ALIFE: Adaptive Logit RegularIzer and Feature REplay for Incremental Semantic Segmentation

Youngmin Oh　　Donghyeon Baek　　Bumsub Ham

Summary

Problem statement: incremental semantic segmentation (ISS)

- ISS aims at continually segmenting novel categories without accessing training samples for previously learned categories.
- Regularization-based methods focus on designing regularization terms. Among them, MiB introduces calibrated cross-entropy (CCE) and calibrated knowledge distillation (CKD) terms. While both are widely adopted in ISS, there is a lack of theoretical understanding of them.
- Replay-based methods exploit a small set of previously seen images together with ground-truth labels. They achieve state-of-the-art performance at the cost of large memory footprint.
- *Goal*: Achieve a better trade-off in terms of accuracy and efficiency.

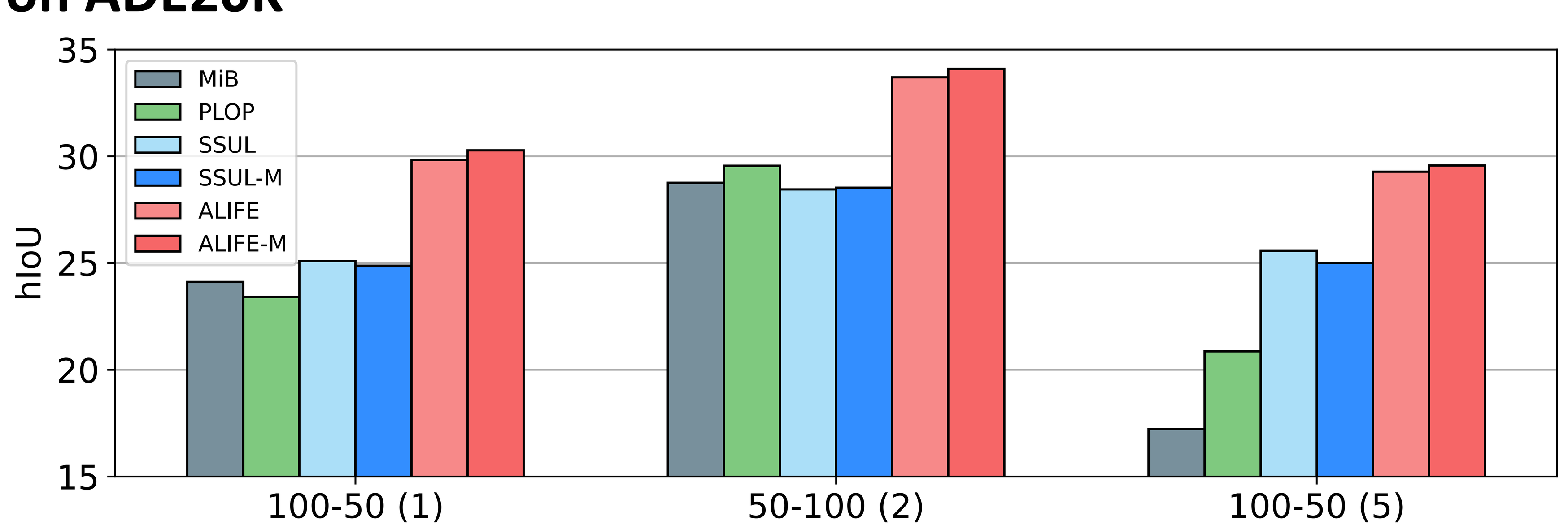
Contributions

- Provide an in-depth analysis of CCE and CKD terms.
- Present a new regularization term, called adaptive logit regularizer, that incorporates the merits of CCE and CKD, while discarding the negative effects.
- Propose to memorize latent features for replaying, reducing memory requirements and avoiding data privacy issues.

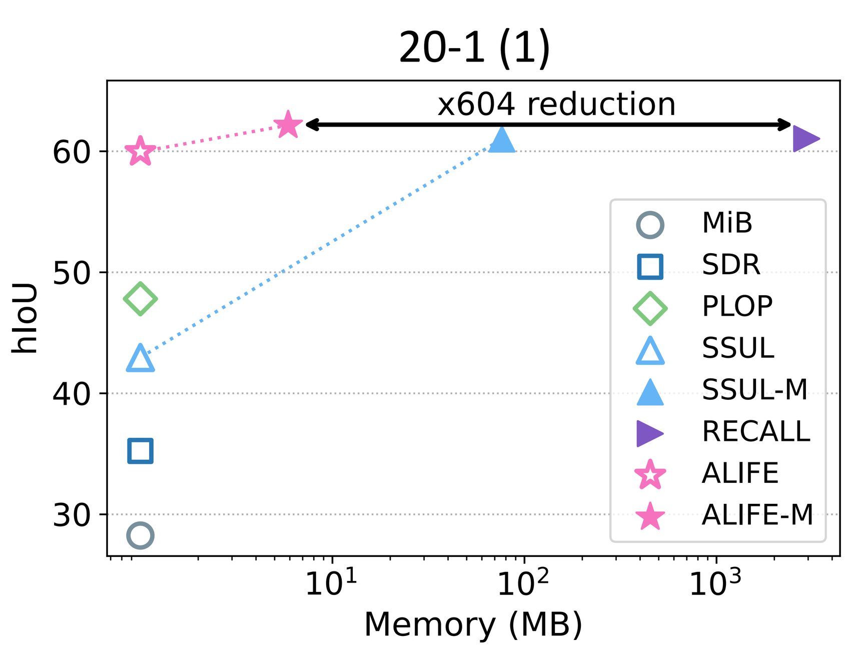
Results

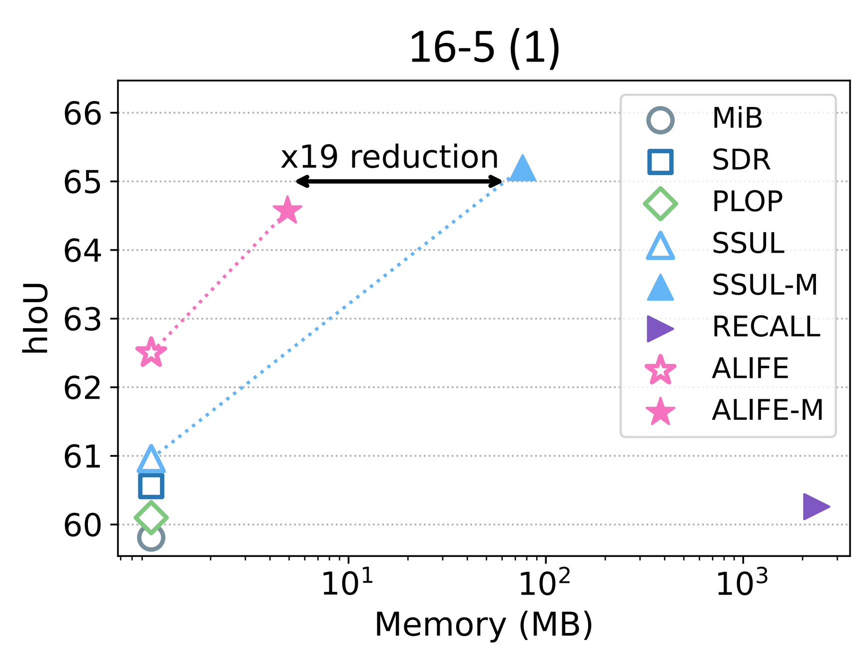
A-B (C) indicates the # of categories at a base stage, the total # of novel categories, and the # of incremental stages, respectively

on ADE20K



on PASCAL VOC 2012





Analysis of CCE and CKD

$$p_c^t(\mathbf{p}) = \frac{e^{z_c^t(\mathbf{p})}}{\sum_{k\in C_{\text{all}}^t} e^{z_k^t(\mathbf{p})}}, \quad c \in C_{\text{all}}^t \qquad q_c^t(\mathbf{p}) = \frac{e^{z_c^t(\mathbf{p})}}{\sum_{k\in C_{\text{prev}}^t} e^{z_k^t(\mathbf{p})}}, \quad c \in C_{\text{prev}}^t$$

$$C_{\text{all}}^t = C_{\text{prev}}^t \cup C_{\text{new}}^t \qquad \varnothing = C_{\text{prev}}^t \cap C_{\text{new}}^t \qquad C_{\text{prev}}^t = C_{\text{all}}^{t-1}$$

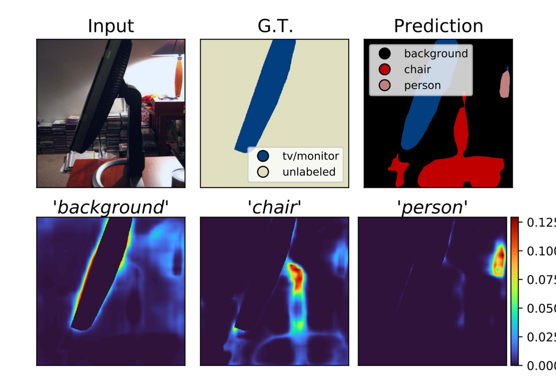
Proposition 1. *For $c \in C_{\text{prev}}^t$, q_c^t is always larger than p_c^t.*

Gradients w.r.t a logit value $z_c^t(\mathbf{p})$ for a category c at location $\mathbf{p}$

- Calibrated Cross-Entropy (CCE)

$$L_{\text{CCE}}(\mathbf{p}) = \begin{cases} -\log p_{c^*}^t(\mathbf{p}), & \mathbf{p} \in \mathcal{R}_{\text{new}}^t \\ -\log p_{\text{cce}}^t(\mathbf{p}), & \mathbf{p} \notin \mathcal{R}_{\text{new}}^t \end{cases} \qquad c^* = y(\mathbf{p}) \qquad p_{\text{cce}}^t(\mathbf{p}) = \textstyle\sum_{k\in C_{\text{prev}}^t} p_k^t(\mathbf{p})$$

	Conditions		Gradients	
Labeled regions	$\mathbf{p} \in \mathcal{R}_{\text{new}}^t$	$c = y(\mathbf{p})$	$p_c^t - 1$	Same as the gradients of vanilla CE
		$c \neq y(\mathbf{p})$	p_c^t	
Unlabeled regions	$\mathbf{p} \notin \mathcal{R}_{\text{new}}^t$	$c \in C_{\text{new}}^t$	p_c^t	○
		$c \in C_{\text{prev}}^t$	$p_c^t - q_c^t$	✕



Visualization of $q_c^t - p_c^t$

It reduces logit values of new categories by gradient descent.
This is important to prevent overfitting to the new categories.

It always raises logit values of all previous categories by gradient descent, regardless of whether predictions of a current model are correct or not.

- Calibrated Knowledge Distillation (CKD)

$$L_{\text{CKD}}(\mathbf{p}) = -p_{bg}^{t-1}(\mathbf{p}) \log p_{\text{ckd}}^t(\mathbf{p}) + \sum_{k\in C_{\text{prev}}^t \setminus \{bg\}} -p_k^{t-1}(\mathbf{p}) \log p_k^t(\mathbf{p}), \quad \forall \mathbf{p} \qquad p_{\text{ckd}}^t(\mathbf{p}) = \textstyle\sum_{k\in\{bg\}\cup C_{\text{new}}^t} p_k^t(\mathbf{p})$$

Conditions		Gradients	
$\forall \mathbf{p}$	$c \in C_{\text{prev}}^t \setminus \{bg\}$	$p_c^t - p_c^{t-1}$	○
	$c \in \{bg\} \cup C_{\text{new}}^t$	$(p_{\text{ckd}}^t - p_{bg}^{t-1})\frac{p_c^t}{p_{\text{ckd}}^t}$	✕

$$L_{\text{KD}}(\mathbf{p}) = \sum_{k\in C_{\text{prev}}^t} -p_k^{t-1}(\mathbf{p}) \log q_k^t(\mathbf{p}), \quad \forall \mathbf{p}$$

Conditions		Gradients
$\forall \mathbf{p}$	$c \in C_{\text{prev}}^t$	$q_c^t - p_c^{t-1}$

It makes p_c^t similar to p_c^{t-1} directly, while vanilla KD makes q_c^t similar to p_c^{t-1}.

It hinders discriminating new categories from a background category at training time.

Step 1:Train a current model

- Based on the analysis, we define a new form of gradients and introduce an adaptive logit regularizer (ALI).

Conditions		Gradients
$\mathbf{p} \notin \mathcal{R}_{\text{new}}^t$	$c \in C_{\text{new}}^t$	p_c^t
	$c \in C_{\text{prev}}^t$	$p_c^t - p_c^{t-1}$

○ Same as the 3rd row in the table of CCE.
○ Similar to the 1st row in the table of CKD, except that ours computes the gradients for all previous categories including the background category.

$$L_{\text{ALI}}(\mathbf{p}) = \log\left(\sum_{k\in C_{\text{all}}^t} e^{z_k^t(\mathbf{p})}\right) - \sum_{k\in C_{\text{prev}}^t} p_k^{t-1}(\mathbf{p}) z_k^t(\mathbf{p})$$

- We train a current model with a new training objective.

$$L_{\text{S1}}(\mathbf{p}) = L_{\text{CE}}(\mathbf{p}) + \lambda_{\text{ALI}} L_{\text{ALI}}(\mathbf{p}) + \lambda_{\text{KD}} L_{\text{KD}}(\mathbf{p}) \mathbb{1}[\mathbf{p} \in \mathcal{R}_{\text{new}}^t]$$

Step 2-1: Extract features

- Extract features of new categories in order to replay them in subsequent stages.

Freeze $\{\phi^t, w^t\}$
for $c \in C_{\text{new}}^t$ **do**
　$s \leftarrow 0$
　repeat
　　$(x, y) \sim D^t$
　　Extract a feature map $f^t \leftarrow \phi^t(x)$
　　Average features for the category c $m_c^t(s) \leftarrow \frac{1}{|\mathcal{R}_c|}\sum_{\mathbf{p}\in\mathcal{R}_c} f^t(\mathbf{p})$
　　$s \leftarrow s + 1$
　until $s = S$ // S indicates the number of features for the category c
end for

ϕ^t : a feature extractor at a stage t
w^t : a classifier at a stage t
$m_c^t(s)$: the s-th extracted feature for the category c at a stage t

Step 2-2: Compensate a distribution shift of memorized features

Train rotation matrices

- Memorized features, which are extracted in the previous stage $t-1$, are not compatible with a current classifier w^t.
- To handle this, we propose to train category-specific rotation matrices.

A rotation transform is light-weight, and enables maintaining the relations between features that belong to the same category.

1. Each rotation matrix is defined using the Cayley transform

$\mathbf{S}_c = \mathbf{U}_c - \mathbf{U}_c^\top$　$\mathbf{U}_c$: a strictly upper triangular matrix (randomly initialized)

$\mathbf{R}_c = (\mathbf{I} - \mathbf{S}_c)(\mathbf{I} + \mathbf{S}_c)^{-1}$　$\mathbf{I} = \mathbf{R}_c\mathbf{R}_c^\top = \mathbf{R}_c^\top\mathbf{R}_c, \quad c \in C_{\text{prev}}^t$

2. Compute correlation scores and define prototypes for previous and current stages
(Note that f^{t-1} and m_c^{t-1} share the same feature space)

$$v_c(\mathbf{p}) = \sum_{s=1}^{S} \text{ReLU}\left(\frac{f^{t-1}(\mathbf{p})}{\|f^{t-1}(\mathbf{p})\|} \cdot \frac{m_c^{t-1}(s)}{\|m_c^{t-1}(s)\|}\right)$$

$$\sigma_c(\mathbf{p}) = \frac{e^{\tau v_c(\mathbf{p})}}{\sum_{\mathbf{p}} e^{\tau v_c(\mathbf{p})}}$$ τ: a temperature parameter

$$r_c^{t-1} = \sum_{\mathbf{p}} \sigma_c(\mathbf{p}) f^{t-1}(\mathbf{p}), \quad r_c^t = \sum_{\mathbf{p}} \sigma_c(\mathbf{p}) f^t(\mathbf{p})$$

3. Each matrix rotates a previous prototype r_c^{t-1} to align with a current prototype r_c^t and is trained with the following objective

$$\hat{r}_c^t = \mathbf{R}_c r_c^{t-1}$$

$$L_{\text{FID}} = \sum_{c\in C_{\text{prev}}^t}\left(1 - \frac{\hat{r}_c^t}{\|\hat{r}_c^t\|} \cdot \frac{r_c^t}{\|r_c^t\|}\right) \qquad L_{\text{REG}} = \sum_{c\in C_{\text{prev}}^t} -\log\left(\frac{e^{\hat{r}_c^t \cdot w_c^t}}{\sum_{i\in C_{\text{all}}^t} e^{\hat{r}_c^t \cdot w_i^t}}\right)$$

$$L_{\text{S2}} = \lambda_{\text{ROT}} L_{\text{FID}} + (1 - \lambda_{\text{ROT}}) L_{\text{REG}}$$

Update features

- $\hat{m}_c^t(s) = \mathbf{R}_c m_c^{t-1}(s)$

Step 3: Fine-tune a classifier

- The updated features along with training samples of D^t are used to fine-tune a classifier w^t with the following objective.

$$L_{\text{S3}}(\mathbf{p}) = L_{\text{FL}}(\mathbf{p}) + \lambda_{\text{ALI}} L_{\text{ALI}}(\mathbf{p}) + \lambda_{\text{MEM}} L_{\text{MEM}}$$

$$L_{\text{FL}}(\mathbf{p}) = -(1 - p_{\hat{c}}^t(\mathbf{p}))^\alpha \log p_{\hat{c}}^t(\mathbf{p}), \ \hat{c} = \begin{cases} y(\mathbf{p}), & \mathbf{p} \in \mathcal{R}_{\text{new}}^t \\ \text{argmax}_{k\in C_{\text{prev}}^t} p_k^{t-1}(\mathbf{p}), & \mathbf{p} \notin \mathcal{R}_{\text{new}}^t \end{cases}$$

$$L_{\text{MEM}} = \sum_{c\in C_{\text{prev}}^t} \sum_{s=1}^{S} -\log\left(\frac{e^{\hat{m}_c^t(s)\cdot w_c^t}}{\sum_{k\in C_{\text{all}}^t} e^{\hat{m}_k^t(s)\cdot w_k^t}}\right)$$