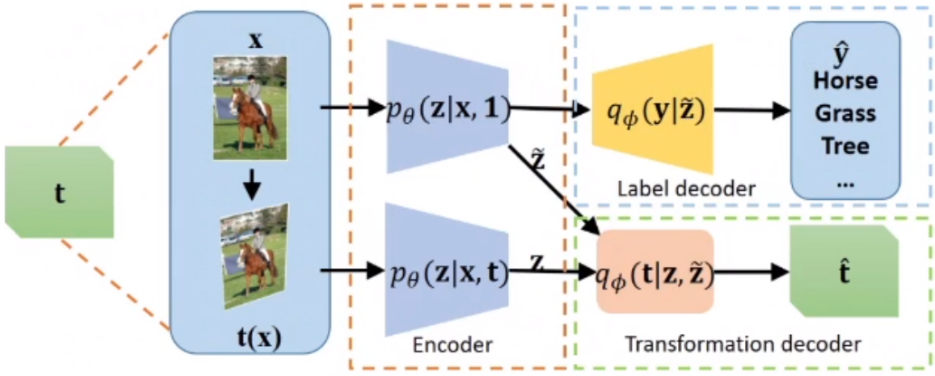


AVT:

- minimize z, x mutual info $\Rightarrow$maxmize $I_\theta(z; \tilde{z}, t)$

SAT: Semi-Supervised Autoencoding Transformation

- Adding a label decoder $q_\phi(\mathbf{y}|\tilde{\mathbf{z}})$to approximate the posterior $p_\theta(\mathbf{y}|\mathbf{x})$



Variational Bound

- By introducing label decoder and transformation decoder, we have

$$I_\theta(\mathbf{y},\mathbf{z};\tilde{\mathbf{z}},\mathbf{t}) \geq \mathbb{E}_{p_\theta(\mathbf{y},\mathbf{z},\tilde{\mathbf{z}})} \log q_\phi(\mathbf{y}|\mathbf{z},\tilde{\mathbf{z}}) + \mathbb{E}_{p_\theta(\mathbf{t},\mathbf{z},\tilde{\mathbf{z}})} \log q_\phi(\mathbf{t}|\mathbf{z},\tilde{\mathbf{z}})$$

Label decoder — Transformation decoder

- Jointly maximizing over encoder θ and decoders ϕ

$$\max_{\theta,\phi} \mathbb{E}_{p_\theta(\mathbf{y},\mathbf{z},\tilde{\mathbf{z}})} \log q_\phi(\mathbf{y}|\mathbf{z},\tilde{\mathbf{z}}) + \mathbb{E}_{p_\theta(\mathbf{t},\mathbf{z},\tilde{\mathbf{z}})} \log q_\phi(\mathbf{t}|\mathbf{z},\tilde{\mathbf{z}})$$

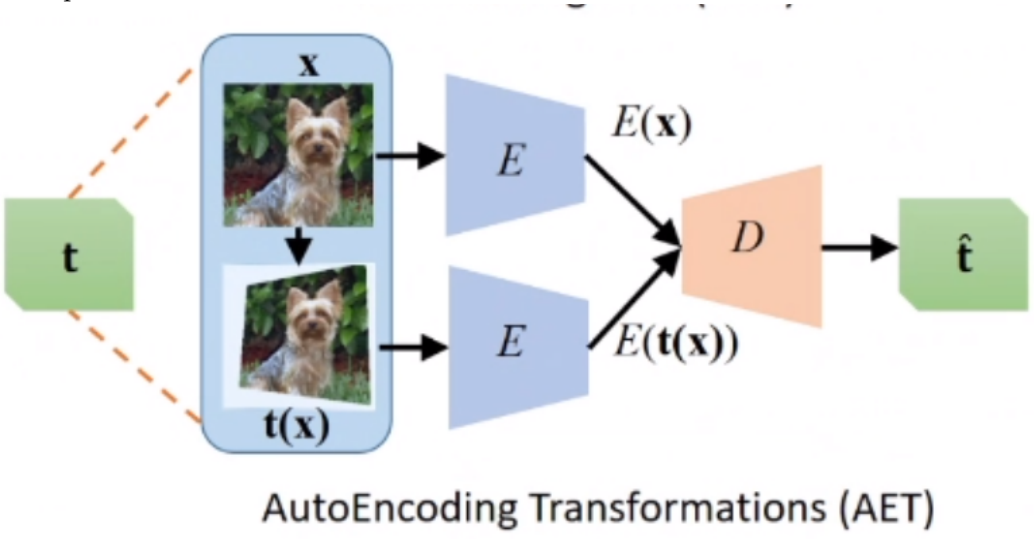
SAT:

- add a label decoder compared to AVT.
- variational surrogate ⇒cross-entropy loss on supervised data + AVT loss

Contrastive Learning: more utilized trans-invariant repr. Future: unifying trans-inv/equiv repr.

3 AET, AVT: Autoencoding Transformations

Idea autoencoders used in modeling transformations rather than images in order to learn general repr.



AutoEncoding Transformations (AET)

AET:

- use autoencoders to learn **transformations**
- trans. generated randomly for self-supervised learing
- use Siamese net as encoder backbone
- AET loss: parameterized, non-parametric, GAN-induced

Losses in AET:

- parameterized transformations: if trans. are parameterized $\mathcal{T} \in \{t_\theta | \theta \in \Theta\}$, loss can be defined as norm of param. diff.
$$l(t_\theta, t_{\hat{\theta}} = ||\theta - \hat{\theta}||.)$$
- for non-parametric trans., use expected distance on source domain
$$l(t, \hat{t}) = \mathbb{E}_{x \sim X}\{dist(t(x), \hat{t}(x)))\}$$
- GAN-induced trans.: image tranformed in form $G(x, z)$, we have loss
$$l(t_z, t_{\hat{z}} = ||z - \hat{z}||.)$$

Idea of AVT use prob. dist. to model trans., VAE like modeling!

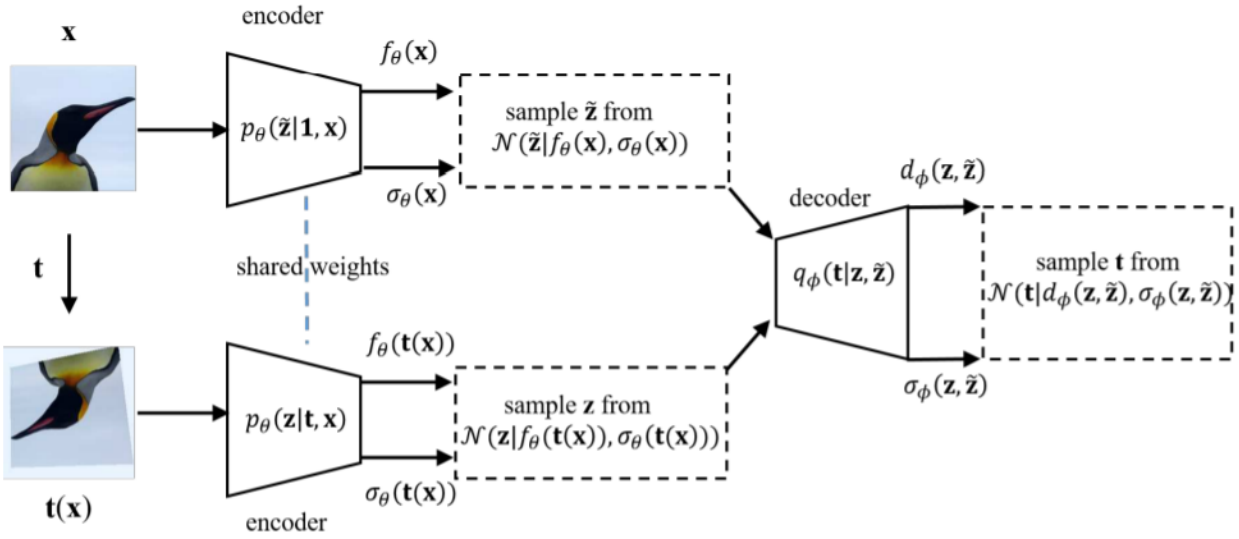


Figure 1: The architecture of the proposed AVT. The original and transformed images are fed through the encoder p_θ where $\mathbf{1}$ denotes an identity transformation to generate the representation of the original image. The resultant representations $\tilde{\mathbf{z}}$ and $\mathbf{z}$ of original and transformed images are sampled and fed into the transformation decoder q_ϕ from which the transformation $\mathbf{t}$ is sampled.

AVT:

- maximize mutual info $I(t;z|\tilde{z})$
- variational bound, introdicing a decoder $q_\theta(t|z,\tilde{z})$:

$$
\begin{aligned}
I(t;z|\tilde{z}) &= H(t|\tilde{z}) - H(t|z,\tilde{z}) && (12)\\
&= H(t|\tilde{z}) + \mathbb{E}_{p_\theta(t,z,\tilde{z})}[p_\theta(t|z,\tilde{z})] && (13)\\
&= H(t|\tilde{z}) + \mathbb{E}_{p(t,z,\tilde{z})}[q_\theta(t|z,\tilde{z})] + \mathbb{E}_{p_(z,\tilde{z})}[D(p_\theta(t,z,\tilde{z})||q_\phi(t|z,\tilde{z}))] && (14)\\
&\geq H(t|\tilde{z}) + \mathbb{E}_{p(t,z,\tilde{z})}[q_\phi(t|z,\tilde{z})] \equiv \tilde{I}(t;z|\tilde{z}) && (15)\\
&\Rightarrow \max_{\theta,\phi} \mathbb{E}_{p(t,z,\tilde{z})}[q_\phi(t|z,\tilde{z})] && (16)
\end{aligned}
$$