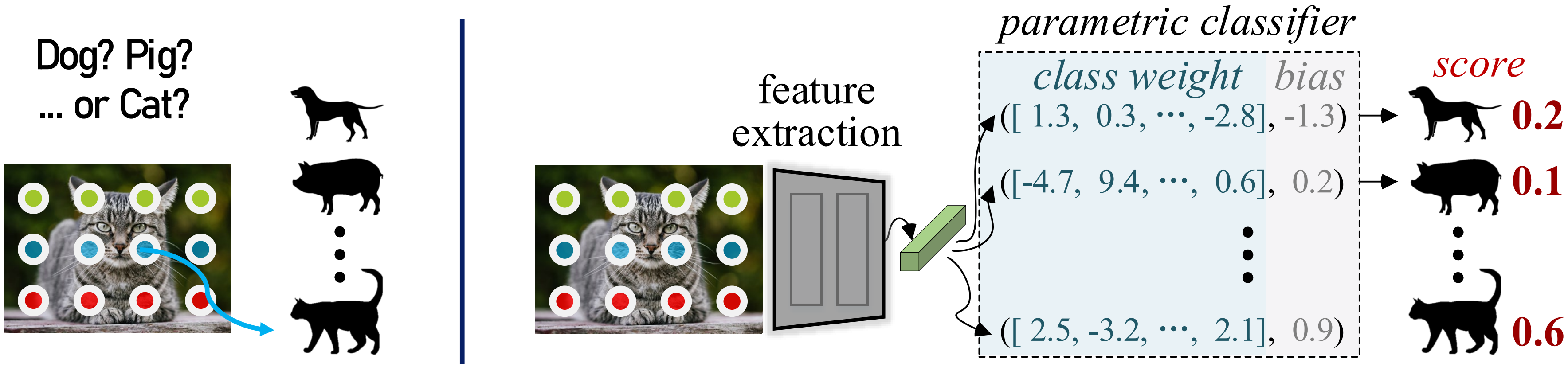
GMMSeg: Gaussian Mixture based Generative Semantic Segmentation Models

Chen Liang, Wenguan Wang, Jiaxu Miao, Yi Yang
Zhejiang University & University of Technology Sydney

NeurIPS 2022

Semantic Segmentation



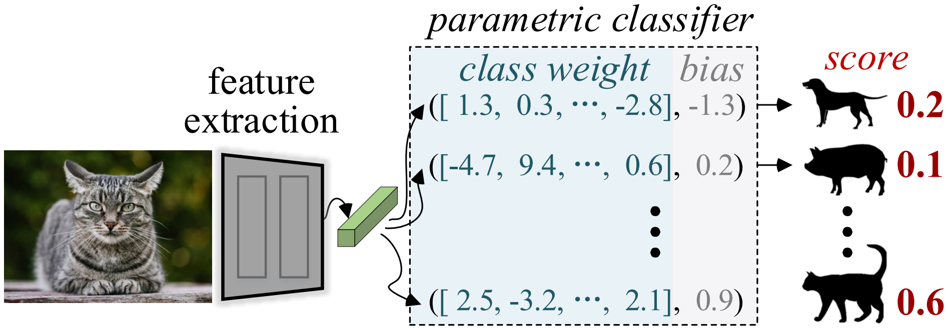
Semantic Segmentation:

Dense Feature extractor + **Parametric softmax classifier**

Deficiency of Parametric Softmax Classifier

Parametric softmax classifier:

$$p(c|\boldsymbol{x};\boldsymbol{\omega},\boldsymbol{\theta}) = \frac{\exp(y_c)}{\sum_{c'}\exp(y_{c'})} = \frac{\exp\boxed{(\boldsymbol{w}_c^\top \boldsymbol{x} + b_c)}}{\sum_{c'}\exp(\boldsymbol{w}_{c'}^\top \boldsymbol{x} + b_{c'})}$$

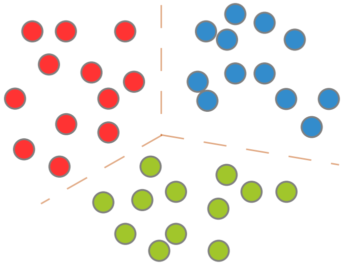


- Only learning **decision boundaries**; Ignoring underlying **data distribution**.

Straightforward: Only learn **decision boundaries**

Fail to capture the intrinsic class characteristics;

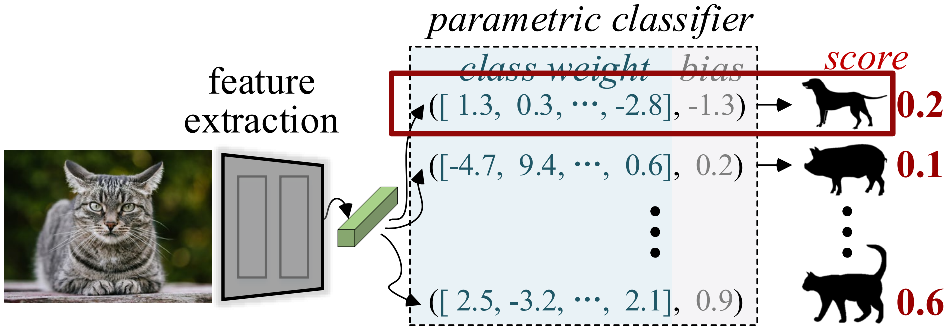
Ignore underlying **data structure**



Deficiency of Parametric Softmax Classifier

Parametric softmax classifier:

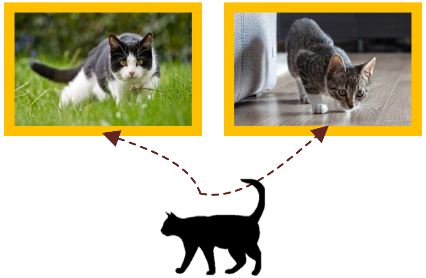
$$p(c|\boldsymbol{x};\boldsymbol{\omega},\boldsymbol{\theta}) = \frac{\exp(y_c)}{\sum_{c'}\exp(y_{c'})} = \frac{\exp(\boldsymbol{w}_c^\top\boldsymbol{x}+b_c)}{\sum_{c'}\exp(\boldsymbol{w}_{c'}^\top\boldsymbol{x}+b_{c'})}$$



- Implicit **unimodality** assumption; Bearing **no within-class variation**.

  The unimodality assumption is rarely the case in real-world scenarios.

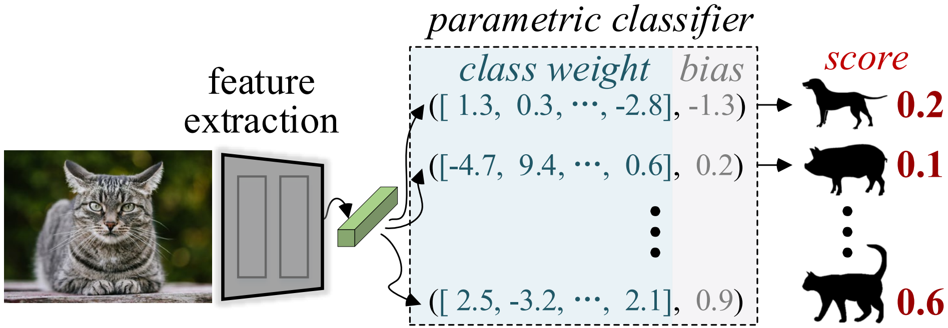
  The model **less tolerant of intra-class variances**.



Deficiency of Parametric Softmax Classifier

Parametric softmax classifier:

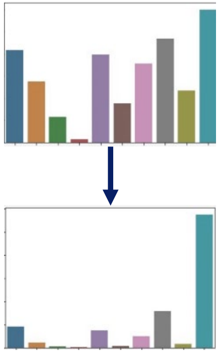
$$p(c|\boldsymbol{x};\boldsymbol{\omega},\boldsymbol{\theta}) = \boxed{\frac{\exp(y_c)}{\sum_{c'}\exp(y_{c'})}} = \frac{\exp(\boldsymbol{w}_c^\top \boldsymbol{x} + b_c)}{\sum_{c'}\exp(\boldsymbol{w}_{c'}^\top \boldsymbol{x} + b_{c'})}$$



- Inferior **robustness to out-of-distribution inputs**; Poorly **calibrated**.

Prediction score is useless besides its comparative value against others; **Struggling to recognize out-of-distribution data**.

Model accuracy deteriorates rapidly away from the decision boundaries; **yields poorly calibrated predictions.**



Rethinking *de facto* Paradigm

- Only learning **decision boundaries**; Ignoring underlying **data structure**.
- Implicit **unimodality** assumption; Bearing **no within-class variation**.
- Inferior **robustness to out-of-distribution inputs**; Poorly **calibrated**.

Is there any way to address the limitations of *de facto* segmentation regime?

↓

Our answer:

Generative Gaussian Mixture Classifier (GMMSeg)

Discriminative *vs.* Generative

Discriminative Classifier

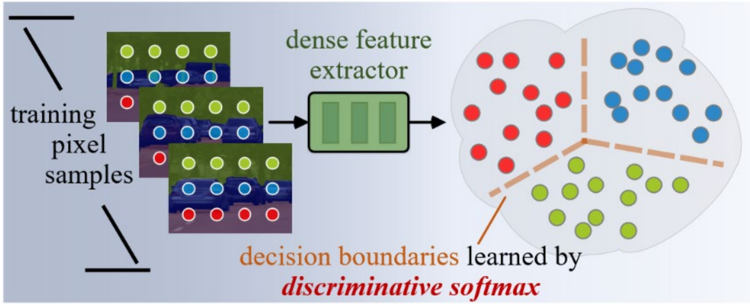
- Model $p(c|\boldsymbol{x})$ directly.
- Optimization: $\Pi_{(x,c)\in\mathcal{D}}p(c|\boldsymbol{x})$
- Model **Decision Boundaries** only
- Example: **Parametric softmax**

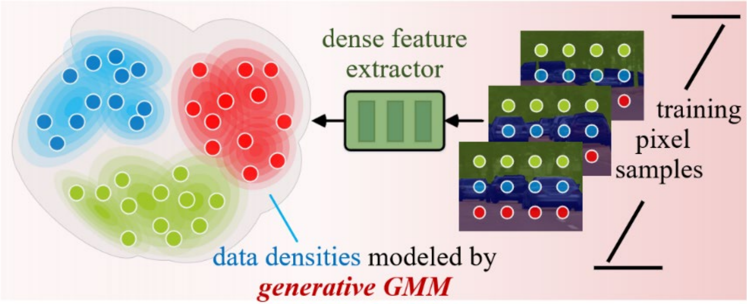
Generative Classifier

- Model joint distribution $p(\boldsymbol{x}, c)$.

  Then deduce $p(c|\boldsymbol{x}) = \frac{p(c)p(\boldsymbol{x}|c)}{\sum_{c'} \underline{p(c')} p(\boldsymbol{x}|c')}$.

  Uniform Prior
- Optimization: $\Pi_{(x,c)\in\mathcal{D}}p(\boldsymbol{x}|c)$
- Model entire **Data Distribution**
- Example: **GMMSeg**





GMMSeg: Distribution Modeling

Use **Gaussian Mixtures** to model arbitrary feature distribution:

- Simple, elegant and powerful.

$$p(\boldsymbol{x}|c;\boldsymbol{\phi}_c) = \sum\nolimits_{m=1}^{M} p(m|c;\boldsymbol{\pi}_c)p(\boldsymbol{x}|c,m;\boldsymbol{\mu}_c,\boldsymbol{\Sigma}_c) = \sum\nolimits_{m=1}^{M} \pi_{cm}\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}_{cm},\boldsymbol{\Sigma}_{cm}).$$

Parameter optimization: Maximizing the log-likelihood over all feature-label pairs $\{(\boldsymbol{x}_n, c_n)\}_{n=1}^{N}$

$$\boldsymbol{\phi}_c^* = \underset{\boldsymbol{\phi}_c}{\arg\max} \sum\nolimits_{\boldsymbol{x}_n:c_n=c} \log p(\boldsymbol{x}_n|c;\boldsymbol{\phi}_c) = \underset{\boldsymbol{\phi}_c}{\arg\max} \sum\nolimits_{\boldsymbol{x}_n:c_n=c} \log \sum\nolimits_{m=1}^{M} p(\boldsymbol{x}_n,m|c;\boldsymbol{\phi}_c)$$

$m|c \sim \mathrm{Multinomial}(\boldsymbol{\pi}_c)$: prior of mixture components, $\sum_m \pi_{cm} = 1$.

$\boldsymbol{\mu}_{cm} \in \mathbb{R}^D$, $\boldsymbol{\Sigma}_{cm} \in \mathbb{R}^{D\times D}$: mean and covariance matrix for component *m* in class *c*.

$\boldsymbol{\phi}_c = \{\boldsymbol{\pi}_c, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\}$: set of all parameters.

GMMSeg: Sinkhorn EM Optimization

Optimization through vanilla EM (F-function form[1]):

$$\textbf{E-Step:}\quad q_c^{(t)} = \arg\max_{q_c} F(q_c, \boldsymbol{\phi}_c^{(t-1)}), \qquad \textbf{M-Step:}\quad \boldsymbol{\phi}_c^{(t)} = \arg\max_{\boldsymbol{\phi}_c} F(q_c^{(t)}, \boldsymbol{\phi}_c).$$

F-function is defined as: $F(q_c, \boldsymbol{\phi}_c) = \mathbb{E}_{q_c}[\log p(\boldsymbol{x}, m|c; \boldsymbol{\phi}_c)] + H(q_c)$

EM starts with some **initial guess** at parameters $\boldsymbol{\phi}_c^{(0)}$, then iterates over

- **E-Step**: given $\boldsymbol{\phi}^{(t-1)}$, compute the posterior $q_c^{(t)}$ over the *M* components.
- **M-Step**: given soft cluster assignment $q_c^{(t)}$, the parameters are updated as $\boldsymbol{\phi}_c^{(t)}$ such that the F function is maximized.

$q_c[m] = p(m|\boldsymbol{x}, c; \boldsymbol{\phi}_c)$: the probability that data *x* is **assigned** to component *m*.

$H(q_c) = -\mathbb{E}_{q_c}[\log q_c[m]]$: the entropy of q_c.

[1] A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, 1998.

GMMSeg: Sinkhorn EM Optimization

Problem: Standard EM suffers from slow convergence; Delivers unsatisfactory results, potentially due to the parameter sensitivity of EM.

GMMSeg: Sinkhorn EM Optimization

Sinkhorn EM[1] with a uniform prior on mixture weight, i.e. $\forall c, m: \pi_{cm} = \frac{1}{M}$:

$$\textbf{E-Step:} \quad q_c^{(t)} = \arg\max_{q_c \in \mathcal{Q}_c} F(q_c, \boldsymbol{\phi}_c^{(t-1)})$$

restricted by a constraint: $\mathcal{Q}_c = \{q_c : \frac{1}{N_c} \sum_{\boldsymbol{x}_n : c_n = c} p(m|\boldsymbol{x}_n, c) = \frac{1}{M}\}$

Analogous to entropy-regularized OT: ← Solved by **Sinkhorn-Knopp** algorithm.

$$\min_{\boldsymbol{Q}_c \in \mathcal{Q}'_c} \sum_{n,m} \boldsymbol{Q}_c(n,m) \boldsymbol{O}_c(n,m) + \epsilon H(\boldsymbol{Q}_c), \quad \mathcal{Q}'_c = \{\boldsymbol{Q}_c \in \mathbb{R}_+^{N_c \times M} : \boldsymbol{Q}_c \mathbf{1}^M = \mathbf{1}^{N_c}, (\boldsymbol{Q}_c)^\top \mathbf{1}^{N_c} = \frac{N_c}{M} \mathbf{1}^M\}$$

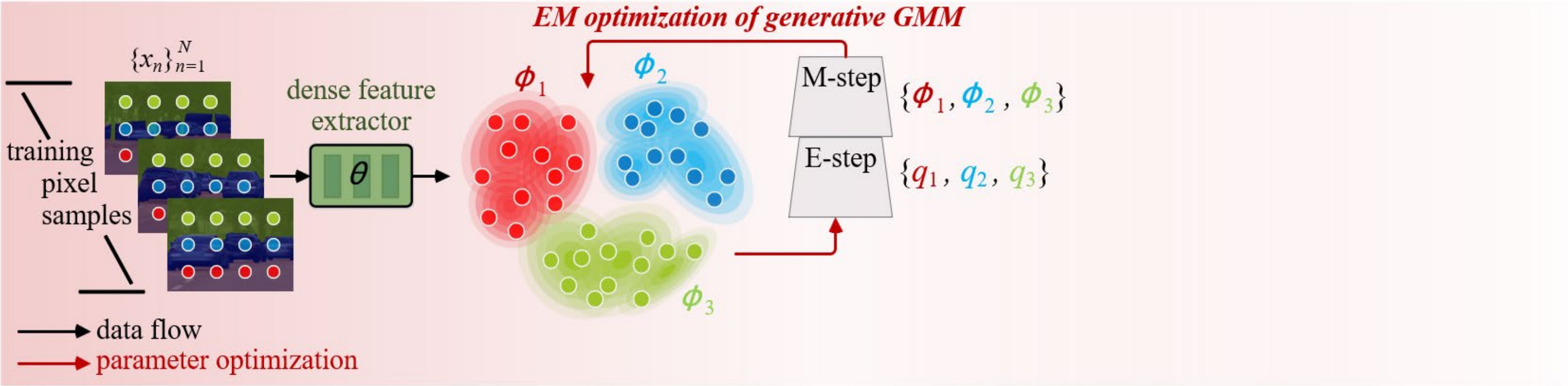
- Sinkhorn EM is proved to have the same global optimum with the standard EM yet is less prone to getting stuck in local optima[1].

$\boldsymbol{Q}_c(n,m) = q_{cn}[m]$: posterior distribution over the *M* components (target solution).

$\boldsymbol{O}_c(n,m) = -\log p(\boldsymbol{x}_n|c,m)$: negative log-likelihood (cost matrix).

[1] Sinkhorn em: an expectation-maximization algorithm based on entropic optimal transport. *arXiv*, 2021.

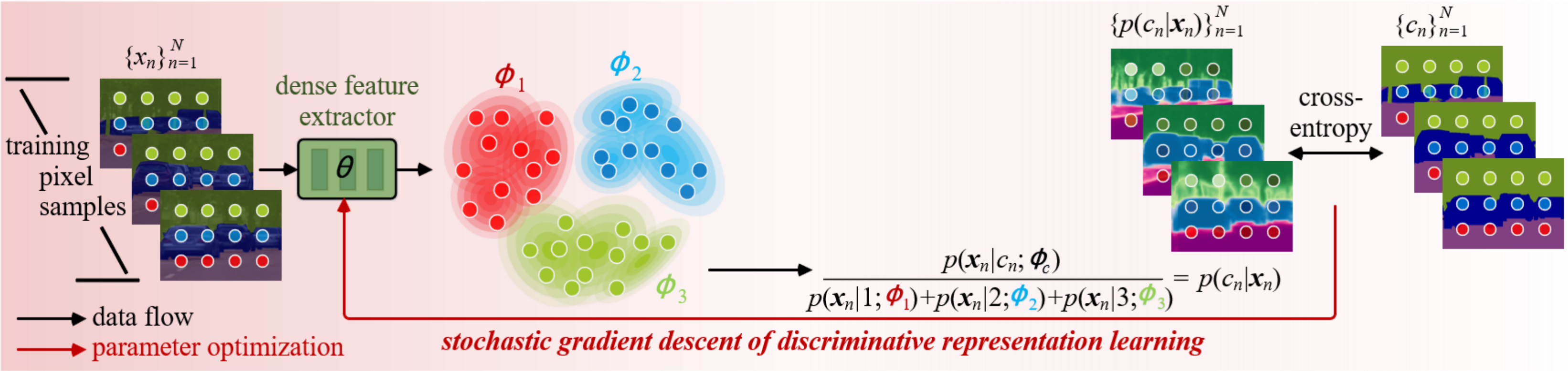
GMMSeg: Hybrid Optimization



Generative Optimization (Sinkhorn EM) of **GMM Classifier**: $\{\phi_c^*\}_{c=1}^{C} =$

$$\{\arg\max_{\phi_c} \sum_{\boldsymbol{x}_n : c_n = c} \log p(\boldsymbol{x}_n | c; \boldsymbol{\phi}_c)\}_{c=1}^{C} = \{\arg\max_{\phi_c} \sum_{\boldsymbol{x}_n : c_n = c} \log \sum_{m=1}^{M} \pi_{cm} \mathcal{N}(\boldsymbol{x}_n; \boldsymbol{\mu}_{cm}, \boldsymbol{\Sigma}_{cm})\}_{c=1}^{C},$$

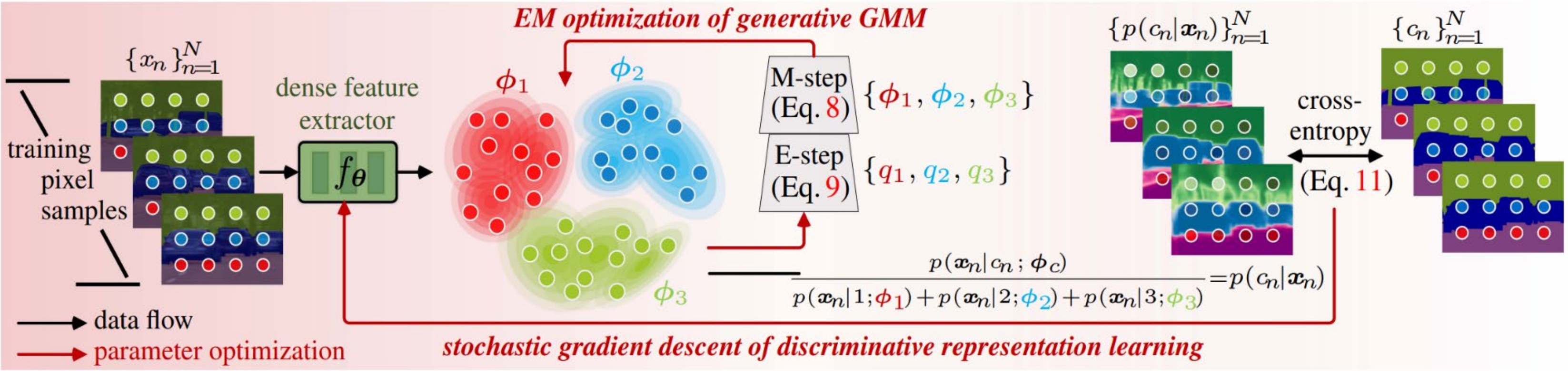
GMMSeg: Hybrid Optimization



Discriminative Learning (Cross-Entropy Loss) of **Dense Representation**: $\boldsymbol{\theta}^* =$

$$\arg\min_{\boldsymbol{\theta}} -\sum_{(x,c)\in\mathcal{D}} \log p(c|\boldsymbol{x};\{\boldsymbol{\phi}_c^*\}_{c=1}^C,\boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} -\sum_{(x,c)\in\mathcal{D}} \log\left(\frac{\sum_{m=1}^M \pi_{cm}\mathcal{N}(f_{\boldsymbol{\theta}}(x);\boldsymbol{\mu}_{cm},\boldsymbol{\Sigma}_{cm})}{\sum_{c'=1}^C\sum_{m=1}^M \pi_{c'm}\mathcal{N}(f_{\boldsymbol{\theta}}(x);\boldsymbol{\mu}_{c'm},\boldsymbol{\Sigma}_{c'm})}\right)$$

GMMSeg: Hybrid Optimization



Generative Optimization (Sinkhorn EM) of **GMM Classifier**: $\{\phi_c^*\}_{c=1}^C =$

$$\{\arg\max_{\phi_c} \sum_{\boldsymbol{x}_n: c_n=c} \log p(\boldsymbol{x}_n|c;\boldsymbol{\phi}_c)\}_{c=1}^C = \{\arg\max_{\phi_c} \sum_{\boldsymbol{x}_n: c_n=c} \log \sum_{m=1}^{M} \pi_{cm}\mathcal{N}(\boldsymbol{x}_n;\boldsymbol{\mu}_{cm},\boldsymbol{\Sigma}_{cm})\}_{c=1}^C,$$

Discriminative Learning (Cross-Entropy Loss) of **Dense Representation**: $\boldsymbol{\theta}^* =$

$$\arg\min_{\boldsymbol{\theta}} -\sum_{(x,c)\in\mathcal{D}} \log p(c|\boldsymbol{x};\{\phi_c^*\}_{c=1}^C,\boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} -\sum_{(x,c)\in\mathcal{D}} \log\left(\frac{\sum_{m=1}^{M}\pi_{cm}\mathcal{N}(f_{\boldsymbol{\theta}}(x);\boldsymbol{\mu}_{cm},\boldsymbol{\Sigma}_{cm})}{\sum_{c'=1}^{C}\sum_{m=1}^{M}\pi_{c'm}\mathcal{N}(f_{\boldsymbol{\theta}}(x);\boldsymbol{\mu}_{c'm},\boldsymbol{\Sigma}_{c'm})}\right)$$

Summary: Generative Gaussian Mixture Based Classifier

- **Versatility.**

GMMSeg is a principled framework, fully compatible with modern seg. architectures.

- Best of Both Worlds: **Strong discriminative performance.**

GMMSeg achieves the merits of both generative and discriminative learning.

Fit data distribution on evolving feature space with online EM generative optimization.

Discriminatively end-to-end trained data space under the guidance of the GMM classifier.

- Best of Both Worlds: **Distribution-preserving in nature.**

GMMSeg can naturally reject abnormal inputs, without any post-processing;

Meaningful likelihood of the example fitting each class GMM distribution; Well-calibrated.

Experiments: Semantic Segmentation

Quantitative results on ADE20K val, Cityscapes val, and COCO-Stuff test with mean IoU as the evaluation metric.

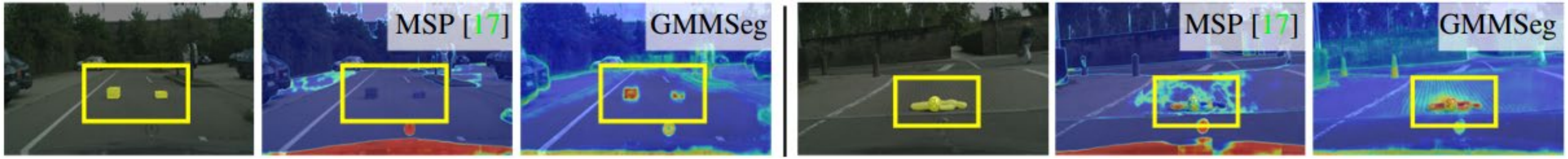
Method	Backbone	ADE_{20K}	Citys.	COCO.
FCN [CVPR15] [1]	$ResNet_{101}$	39.9	75.5	32.6
PSPNet [CVPR17] [3]	$ResNet_{101}$	44.4	79.8	37.8
SETR [CVPR21] [9]	$^{\dagger}ViT_{Large}$	48.2	79.2	-
Segmenter [ICCV21] [8]	$^{\dagger}ViT_{Large}$	‡51.8	79.1	-
MaskFormer [NeurIPS21] [81]	$^{\dagger}Swin_{Base}$	‡52.7	-	-
$DeepLab_{V3+}$ [ECCV18] [47]	$ResNet_{101}$	45.5	80.6	33.8
GMMSeg		**46.7** $\uparrow$1.2	**81.1** $\uparrow$0.5	**35.5** $\uparrow$1.7
OCRNet [ECCV20] [48]	$HRNet_{V2W48}$	43.3	80.4	37.6
GMMSeg		**44.8** $\uparrow$1.5	**81.2** $\uparrow$0.8	**39.2** $\uparrow$1.6
UPerNet [ECCV18] [49]	$Swin_{Base}$	48.0	81.1	43.4
GMMSeg		**49.0** $\uparrow$1.0	**81.8** $\uparrow$0.7	**44.3** $\uparrow$0.9
SegFormer [NeurIPS21] [7]	MiT_{B5}	50.0	82.0	44.0
GMMSeg		**50.8** $\uparrow$0.8	**82.6** $\uparrow$0.6	**44.7** $\uparrow$0.7

†: pretrained on $ImageNet_{22K}$; ‡: using larger crop-size, *i.e.*, 640×640

Experiments: Anomaly Segmentation

Fishyscapes Lost&Found test and Static test.

Method	Re-training	Extra Network	OoD Data	FS Lost&Found AP ↑	FS Lost&Found FPR_{95} ↓	FS Static AP ↑	FS Static FPR_{95} ↓
Density - Single-layer NLL [12]	✗	✓	✗	3.01	32.9	40.86	21.29
Density - Minimum NLL [12]	✗	✓	✗	4.25	47.15	62.14	17.43
Density - Logistic Regression [12]	✗	✓	✓	4.65	24.36	57.16	13.39
Image Resynthesis [15]	✗	✓	✗	5.70	48.05	29.6	27.13
Bayesian Deeplab [16]	✓	✗	✗	9.81	38.46	48.70	15.05
OoD Training - Void Class [17]	✓	✗	✓	10.29	22.11	45.00	19.40
Discriminative Outlier Detection Head [18]	✓	✓	✓	31.31	19.02	96.76	0.29
Dirichlet Deeplab [19]	✓	✗	✓	34.28	47.43	31.30	84.60
SynBoost [20]	✗	✓	✓	43.22	15.79	72.59	18.75
MSP [21]	✗	✗	✗	1.77	44.85	12.88	39.83
Entropy [22]	✗	✗	✗	2.93	44.83	15.41	39.75
kNN Embedding - density [12]	✗	✗	✗	3.55	30.02	44.03	20.25
SML [14]	✗	✗	✗	31.05	21.52	53.11	19.64
GMMSeg-$\text{DeepLab}_{\text{V3+}}$	✗	✗	✗	**55.63**	**6.61**	**76.02**	**15.96**



Summary: Rethinking *de facto* Paradigm in Segmentation

- Only learning **decision boundaries**; Ignoring underlying **data structure**.
- Implicit **unimodality** assumption; Bearing **no within-class variation**.
- Inferior **robustness to out-of-distribution inputs**; Poorly **calibrated**.

Our answer:

Generative Classifier: **Gaussian Mixture Model based Classifier (GMMSeg)**.

Paper

Code

Thanks!