Vehicle Interaction Learning

Xiaosong Jia

07/12/2019

Preprocess Data

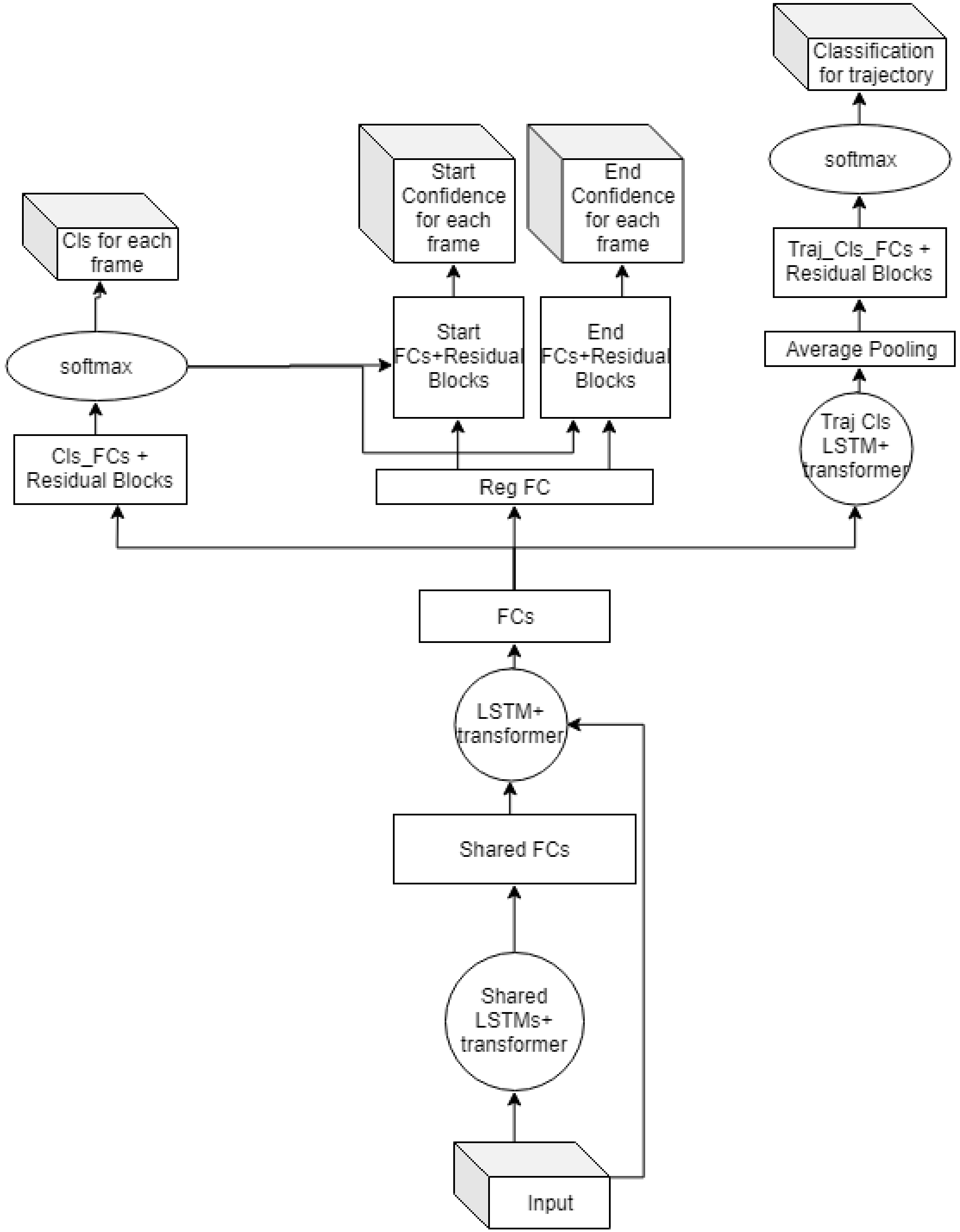
- Samples without interactions: too much (Tracks 1: 16157:47) -> try 1:1

- Symmetry: Overall Recognition vs Relative Motion

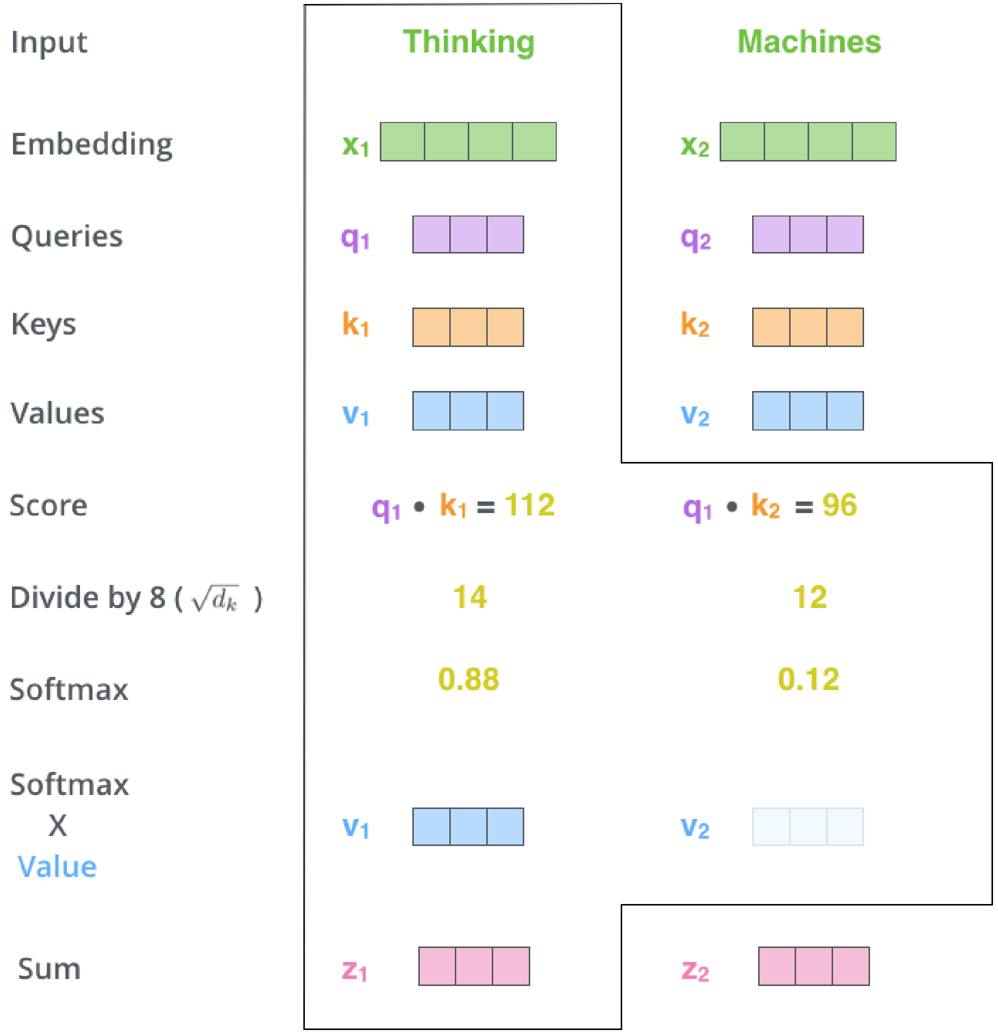
- Overall Recognition -> 1. Feed samples and their mirrors together into the neural network 2. Use (frame[0].x1 + frame[0].x2)/2 as ref

Model

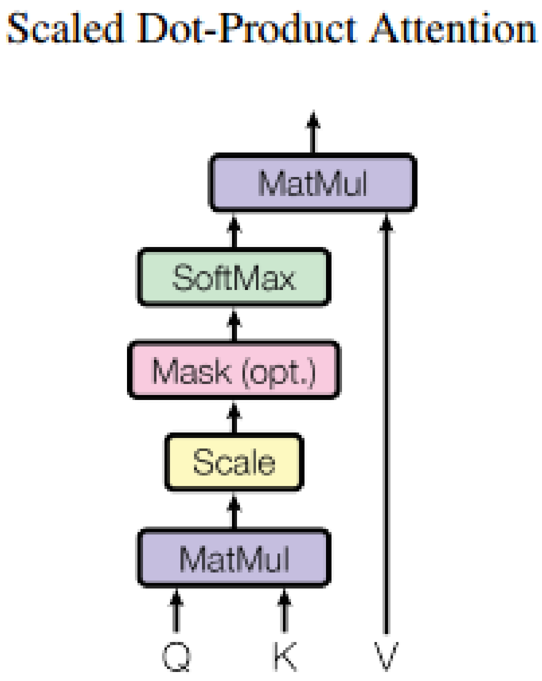
- Shared encoder
- Residual blocks make converge faster
- Transformer
- Interaction between tasks
- Conditional loss function
- Has interaction: Loss = w1 * cls_loss + w2 * reg_loss +w3 * traj_cls_loss
- Not: Loss = traj_cls_loss
- Huge Model -> Slow
- Sol: Pad+Pack for variable-length sequences to do mini-batch SGD
- -> ~600s for 10304 samples each epoch (batch size=16)

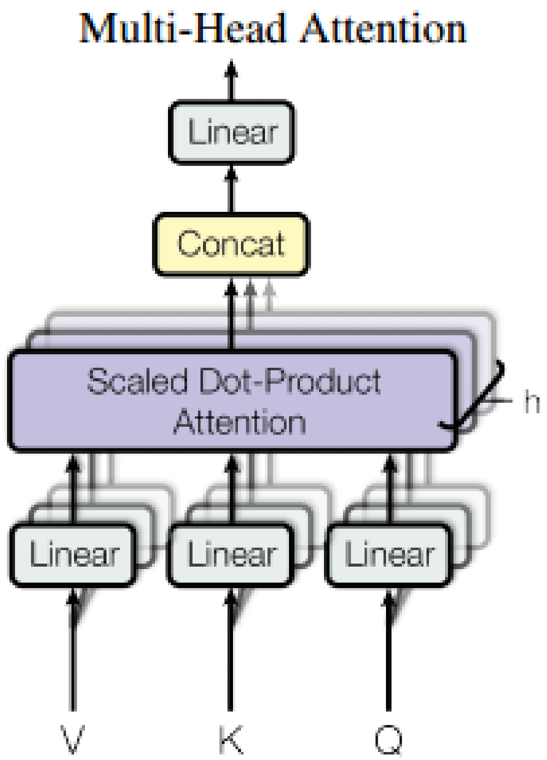


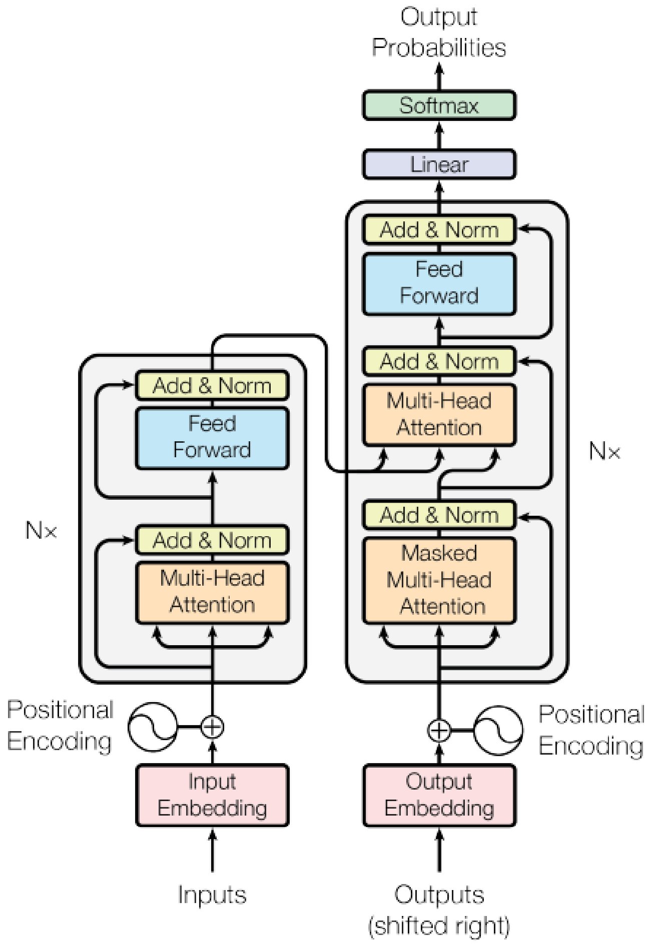
Transformer [Vaswani et.al. NIPS 2017]



Transformer [Vaswani et.al. NIPS 2017]



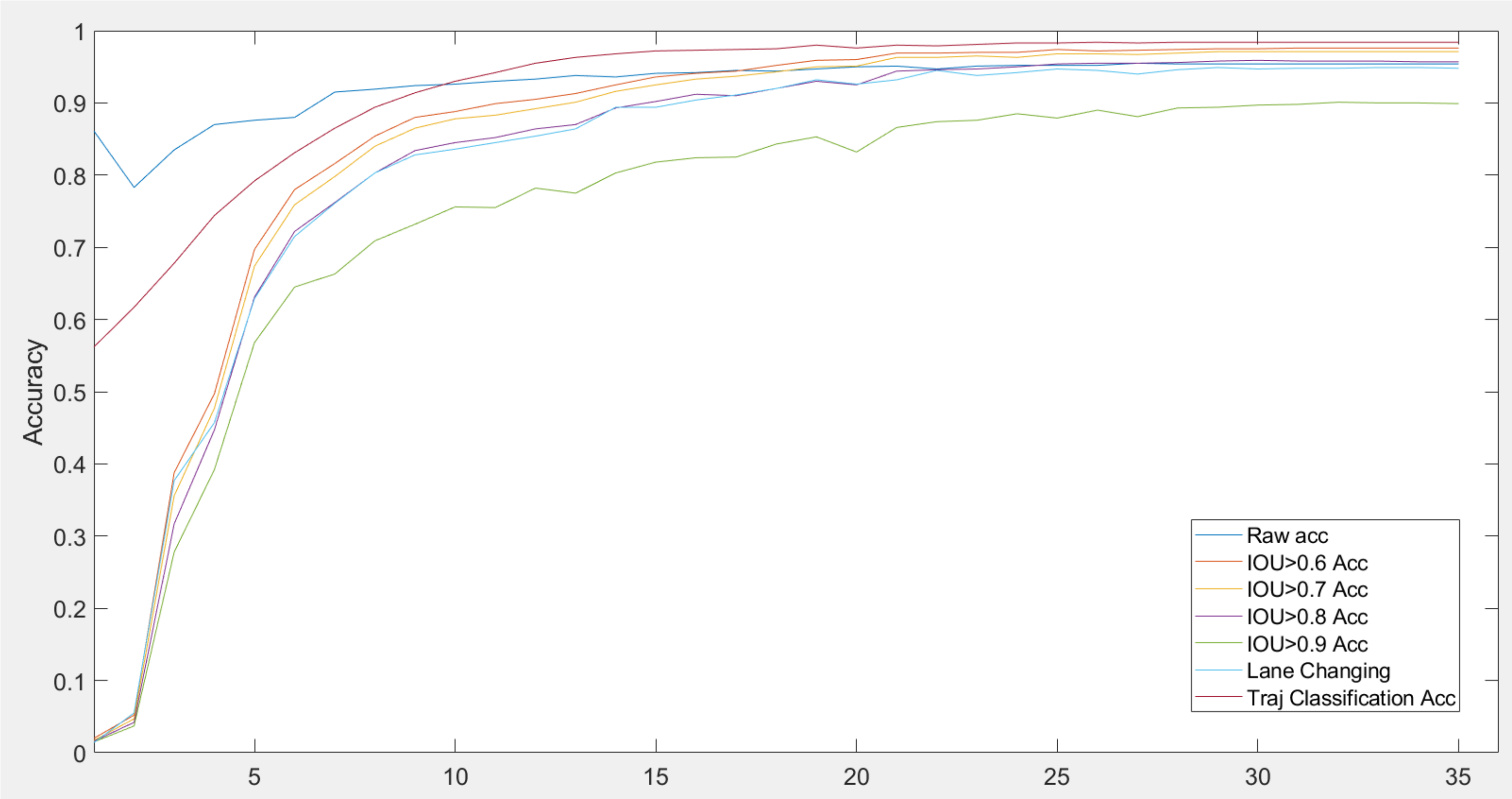




Result

w1 = 1, w2=0->10 from epoch 1 to epoch 20, w3 = 1; Total 30 epochs

10304 samples (Has interaction : Not = 1:1)
8242 training samples, 2062 test samples



Metric	Best
Raw Acc	95.4%
IOU>0.6 Acc	97.6%
IOU>0.7 Acc	97.1%
IOU>0.8 Acc	95.7%
IOU>0.9 Acc	90.0%
Lane Changing	95.8%
Traj Classification Acc	98.4%

Future Work

- 1. Entirely transformer model + window model
- 2. All datasets + unsupervised/semi-supervised learning

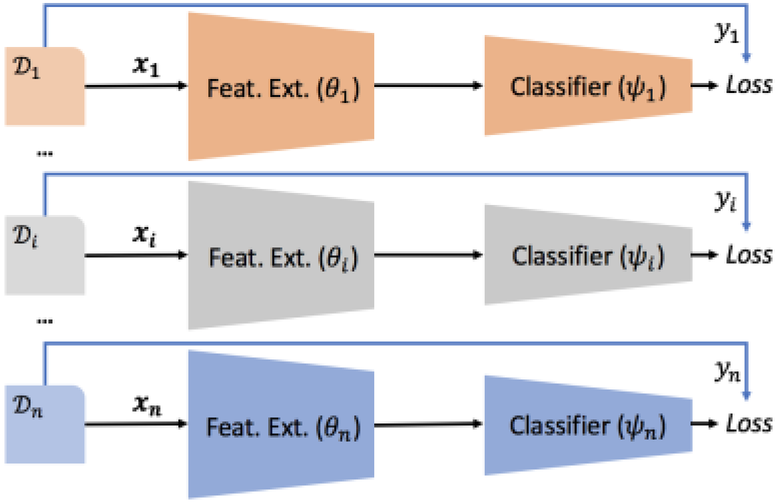
Episodic Training [Da Li et. al. 2019.01 arXiv]

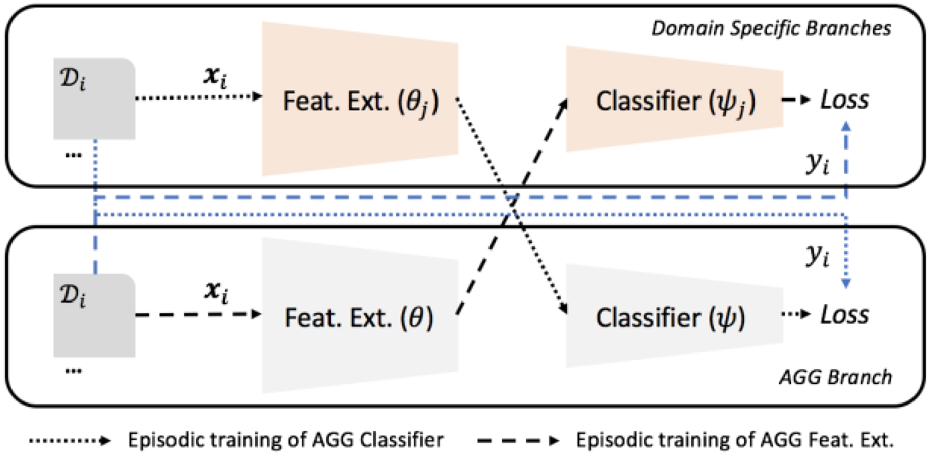
$$\underset{\theta}{\operatorname{argmin}} \mathbb{E}_{i,j\sim[1,n],i\neq j}\left[\mathbb{E}_{(\mathbf{x}_i,y_i)\sim\mathcal{D}_i}\left[\ell(y_i,\overline{\psi}_j(\theta(\mathbf{x}_i)))\right]\right]$$

$$\underset{\psi}{\operatorname{argmin}} \mathbb{E}_{i,j\sim[1,n],i\neq j}\left[\mathbb{E}_{(\mathbf{x}_i,y_i)\sim\mathcal{D}_i}\left[\ell(y_i,\psi(\overline{\theta}_j(\mathbf{x}_i)))\right]\right]$$

Algorithm 1 Episodic training

1: **Input**: $\mathcal{D} = [\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_n]$
2: **Initialise hyper parameters**: $\lambda_1, \lambda_2, \lambda_3, \alpha$
3: **Initialise model parameters**: domain specific modules $\theta_1, ..., \theta_n$ and $\psi_1, ..., \psi_n$; AGG modules θ, ψ; random classifier ψ_r
4: **while** not done training **do**
5:     **for** $(\theta_i, \psi_i) \in [(\theta_1, \psi_1), ..., (\theta_n, \psi_n)]$ **do**
6:         Update $\theta_i := \theta_i - \alpha\nabla_{\theta_i}(L_{ds})$
7:         Update $\psi_i := \psi_i - \alpha\nabla_{\psi_i}(L_{ds})$
8:     **end for**
9:     Update $\theta := \theta - \alpha\nabla_\theta(L_{agg} + \lambda_1 L_{epif} + \lambda_3 L_{epir})$
10:     Update $\psi := \psi - \alpha\nabla_\psi(L_{agg} + \lambda_2 L_{epic})$
11: **end while**
12: **Output**: θ, ψ

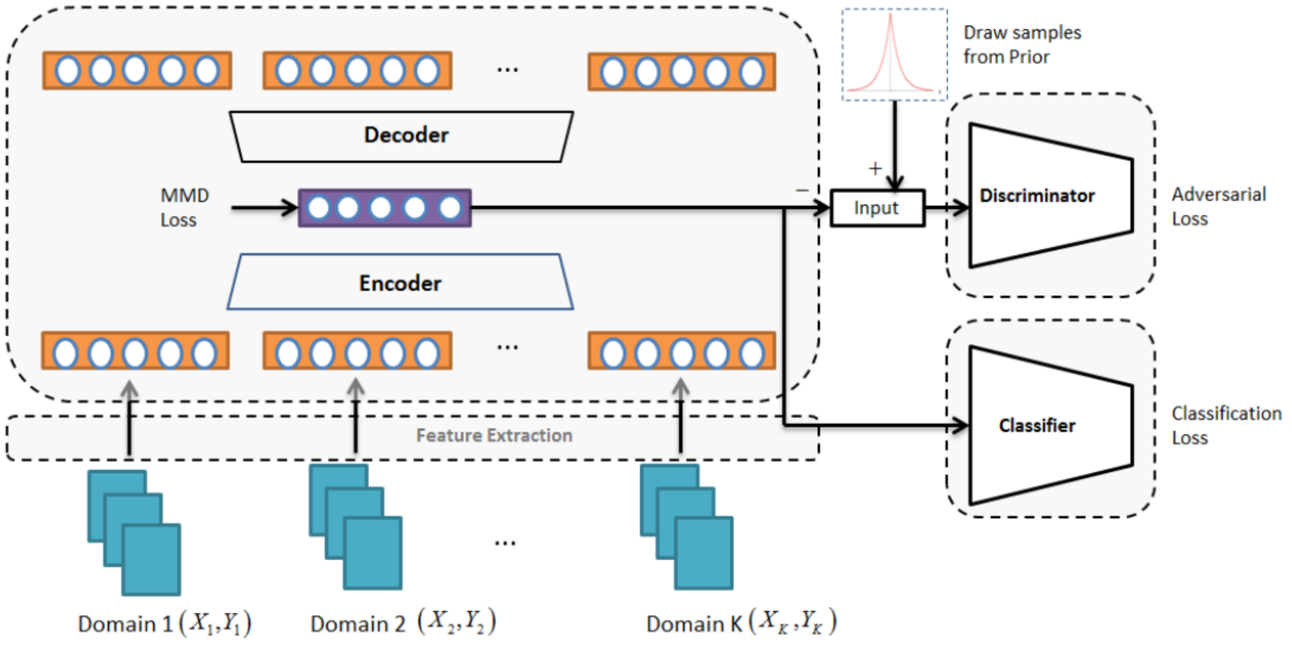




MMD-AAE [Haoliang Li et. al. CVPR 2018]

$$\min_{C,Q,P}\max_{D}\mathcal{L}_{\text{err}}+\lambda_0\mathcal{L}_{\text{ae}}+\lambda_1\mathcal{R}_{\text{mmd}}+\lambda_2\mathcal{J}_{\text{gan}}$$

- Reconstruction
- Task
- GAN
- Domain Generalizaiton



Unsupervised Learning (e.g. BERT)

- Unsupervised Pretrain
- Supervised Finetune
- Tremendous Data