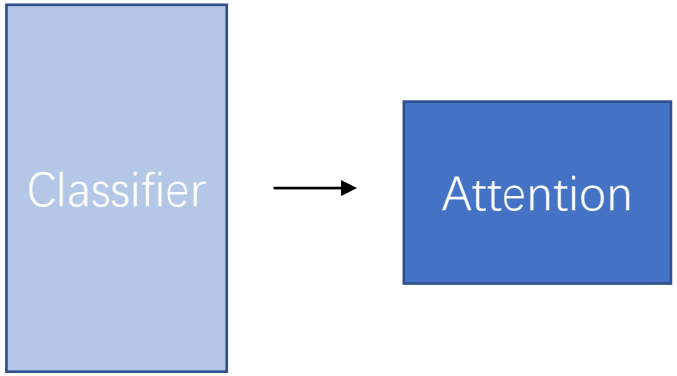
CVPR2020

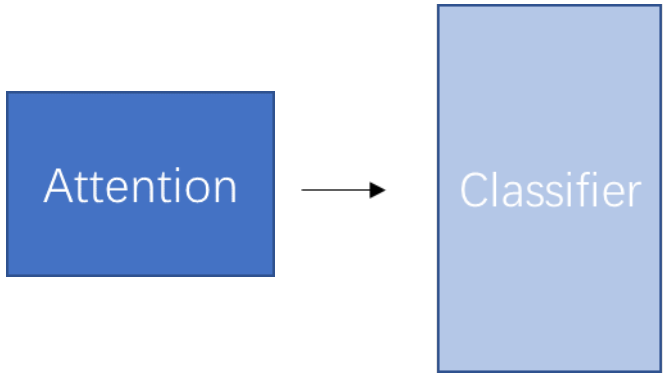
Weakly-Supervised Action Localization by Generative Attention Modeling

Baifeng Shi[1*] Qi Dai[2] Yadong Mu[1] Jingdong Wang[2]
[1]Peking University [2]Microsoft Research Asia
{bfshi,myd}@pku.edu.cn, {qid,jingdw}@microsoft.com

Background

- WSAL methods:
  - Top-down: learns a video-level classifier and then obtains frame attention by checking the produced temporal class activation map;
  - Bottom-up: temporal attention is directly predicted from raw data. Attention is optimized in the task of video classification with video-level supervision.





Motivation

- Above methods result in the action-context confusion issue: context frames near action clips tend to be recognized as action frames themselves, since they are closely related to the specific classes.
- With the observation that the context exhibits notable difference from the action at representation level, a probabilistic model, i.e., conditional VAE, is learned to model the likelihood of each frame given the attention.

Method

- Feature: $X = (x_t)_{t=1}^{T}, \quad \mathrm{x_t} \in \mathbb{R}^d$
- Video-level label: $y \in \{0, 1, 2, \ldots, C\}$, C is the number of classes and 0 corresponds to background
- Attention: $\lambda = (\lambda_t)_{t=1}^{T}$
- In attention-based action localization problem, the target is to predict the frame attention, which is equivalent to solving the maximum a posteriori (MAP) problem:

$$\max_{\lambda_t \in [0,1]} \log p(\boldsymbol{\lambda}|\mathbf{X}, y)$$

$$\begin{aligned}
\log p(\boldsymbol{\lambda}|\mathbf{X}, y) &= \log p(\mathbf{X}, y|\boldsymbol{\lambda}) + \log p(\boldsymbol{\lambda}) - \log p(\mathbf{X}, y) \\
&= \log p(y|\mathbf{X}, \boldsymbol{\lambda}) + \log p(\mathbf{X}|\boldsymbol{\lambda}) + \log p(\boldsymbol{\lambda}) \\
&\quad - \log p(\mathbf{X}, y) \\
&\propto \log p(y|\mathbf{X}, \boldsymbol{\lambda}) + \log p(\mathbf{X}|\boldsymbol{\lambda})
\end{aligned}$$

- By discarding the constant term, the optimization problem becomes:

$$\max_{\lambda \in [0,1]} \log p(y|\mathbf{X}, \boldsymbol{\lambda}) + \log p(\mathbf{X}|\boldsymbol{\lambda})$$

Method

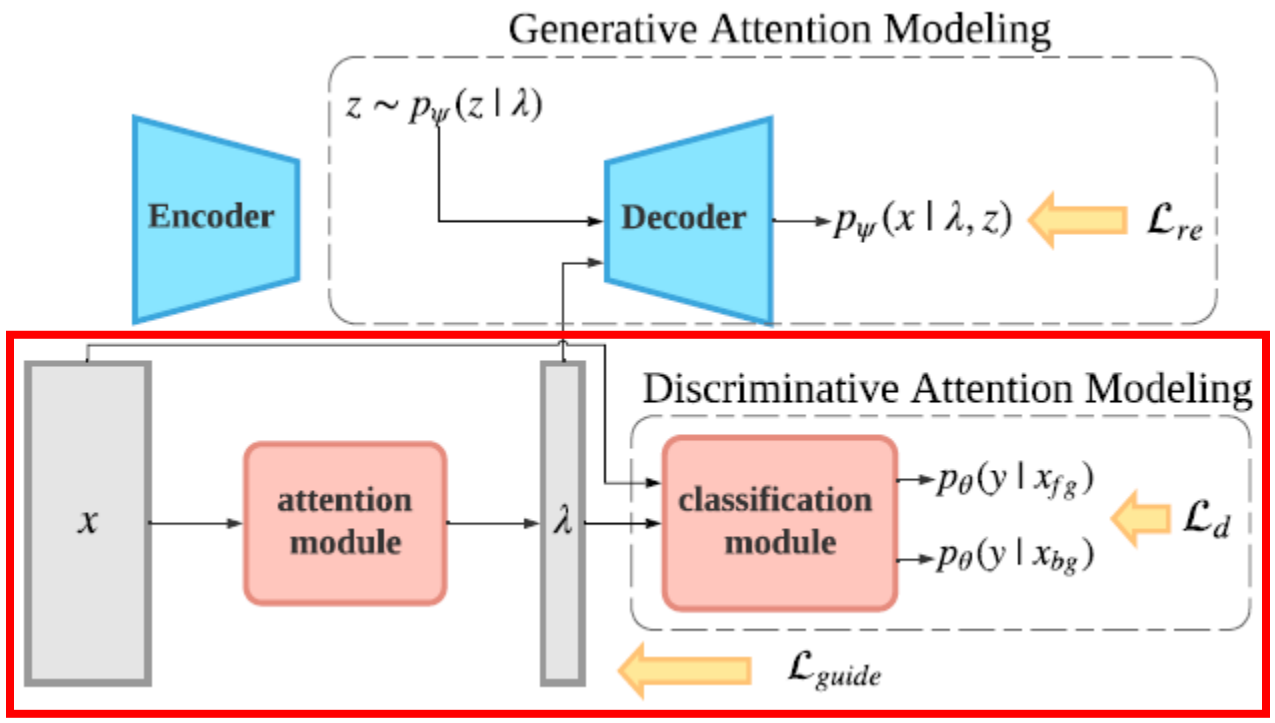
$$\max_{\lambda \in [0,1]} \log p(y|\mathbf{X}, \boldsymbol{\lambda}) + \log p(\mathbf{X}|\boldsymbol{\lambda})$$

- The first term $\log p(y|X, \lambda)$ prefers λ with high discriminative capacity for action classification, which is the main optimization target in previous works.
- The second term $\log p(X|\lambda)$ forces the representation of frames to be accurately predicted from the attention λ, this objective encourages the model to impose different attentions on different features.

Architecture

Discriminative Attention Modeling: learn the frame attention by optimizing the video-level recognition task.

Generative Attention Modeling: generate the representation based on the attention.



Discriminative Attention Modeling

Video-level foreground feature: utilize attention λ as weight to perform temporal average pooling over all frames in the video.

$$\mathbf{x}_{fg} = \frac{\sum_{t=1}^{T} \lambda_t \mathbf{x}_t}{\sum_{t=1}^{T} \lambda_t}$$

Video-level background feature: utilize attention $1 - \lambda$ as weight to perform temporal average pooling over all frames in the video.

$$\mathbf{x}_{bg} = \frac{\sum_{t=1}^{T} (1 - \lambda_t) \mathbf{x}_t}{\sum_{t=1}^{T} (1 - \lambda_t)}$$

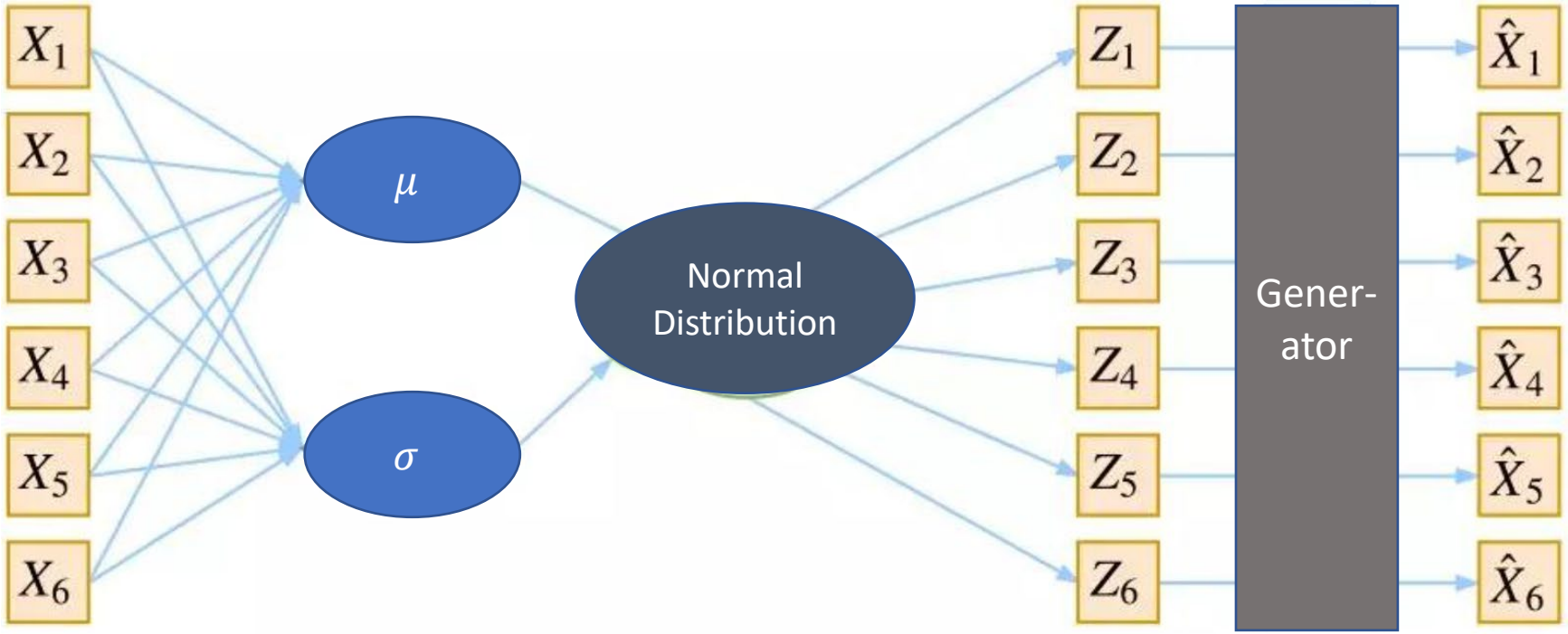
Discriminative loss: encourage high discriminative capability of the foreground feature and simultaneously punish any discriminative capability of the background feature.

$$\mathcal{L}_d = \mathcal{L}_{fg} + \alpha \cdot \mathcal{L}_{bg} = -\log p_\theta(y|\mathbf{x}_{fg}) - \alpha \cdot \log p_\theta(0|\mathbf{x}_{bg})$$

VAE

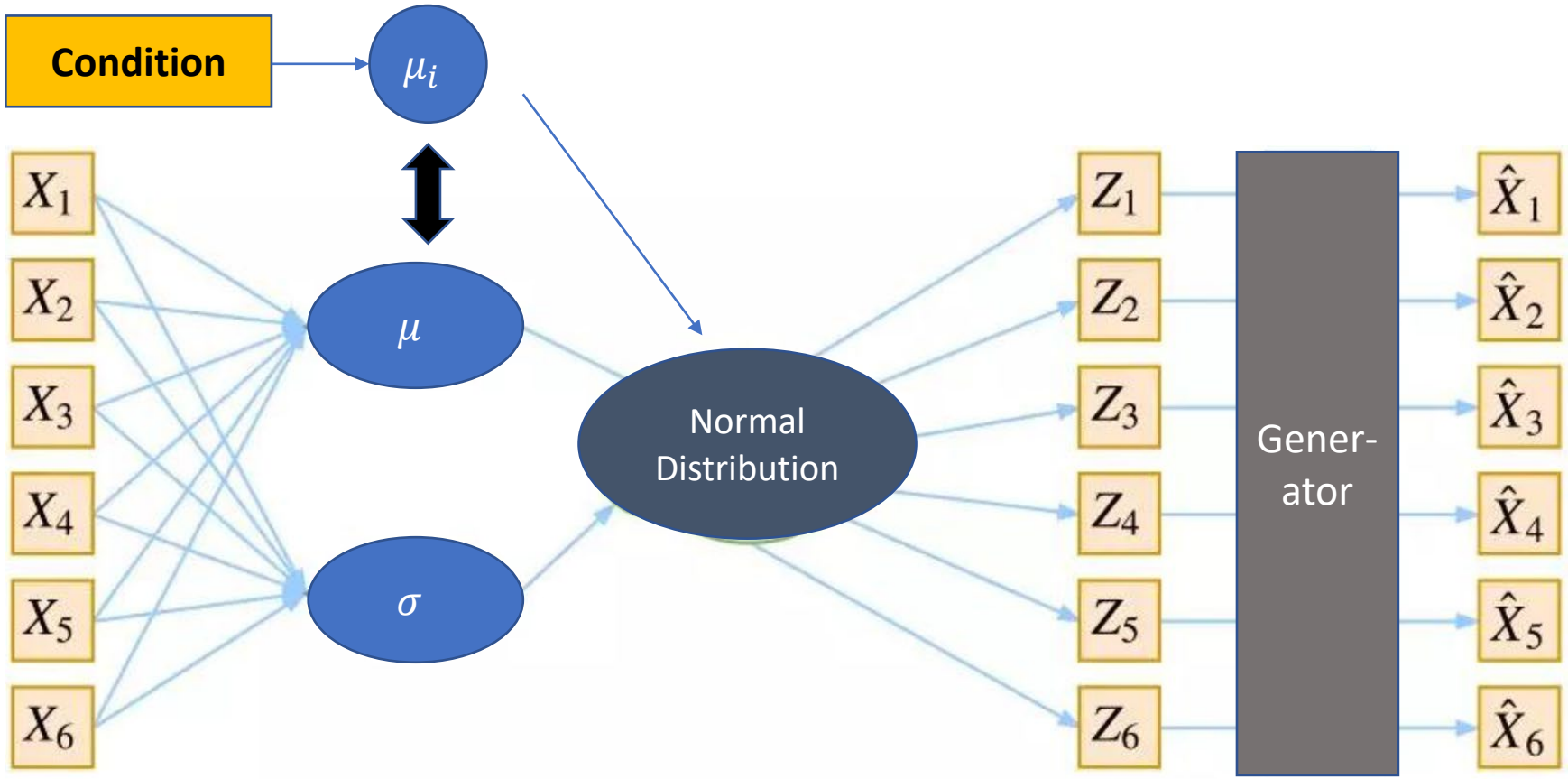
Variational Auto-Encoder: learn the mapping between probability distributions.

Loss function: $\mathcal{J}_{VAE} = -KL(q_\phi(\mathbf{z}|\mathbf{x})||p_\psi(\mathbf{z})) + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\psi(\mathbf{x}|\mathbf{z})]$



CVAE

Conditional Variational Auto-Encoder: learn the mapping between probability distributions.



Generative Attention Modeling

GAM: generate the representation based on the attention. By assuming independence between frames in a video, we get:

$$p(\mathbf{X}|\boldsymbol{\lambda}) = \Pi_{t=1}^{T} p(\mathbf{x}_t|\lambda_t)$$

CVAE: introduce a latent variable z_t, and attempt to generate each x_t from z_t and λ_t.

$$p_\psi(\mathbf{x}_t|\lambda_t) = \mathbb{E}_{p_\psi(\mathbf{z}_t|\lambda_t)}[p_\psi(\mathbf{x}_t|\lambda_t, \mathbf{z}_t)]$$

Loss function:

$$\begin{aligned}\mathcal{L}_{CVAE} &= -\mathbb{E}_{q_\phi(\mathbf{z}_t|\mathbf{x}_t,\lambda_t)} \log p_\psi(\mathbf{x}_t|\lambda_t, \mathbf{z}_t) \\ &\quad + \beta \cdot KL(q_\phi(\mathbf{z}_t|\mathbf{x}_t, \lambda_t)||p_\psi(\mathbf{z}_t|\lambda_t)) \\ &\simeq -\frac{1}{L}\sum_{l=1}^{L} \log p_\psi(\mathbf{x}_t|\lambda_t, \mathbf{z}_t^{(l)}) \\ &\quad + \beta \cdot KL(q_\phi(\mathbf{z}_t|\mathbf{x}_t, \lambda_t)||p_\psi(\mathbf{z}_t|\lambda_t))\end{aligned}$$

Optimization

Temporal class activation maps: given a video with label y, the TCAM are computed by

$$\hat{\lambda}_t^{fg} = G(\sigma_s) * \frac{\exp^{\mathbf{w}_y^T \mathbf{x}_t}}{\sum_{c=0}^{C} \exp^{\mathbf{w}_c^T \mathbf{x}_t}}$$

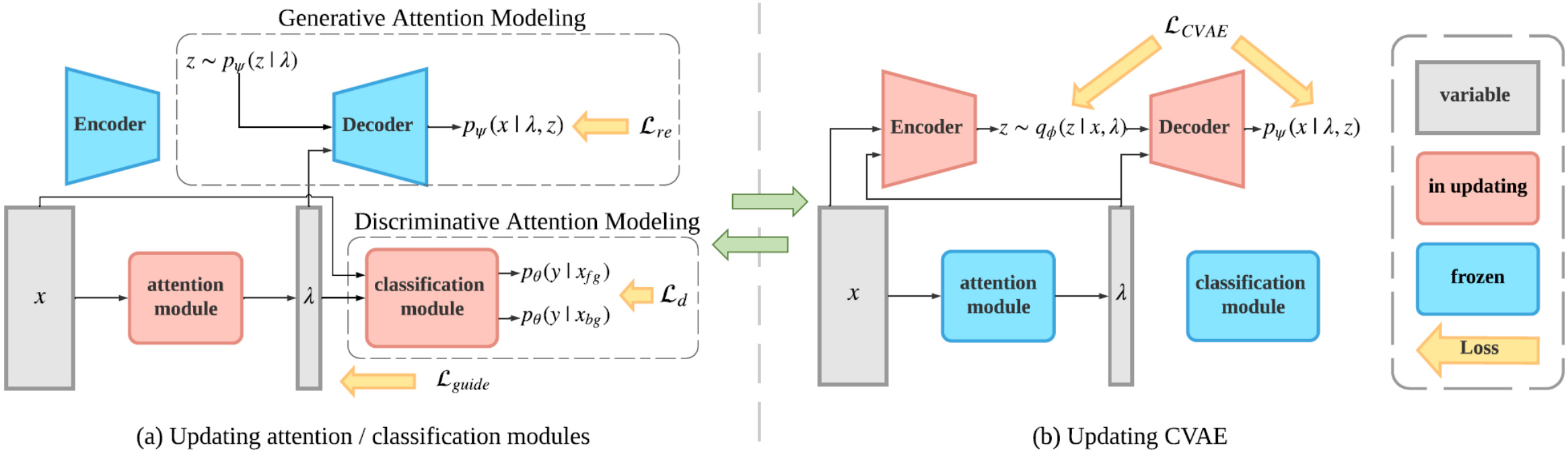
$$\hat{\lambda}_t^{bg} = G(\sigma_s) * \frac{\sum_{c=1}^{C} \exp^{\mathbf{w}_c^T \mathbf{x}_t}}{\sum_{c=0}^{C} \exp^{\mathbf{w}_c^T \mathbf{x}_t}}$$

Loss function: The generated $\hat{\lambda}_t^{fg}$ and $\hat{\lambda}_t^{bg}$ are expected to be consistent with the bottom-up, class-agnostic attention λ, hence the loss function can be formulated as:

$$\mathcal{L}_{guide} = \frac{1}{T} \sum_{t=1}^{T} |\lambda_t - \hat{\lambda}_t^{fg}| + |\lambda_t - \hat{\lambda}_t^{bg}|$$

Training process

1. Update attention and classification modules with loss $L = L_d + \gamma_1 L_{re} + \gamma_2 L_{guide}$, where γ_1, γ_2 denote the hyper-parameters. L_{re} only has the first term of L_{CVAE}.
2. Update CVAE with loss L_{CVAE}.



Experiments

- **THUMOS14** contains videos from 20 classes for action localization task. Each video contains 15.5 action clips on average. Length of action instance varies widely, from a few seconds to minutes. Video length also ranges from a few seconds to 26 minutes, with an average of around 3 minutes.
- **ActivtyNet1.2** contains 100 classes of videos with both video-level labels and temporal annotations. Each video contains 1.5 action instances on average.
- **Evaluation Metrics:** mean Average Precision

Experiments

Table 1: Attention evaluation on THUMOS14. The "Old" model (O) is trained without the generative attention modeling, and the "New" model (N) is our DGAM. We assemble specific models by alternately choosing Attention (Att) and Classification (Cls) modules from the two models.

Att	Cls	mAP@IoU				
		0.3	0.4	0.5	0.6	0.7
O	O	43.8	35.8	26.7	18.2	9.7
O	N	44.2	36.1	27.0	18.7	9.8
N	O	46.1	38.2	28.8	19.4	11.2
N	N	46.8	38.2	28.8	19.8	11.4

Table 2: Statistics comparison on THUMOS14 with/without generative attention modeling. ↓ indicates lower is better, ↑ indicates higher is better. For details of notation, please refer to Section 4.3.

Metric		w/o	w/
$\lvert att - gt\rvert \,/\, \lvert gt\rvert$	↓	0.777	**0.698**
$\lvert gt - att\rvert \,/\, \lvert gt\rvert$	↓	0.858	**0.707**
$\lvert (cls - gt) \cap \overline{att}\rvert \,/\, \lvert gt\rvert$	↑	1.522	**1.543**
$\lvert (att \cap gt) - cls\rvert \,/\, \lvert gt\rvert$	↑	**0.001**	**0.001**

Ablation Studies

Table 4: Contribution of each design in DGAM on THUMOS14. Note that when adding $\mathcal{L}_{re}$, $\mathcal{L}_{CVAE}$ is involved simultaneously.

$\mathcal{L}_{fg}$	$\mathcal{L}_{bg}$	$\mathcal{L}_{guide}$	$\mathcal{L}_{re}$	mAP@0.5
✓	-	-	-	21.5
✓	✓	-	-	24.8
✓	✓	✓	-	26.7
✓	✓	✓	✓	28.8

Table 6: Evaluation on dimension of latent space on THUMOS14. We experiment with different dimensions of 2^n, $n = 4, 5, \cdots, 9$.

$\log_2$(dim)	4	5	6	7	8	9
mAP@0.5	26.5	27.5	28.0	28.8	28.3	27.7

Table 7: Evaluation on parameter for reconstruction-sampling trade-off in CVAE. mAP@0.5 is reported on THUMOS14.

β	0.01	0.03	0.07	0.1	0.3	0.7
mAP@0.5	28.2	28.1	28.4	28.8	28.0	28.4

Experiments

Table 3: Results on THUMOS14 testing set. We report mAP values at IoU thresholds 0.1:0.1:0.9. Recent works in both fully-supervised and weakly-supervised settings are reported. UNT and I3D represent UntrimmedNet and I3D feature extractor, respectively. Our method outperforms the state-of-the-art methods, especially at high IoU threshold, which means that our model could produce finer and more precise predictions. Compared to fully-supervised methods, our DGAM can achieve close or even better performance.

Method	Supervision	Feature	mAP@IoU								
			0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
S-CNN [42]	Full	-	47.7	43.5	36.3	28.7	19.0	10.3	5.3	-	-
R-C3D [52]	Full	-	54.5	51.5	44.8	35.6	28.9	-	-	-	-
SSN [58]	Full	-	66.0	59.4	51.9	41.0	29.8	-	-	-	-
Chao *et al.* [5]	Full	-	59.8	57.1	53.2	48.5	42.8	33.8	20.8	-	-
BSN [23]	Full	-	-	-	53.5	45.0	36.9	28.4	20.0	-	-
P-GCN [56]	Full	-	**69.5**	**67.8**	**63.6**	**57.8**	**49.1**	-	-	-	-
Hide-and-Seek [44]	Weak	-	36.4	27.8	19.5	12.7	6.8	-	-	-	-
UntrimmedNet [49]	Weak	-	44.4	37.7	28.2	21.1	13.7	-	-	-	-
Zhong *et al.* [59]	Weak	-	45.8	39.0	31.1	22.5	15.9	-	-	-	-
AutoLoc [41]	Weak	UNT	-	-	35.8	29.0	21.2	13.4	5.8	-	-
CleanNet [26]	Weak	UNT	-	-	37.0	30.9	23.9	13.9	7.1	-	-
STPN [30]	Weak	I3D	52.0	44.7	35.5	25.8	16.9	9.9	4.3	1.2	0.1
MAAN [55]	Weak	I3D	59.8	50.8	41.1	30.6	20.3	12.0	6.9	2.6	0.2
W-TALC [34]	Weak	I3D	55.2	49.6	40.1	31.1	22.8	-	7.6	-	-
Liu *et al.* [24]	Weak	I3D	57.4	50.8	41.2	32.1	23.1	15.0	7.0	-	-
TSM [54]	Weak	I3D	-	-	39.5	-	24.5	-	7.1	-	-
3C-Net [29]	Weak	I3D	56.8	49.8	40.9	32.3	24.6	-	7.7	-	-
Nguyen *et al.* [31]	Weak	I3D	**60.4**	**56.0**	46.6	37.5	26.8	17.6	9.0	3.3	**0.4**
DGAM	Weak	I3D	60.0	54.2	**46.8**	**38.2**	**28.8**	**19.8**	**11.4**	**3.6**	**0.4**

Experiments

Table 5: Results on ActivityNet1.2 validation set. We report mAP at different IoU thresholds and mAP@AVG (average mAP on thresholds 0.5:0.05:0.95). Note that * indicates utilization of weaker feature extractor than others. Our method outperforms state-of-the-art methods by a large margin, where an improvement of 2% is made on mAP@AVG. Our result is also comparable to fully-supervised models.

Method	Supervision	mAP@IoU										
		0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	AVG
SSN [58]	Full	41.3	38.8	35.9	32.9	30.4	27.0	22.2	18.2	13.2	6.1	26.6
UntrimmedNet* [49]	Weak	7.4	6.1	5.2	4.5	3.9	3.2	2.5	1.8	1.2	0.7	3.6
AutoLoc* [41]	Weak	27.3	24.9	22.5	19.9	17.5	15.1	13.0	10.0	6.8	3.3	16.0
W-TALC [34]	Weak	37.0	33.5	30.4	25.7	14.6	12.7	10.0	7.0	4.2	1.5	18.0
TSM [54]	Weak	28.3	26.0	23.6	21.2	18.9	17.0	14.0	11.1	7.5	3.5	17.1
3C-Net [29]	Weak	35.4	-	-	-	22.9	-	-	-	8.5	-	21.1
CleanNet [26]	Weak	37.1	33.4	29.9	26.7	23.4	20.3	17.2	13.9	9.2	5.0	21.6
Liu *et al.* [24]	Weak	36.8	-	-	-	-	22.0	-	-	-	**5.6**	22.4
DGAM	Weak	**41.0**	**37.5**	**33.5**	**30.1**	**26.9**	**23.5**	**19.8**	**15.5**	**10.8**	5.3	**24.4**

Thanks