

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.

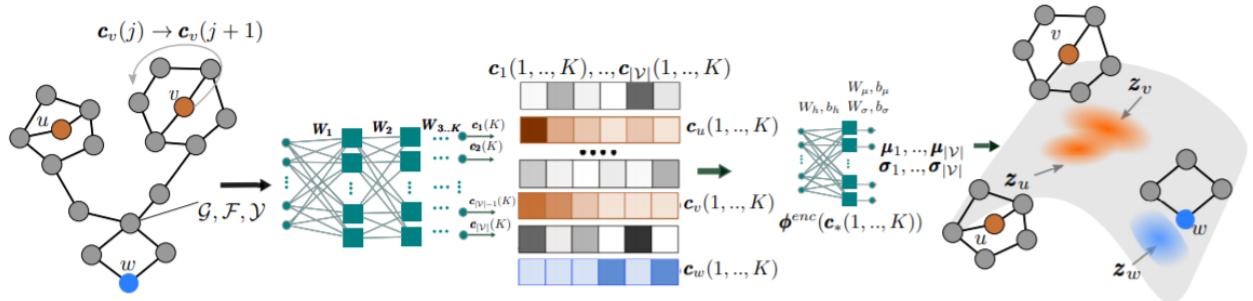


Figure 1: The encoder of our variational autoencoder for molecular graphs. From left to right, given a molecular graph \mathcal{G} with a set of node features \mathcal{F} and edge weights \mathcal{Y} , the encoder aggregates information from a different number of hops $j \leq K$ away for each node $v \in \mathcal{G}$ 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:

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

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

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

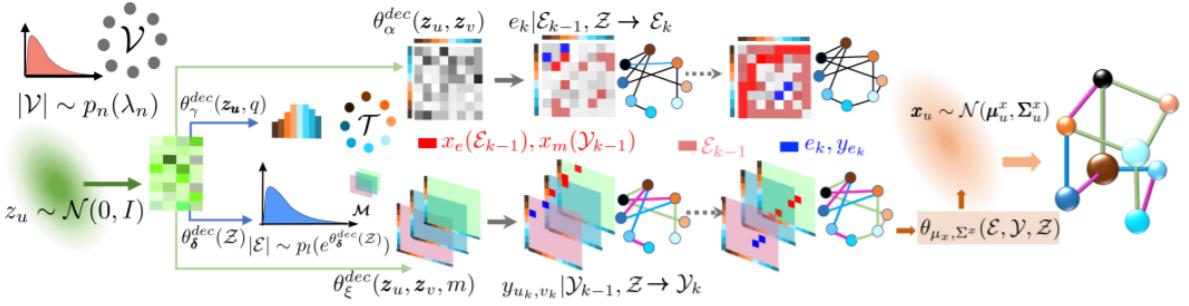


Figure 2: The decoder of our variational autoencoder for molecular graphs. From left to right, the decoder first samples the number of nodes $n = |\mathcal{V}|$ from a Poisson distribution $p_n(\lambda_n)$ and it samples a latent vector \mathbf{z}_u per node $u \in \mathcal{V}$ from $\mathcal{N}(\mathbf{0}, \mathbf{I})$. 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 \mathbf{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 $\theta_\delta^{dec}(\mathcal{Z})$ 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 $(\mathbf{z}_u, \mathbf{z}_v)$. 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 $(\mathbf{z}_u, \mathbf{z}_v, m)$. 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 \mathbf{x}_u from a multidimensional Gaussian distribution whose mean μ_x and variance Σ_x depends on the latent vectors of the corresponding atom and its neighbors and the underlying chemical bonds.

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

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

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

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

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

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

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

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

$$[\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x] = [] \quad (11)$$

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! \Rightarrow 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*

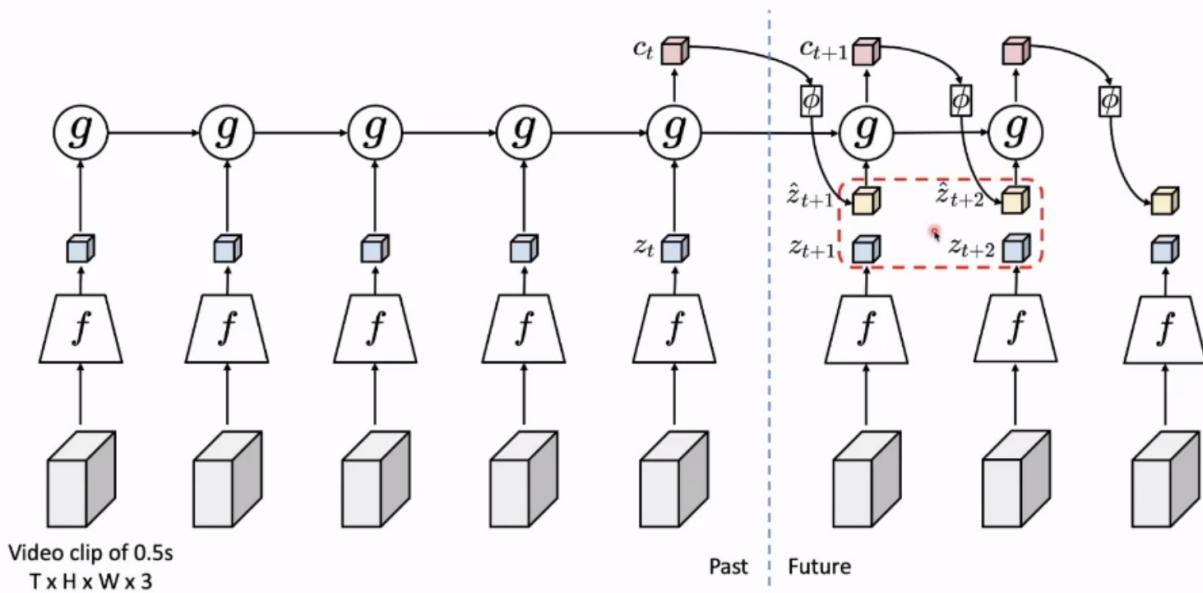
Why video-based self-supervised: like what human percepts, rich info; might with audio.

Proxy loss design: temporal info, spatial cohenrence, motions of obj., multimodal

Temporal:

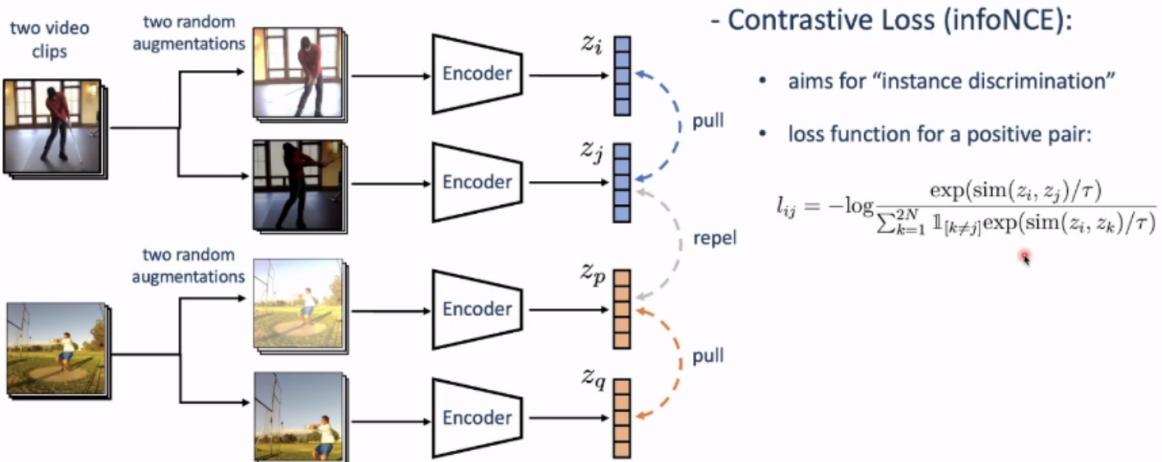
- shuffle & learn
- forward or backward?(arrow of time)
- SpeedNet: which speed(frame-rate) is normal/speed-up
- ===Weak, irrelative with downstream tasks==
- DPC: *learn repr. in predicting future in video*

Approach - DPC



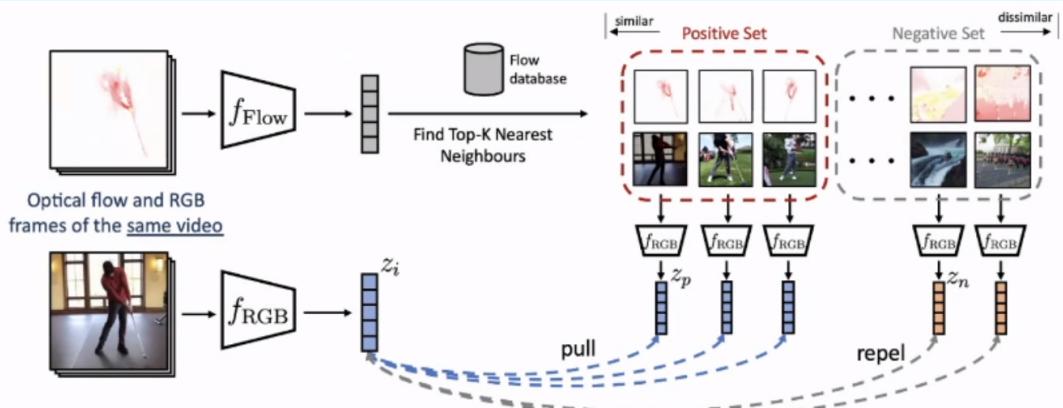
DPC Arch.:encoder-decoder like, contrast learning(infoNCE)

Self-supervised learning with videos (CoCLR)



Oord et al., ‘Representation Learning with Contrastive Predictive Coding’, arXiv:1807.03748
He et al., ‘Momentum Contrast for Unsupervised Visual Representation Learning’, CVPR2020
Chen et al., ‘A Simple Framework for Contrastive Learning of Visual Representations’, ICML2020

Self-supervised learning with videos (CoCLR)



Multi-Instance Contrastive Loss (MIL-NCE):

- Features from the positive set are pulled together
- Features **NOT** from the positive set are pushed apart
- Optical flow helps RGB frames to go beyond instance discrimination

$$\mathcal{L}_{\text{CoCLR}} = -\mathbb{E} \left[\log \frac{\sum_{p \in \mathcal{P}_i} \exp(z_i \cdot z_p)}{\sum_{p \in \mathcal{P}_i} \exp(z_i \cdot z_p) + \sum_{n \in \mathcal{N}_i} \exp(z_i \cdot z_n)} \right]$$

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 sequential segmentations

Next:

- More efficient learning
- Scale up model to uncurated 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 \Rightarrow Auto proxy task design?
- Theoretic: 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:

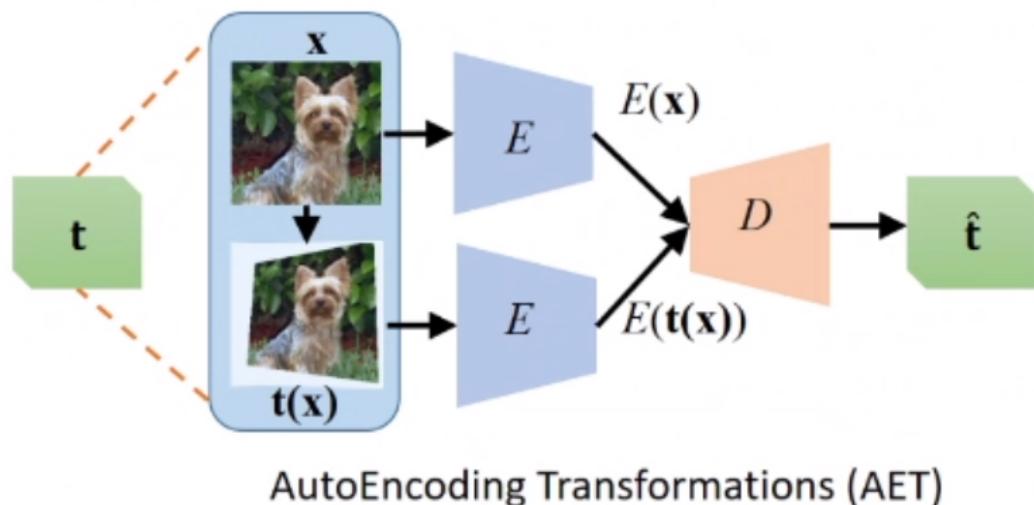
- TER(Transformation Equivariance Repr.)
- AET(AutoEncoding Transformation)
- AVT(Autoencoding Variational Transformation)
- SAT(Semi-supervised Autoencoding Transformation)

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

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

Steerability: ρ is inpend. with sample \mathbf{x} .

Targets: Non-linear ρ , General Transformation(e.g. recoloring)



AET loss for training

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

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

- GAN-Induced Transformations: transformed image $G(x, z)$

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

- Non-Parametric Transformations

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

Activ
Go to

AET:

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

An Information-Theoretical Insight

- Train a TER model θ by maximizing
$$\max_{\theta} I_{\theta}(z; \tilde{z}, t)$$
- By chain rule of mutual information, we have

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

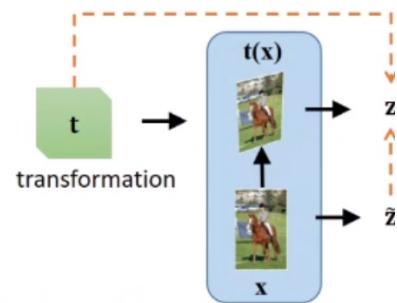
- $I_{\theta}(z; \tilde{z}, t)$ attains its maximum value $I_{\theta}(z; \tilde{z}, t, x)$ (the upper bound) when

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

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

- **Nonlinearity** of transformation $p(t)$ in representations.

Activate Windows
Go to Settings to activate Windows.



AVT: Autoencoding Variational Transformations

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

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

- Unsupervised loss to learn AVT

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

AVT: Autoencoding Variational Transformations

- Generative process

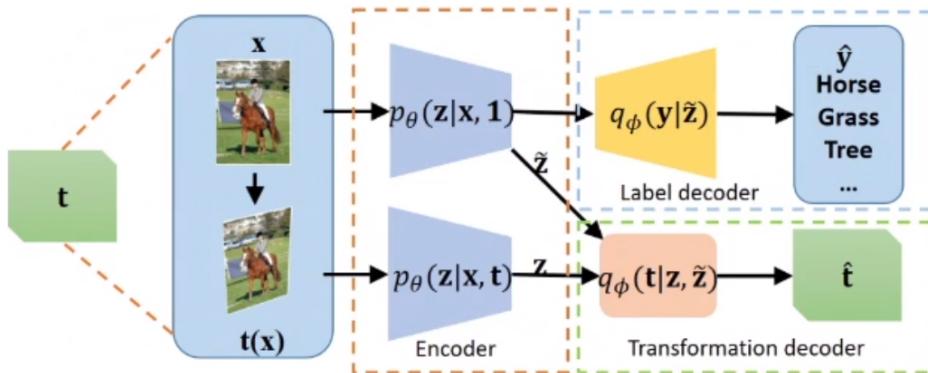
- Given an image \mathbf{x} sampled from $p(\mathbf{x})$
- Sample a transformation \mathbf{t} from $p(\mathbf{t})$
- Apply \mathbf{t} to \mathbf{x} , resulting in $\mathbf{t}(\mathbf{x})$
- Sample a representation \mathbf{z} of $\mathbf{t}(\mathbf{x})$ from $p_\theta(\mathbf{z}|\mathbf{x}, \mathbf{t})$
 - $\tilde{\mathbf{z}}$ 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 maximize $I_\theta(z; \tilde{z}, t)$

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

- Jointly maximizing over encoder θ and decoders ϕ

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

Activate Windows

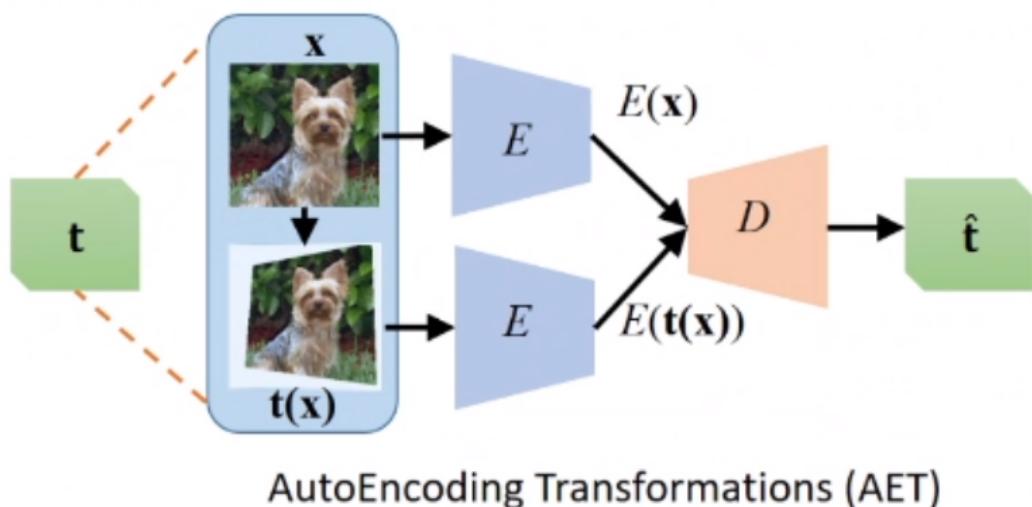
SAT:

- add a label decoder compared to AVT.
 - variational surrogate \Rightarrow 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.



AET:

- use autoencoders to learn **transformations**
- trans. generated randomly for self-supervised learning
- 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 transformed in form $G(x, z)$, we have loss $l(t_z, \hat{t}_z) = \|z - \hat{z}\|.$)

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

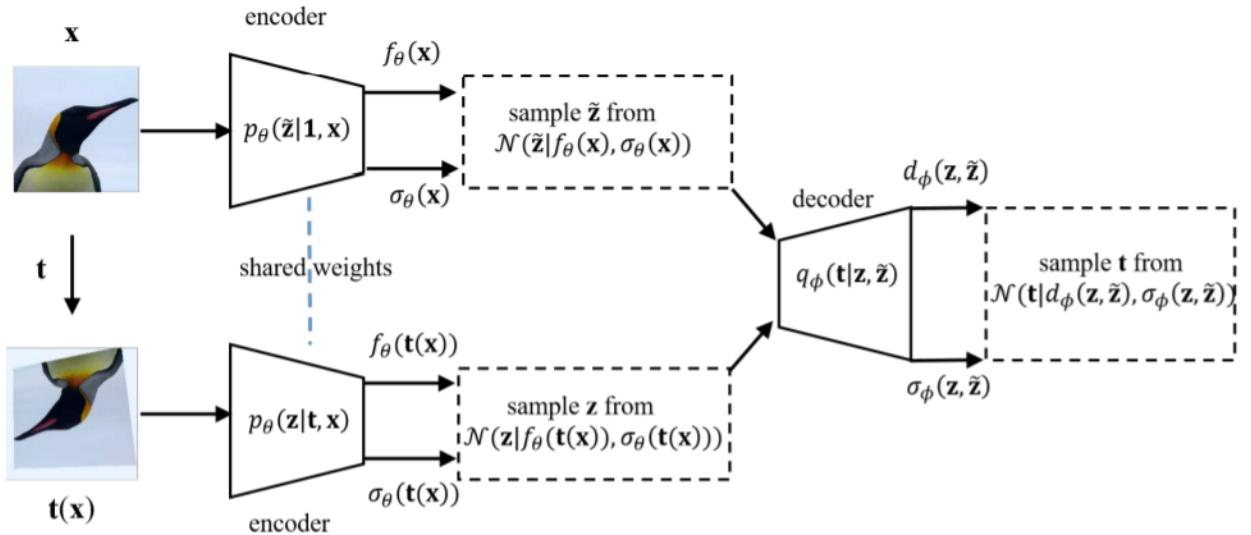


Figure 1: The architecture of the proposed AVT. The original and transformed images are fed through the encoder p_θ where $\mathbf{1}$ denotes an identity transformation to generate the representation of the original image. The resultant representations $\tilde{\mathbf{z}}$ and \mathbf{z} of original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