2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)

2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) | 978-1-6654-0915-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/WACV51458.2022.00214

Multi-Domain Incremental Learning for Semantic Segmentation

Prachi Garg[1] Rohit Saluja[1] Vineeth N Balasubramanian[2] Chetan Arora[3]
Anbumani Subramanian[1] C.V. Jawahar[1]
[1]CVIT - IIIT Hyderabad, India [2]IIT Hyderabad, India [3]IIT Delhi, India
[1]prachigarg2398@gmail.com, [1]rohit.saluja@research.iiit.ac.in, [2]vineethnb@iith.ac.in,
[3]chetan@cse.iitd.ac.in, [1]{anbumani, jawahar}@iiit.ac.in

Abstract

Recent efforts in multi-domain learning for semantic segmentation attempt to learn multiple geographical datasets in a universal, joint model. A simple fine-tuning experiment performed sequentially on three popular road scene segmentation datasets demonstrates that existing segmentation frameworks fail at incrementally learning on a series of visually disparate geographical domains. When learning a new domain, the model catastrophically forgets previously learned knowledge. In this work, we pose the problem of multi-domain incremental learning for semantic segmentation. Given a model trained on a particular geographical domain, the goal is to (i) incrementally learn a new geographical domain, (ii) while retaining performance on the old domain, (iii) given that the previous domain's dataset is not accessible. We propose a dynamic architecture that assigns universally shared, domain-invariant parameters to capture homogeneous semantic features present in all domains, while dedicated domain-specific parameters learn the statistics of each domain. Our novel optimization strategy helps achieve a good balance between retention of old knowledge (stability) and acquiring new knowledge (plasticity). We demonstrate the effectiveness of our proposed solution on domain incremental settings pertaining to real-world driving scenes from roads of Germany (Cityscapes), the United States (BDD100k), and India (IDD). [1]

1. Introduction

Driving is a skill that humans do not forget under natural circumstances. They can easily drive in multiple geographies. This shows that humans are naturally capable of lifelong learning and barely forget previously learned visual patterns when faced with a domain shift or given new objects to identify. In recent times, there has been an active interest in developing universal vision systems capable of performing well in multiple visual domains. We ask the question: can a semantic segmentation model trained on road scenes of a particular city extend to learn novel geographic environments?

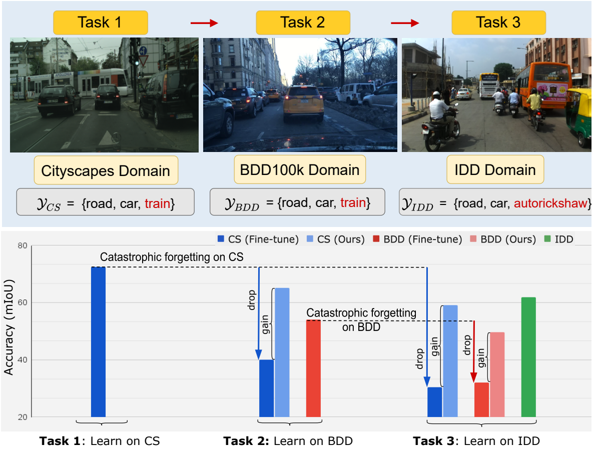


Figure 1: *Top row:* Our setting over three incremental tasks: learning a model on CS (task 1), followed by learning on BDD (task 2) and IDD (task 3). The domains have non-overlapping label spaces, where the black categories are shared among all domains and red categories are domain-specific. *Bottom row:* The problem of **catastrophic forgetting**: as the CS model is fine-tuned on BDD and further on IDD, we witness a sharp degradation in performance of old datasets; our method significantly mitigates this forgetting.

Consider learning incrementally over three autonomous driving datasets: Cityscapes → BDD100k → Indian Driving Dataset. We conduct a fine-tuning experiment over these three datasets and find that deep neural networks forget previously acquired knowledge when trained on novel geographic domains. The degradation of performance is evident in Figure 1. This phenomenon where new knowledge overwrites previous knowledge is referred to as catastrophic forgetting [29] in incremental learning. We observe that when shifting from one geography to another, catastrophic forgetting can be due to two factors: (i) a domain

[1] Code is available at https://github.com/prachigarg23/MDIL-SS

978-1-6654-0915-5/21/$31.00 ©2021 IEEE
DOI 10.1109/WACV51458.2022.00214

shift is encountered in the road scene environment due to varying background conditions, such as driving culture, illumination, and weather; and (ii) the label space for semantic segmentation might change when encountered with novel classes in the new geography while missing some of the old classes.

Most semantic segmentation research today focuses on developing models that are specialized to a specific dataset or environment. They fail to work in continual learning settings, where we want to extend the scope of our autonomous driving model to road environments with a potential domain shift. In the absence of an incremental model, a naive way to solve this problem is to train a separate, independent model for each geography, store all models and deploy the corresponding model when the road scene environment changes. Another option is to store data from all these domains and re-train a single, joint model from scratch each time a new domain's data is collected. Both these approaches involve a significant computational overhead, are not scalable, and data-inefficient as it requires storing large amounts of data that may be proprietary or unavailable. Moreover, when training separate models, the new domain cannot benefit from an old one's existing knowledge (forward transfer in continual learning [9]).

Kalluri *et al*. [18] proposed a universal semi-supervised semantic segmentation technique that models multiple geographic domains simultaneously in a universal model. Their method requires simultaneous access to all the datasets involved, and does not follow the incremental learning setting. We extend this literature by considering the case of incremental learning where multiple domains are learnt in a single model, *sequentially*, eliminating the need to store previously learnt data. We draw inspiration from existing literature on multi-domain incremental learning (MDIL) for classification [36, 37, 26], and reparametrize a semantic segmentation architecture into domain-invariant parameters (shared among domains) and domain-specific parameters that are exclusively added, trained on and used for each novel domain being learnt. To the best of our knowledge, our work is the first attempt at MDIL for semantic segmentation. Our key contributions can be outlined as follows:

1. We define the problem of multi-domain incremental semantic segmentation and propose a dynamic framework that reparameterizes the network into a set of domain-invariant and domain-specific parameters. We achieve this with a 78.83% parameter sharing across all domains.
2. In continual learning, plasticity is the ability to acquire new knowledge, while stability refers to retaining existing knowledge [31]. Our primary objective in this work is to tackle this stability-plasticity dilemma. We propose a novel optimization strategy designed to fine-tune the domain-invariant and domain-specific layers differently towards a good stability-plasticity trade-off. In a first, we find a combination of differential learning rates and domain adaptive knowledge distillation to be highly effective towards achieving this goal.
3. We consider the challenging issue of non-overlapping label spaces in multi-domain incremental semantic segmentation owing to its relevance in real-world autonomous driving scenarios. We show that our model performs well on both: (i) datasets that have a domain shift but an overlapping label space (Cityscapes → BDD100k); (ii) datasets that have non-overlapping label spaces in addition to domain shift (Cityscapes → Indian Driving Dataset). We also analyze forward transfer and domain interference in these cases (Section 4).

2. Related Work

2.1. Incremental Learning

Incremental learning (IL) involves lifelong learning of new concepts in an existing model without forgetting previously learned concepts. IL in computer vision has most widely been studied for image classification [35, 8], where the methods can be broadly grouped into three categories [8]: memory or replay-based, regularization-based, and parameter-isolation based methods. Replay-based techniques store previous experience either implicitly via generative replay [42, 49, 34] or explicitly [38, 4, 17, 50] in the form of raw samples or dataset statistics of previous data. Regularization-based methods can be further categorized as prior-focused [56, 20, 6, 1] and data-focused methods [16, 24, 11]. In parameter isolation methods [27, 26, 37, 2], additional task-specific parameters are added to a dynamic architecture for each new task.

Multi-Domain Incremental Learning. Multi-domain IL is concerned with sequentially learning a single task, say image classification, on multiple visual domains with possibly different label spaces. The earliest works in this space on the classification task are Progressive Neural Networks [41], Dynamically Expandable Networks (DENs) [54], and attaching controller modules to a base network [40]. [26] and [28] learn a domain-specific binary mask over a fixed backbone architecture to get a compact and memory-efficient solution. Other works using parameter-isolation based techniques dedicate a domain-specific subset of parameters to each unique task, to mitigate forgetting by construction. To this end, Rebuffi *et al*. introduced residual adapters in series [36] and parallel [37] in an attempt to define universal parametrizations for multi-domain networks by using certain domain-specific and shared network parameters. Other recent works [14, 3, 13, 43] also share a similar approach, but focus on the classification task. Recently, [25] proposed incremental learning across various domains and categories for object detection. Also related to our work is *multi-task* incremental learning [19] over tasks like edge detection and human

Problem Setting	Sequential	Differences, Source vs target		Data (availability, supervision)		Goals	Solution Type	
		Label Space	Domain Shift	Source	Target		Task-Aware	Multi-Head
UDA [46, 47, 53]	✓	same	✓	✓	✓(unlabeled)	learn new	×	×
Class-IL [32, 5, 12, 33]	✓	different	×	×	✓	retain old, learn new	×	×
MDL [18]	×	different	✓	✓	✓	retain all	✓	✓
MDIL *(ours)*	✓	different	✓	×	✓	retain old, learn new	✓	✓

Table 1: A comparison of different semantic segmentation settings: Unsupervised Domain Adaptation (UDA), Class Incremental Learning (Class-IL), Multi-Domain Learning (MDL) and Multi-Domain Incremental Learning (MDIL)

parts segmentation. Our work in multi-domain incremental learning, while inspired by these methods, seeks to address the semantic segmentation setting for the first time.

Incremental Learning for Semantic Segmentation. IL methods have been developed for semantic segmentation in recent years, although from a ***class-incremental*** perspective. [32, 5] and [21] were the first to solve class-IL for semantic segmentation. Recent methods [5, 12, 33] focus on the problem of semantic shift in the background class distribution, which is typical to *strict* class incremental learning for semantic segmentation. In their setting, labels occurring in previous steps are not used for training in subsequent steps and all classes belong to the same domain. Not only does MDIL have a domain drift between any two consecutive steps, there are no restrictions on the label spaces which may or may not have common labels (Table 1).

2.2. Domain Adaptation

Our work may also be related in a sense to domain adaptation [57, 46, 53, 30, 51] for semantic segmentation. [52] proposed incremental unsupervised domain adaptation and showed the effectiveness of learning domain shift by adapting the model incrementally over smaller, progressive domain shifts. Class-incremental domain adaptation [23] focused on source-free domain adaptation while also learning novel classes in target domain. However, all such efforts tackle domain adaptation where source knowledge is adapted to target domains in general, unlike our work in IL which focuses on retaining source domain performance while learning on the target domain (Table 1).

3. Multi-Domain Incremental Learning for Semantic Segmentation: Methodology

Problem Setting. In incremental learning, $\mathcal{T}$ tasks are presented sequentially, each corresponding to a different dataset of domains $\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_t, ..., \mathcal{D}_T$ having label spaces $\mathcal{Y}_1, \mathcal{Y}_2, ..., \mathcal{Y}_t, ..., \mathcal{Y}_T$, respectively. Learning takes place in *incremental steps*, where each step involves learning an existing model on the current task $\mathcal{T}_t$, which in our case is $\mathcal{D}_t$. A domain $\mathcal{D}_t$ represents image data collected from a particular geographic road environment, and $\mathcal{Y}_t$ represents the semantic label space of the classes present in that domain. We consider the general case of non-overlapping label spaces such that the label space $\mathcal{Y}_t$ will either have a full overlap or a partial overlap w.r.t. $\mathcal{Y}_{t-1}$. $\mathcal{Y}_t$ may contain novel classes that were not present in $\mathcal{Y}_{t-1}$ and $\mathcal{Y}_{t-1}$ may also have novel classes absent in $\mathcal{Y}_t$. $\mathcal{D}_t$ has a domain shift with respect to $\mathcal{D}_{t-1}$, as expected.

Our goal is to train a single semantic segmentation model M that learns to classify data on each domain $\mathcal{D}_t$, sequentially. Hence, given $\mathcal{T}$ tasks, at each IL step t, we aim to learn a task-aware mapping $M_t(\mathcal{X}_t, t) = \mathcal{Y}_t$ for the t^{th} domain $\mathcal{D}_t = (\mathcal{X}_t, \mathcal{Y}_t)$, such that performance on any of the previous domains $\mathcal{D}_{t-i},\ 0 < i < t$ does not degrade when learning on the t^{th} domain. At any given step t, input $\mathcal{X}_{t-i}$ or annotation $\mathcal{Y}_{t-i}$ data pertaining to any previous domain $\mathcal{D}_{t-i}$ is not available for training. Note that we use the terms *task*, *domain*, and *dataset* interchangeably to refer to $\mathcal{D}_t$.

Proposed Framework. Our framework M, as illustrated in Figure 2, is composed of a shared encoder module $\mathcal{F}$ and different domain-specific decoder modules $\mathcal{G}_t$ for prediction in the domain-specific label spaces. For a given input image $x_t \in \mathcal{D}_t$ at incremental step t, our method learns a mapping $M_t(x_t, t; \mathcal{W}_s, \mathcal{W}_t) = \mathcal{G}_t(\mathcal{F}(x_t, t; \mathcal{W}_s, \alpha_t))$, composed of a set of shared, domain-invariant parameters $\mathcal{W}_s$ which are universal for all domains and a domain-specific set of parameters $\mathcal{W}_t$ which are exclusive to its respective domain $\mathcal{D}_t$. The idea is to factorize the network latent space such that homogeneous semantic representation among all datasets gets captured in the shared parameters $\mathcal{W}_s$. In contrast, heterogeneous dataset statistics are learned by the corresponding domain-specific layers $\mathcal{W}_t$. This way, *by construction*, we get a good stability-plasticity trade-off. Our approach is designed for segmentation models with a ResNet [15] based encoder backbone. We modify each residual unit in the encoder to a Domain-Aware Residual Unit (DAU).

Domain-Aware Residual Unit: As shown in Figure 3, each DAU consists of (i) a set of domain-invariant parameters $W_s = \{w_1, w_2\}$, and (ii) a set of domain-specific parameters for each task t given as, $\alpha_t = \{\alpha^w, \alpha^s, \alpha^b\}$. w_1, w_2 are the 3×3 convolutional layers present in a traditional residual unit [15] and are shared among all domains. Domain-specific layers in the DAU are of two kinds: (i) Domain-Specific Parallel Residual Adapter layers (DS-RAP), and (ii) Domain-Specific Batch Normalization layers (DS-BN). We use the concept of Parallel Residual Adapters (RAP)

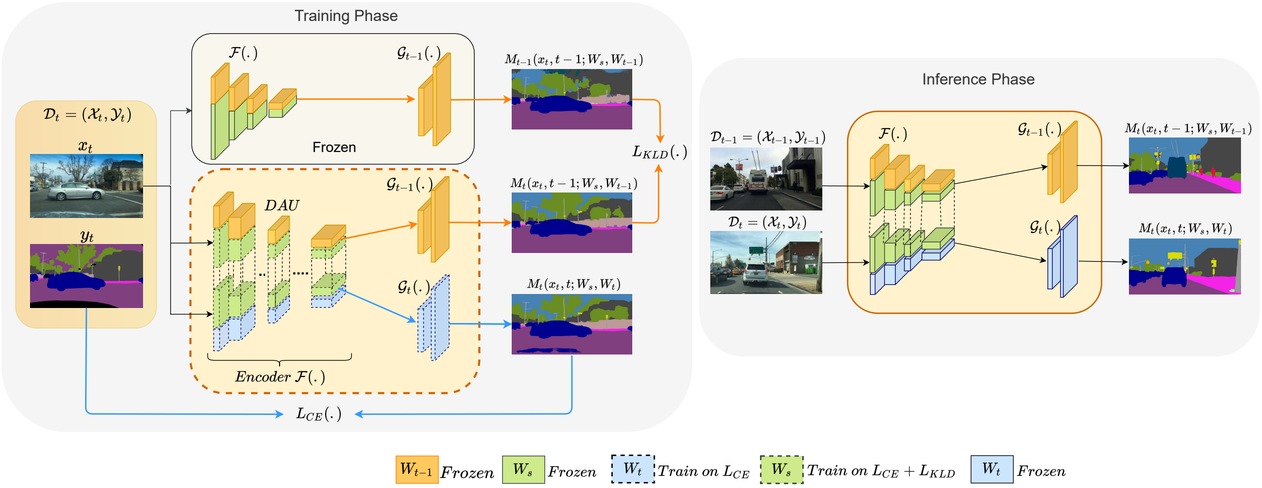


Figure 2: **Multi-domain incremental semantic segmentation framework.** *Training Phase:* Training on domain D_t in incremental step t. Our model consists of domain-specific decoders and a single encoder. The encoder is composed of Domain-Aware Residual Units (DAU), illustrated in Figure 3. Layers indicated in green inside the encoder (W_s) are common to all domains. They have been shown separately for illustration of separate domain-specific paths. Domain-specific layers of current domain (W_t) are in blue; domain-specific layers of previous domain (W_{t-1}) are in orange. *Inference phase:* For evaluation on a particular domain, the corresponding domain-specific path is used to get the segmentation output.

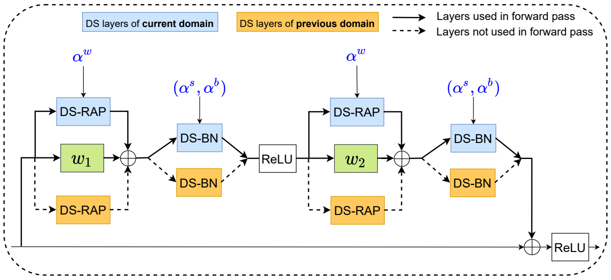


Figure 3: **Domain-Aware Residual Unit (DAU).** These units constitute encoder $\mathcal{F}(\cdot)$. For testing current domain, domain-specific layers for current domain (blue) and shared layers (green) are used for forward pass through DAU.

from [37] and modify its optimization for our setting. DS-RAP layers, α^w are 1×1 convolutional layers added to the shared convolutional layers in parallel. They act as layer-level domain adapters. Differently from the residual adapter module in [37], we make the Batch Normalization layers also domain-specific. In BN, the normalized input is scaled and shifted as $s \odot x + b$; here, (α^s, α^b) denote the learnable scale and shift parameters of the DS-BN layers. The shared weights act like universal filter banks, learning domain-generalized knowledge. In contrast, the DS-RAP and DS-BN layers are exclusive to their particular domain, responsible for learning domain-specific features.

Existing residual adapter-based approaches for image classification [19, 36, 37] freeze their shared parameters W_s to a generic initialization such as Imagenet [10] pre-trained weights and train only domain-specific parameters. We find that Imagenet initialization for W_s does not work well for fine-grained tasks like semantic segmentation (Section 5). Thus, instead of freezing shared parameters W_s, we fine-tune them on the new domain $\mathcal{D}_t$, in an end-to-end training. We propose an optimization strategy that makes W_s parameters learn domain-agnostic features and a different optimization for W_t parameters to make them domain-specific, as described below.

Optimization Strategy. *Domain-specific parameters:* For a particular task t, the composition of domain-specific parameters is given as $W_t = \{\alpha_t, \mathcal{G}_t\}$. To learn a new domain $\mathcal{D}_t$ at step t, we add new parameters W_t to the model from the previous step M_{t-1} and call this model M_t. We initialize all W_t from W_{t-1}, except the output classifier layer which is randomly initialized (label space $\mathcal{Y}_t$ may be different from $\mathcal{Y}_{t-1}$). We refer to this initialization strategy as $init_{W_t}$. The domain-specific layers W_t are trained only on the task-specific loss for domain $\mathcal{D}_t$ given as:

$$L_{CE_t} = \frac{1}{N} \sum_{x_t \epsilon \mathcal{D}_t} \psi_t(y_t, \mathcal{G}_t(\mathcal{F}(x_t, t; \mathcal{W}_s, \alpha_t)) \quad (1)$$

where ψ_t is the task-specific softmax cross-entropy loss function over the label space $\mathcal{Y}_t$. All domain-specific layers of previous domains $W_{t-i}, 0 < i < t$ remain frozen during current domain training.

Domain-invariant parameters: The W_s layers in the encoder are shared among all tasks. In IL step t, we initialize these weights from the corresponding weights in M_{t-1}. In addition to the task-specific cross entropy loss L_{CE_t}, we use a regularization loss L_{KLD} to optimize the shared weights:

$$q_i^s = M_t(x_t, t-1; W_s, W_{t-1}) \tag{2}$$

$$q_i^t = M_{t-1}(x_t, t-1; W_s, W_{t-1}) \tag{3}$$

$$L_{KLD} = \lambda_{KLD} \cdot \sum_{i=1}^{t-1} \sum_{x_t \epsilon \mathcal{D}_t} \phi(q_i^s, q_i^t) \tag{4}$$

where q_i^s is the prediction map of the current model M_t on the current input x_t for the *previous task* $t-1$; q_i^t is the prediction map of the previous model M_{t-1} on the current input x_t for the *previous task* $t-1$; ϕ is the KL-divergence (KLD) loss between these two softmax probability distribution maps, computed and summed over each previously learned task i, $0 < i < t$; λ_{KLD} is the regularization hyperparameter for KL-divergence. KLD here effectively *distills domain knowledge* from the teacher q_i^t to the student q_i^s, and can be seen as domain adaptive knowledge distillation [22]. Total loss for domain-invariant parameters W_s is hence given as:

$$L_{W_s} = L_{CE_t} + L_{KLD} \tag{5}$$

Optimization of the W_s and W_t parameters in our model is at a differential learning rate, *dlr*. We observe that optimizing W_s at a learning rate 100× lower than the learning rate of W_t stabilizes W_s w.r.t. W_{t-1} and prevents forgetting. As shown in Section 5, a combination of the $init_{W_t}$ and *dlr* learning strategies is responsible for preserving old knowledge and learning new knowledge simultaneously. We summarize the optimization protocol in Algorithm 1.

When the shared weights W_s are trained on the domain-specific loss L_{CE_t} of the current step, they learn the current domain's features and quickly forget the domain-specific representation learned on the previous domain. The *dlr* strategy prevents the shared weights W_s from drifting away from the domain-specific features learned in the previous step, when learning the current domain. This model is referred to as DAU-FT-*dlr* in our results (Table 5). Minimizing KLD between the output feature maps of the previous and current models *preserves previous tasks' domain knowledge in the shared weights*. A combination of L_{KLD}, L_{CE_t} and *dlr* learning strategy thus help train domain-invariant shared layers in the encoder. W_t weights are domain-specific as they are trained only on the domain-specific loss. Together, the above steps reparametrize the model into domain-specific and domain-invariant features, which in turn achieve strong performance on the new domain while retaining performance on older domains.

Inference Phase. For a query image $x_t \in \mathcal{D}_t, t \in T$ (set of tasks the model has learned on so far), our model gives an output segmentation map of pixel-wise predictions $\hat{y}_t = M_t(x_t, t)$ over the label space $\mathcal{Y}_t$. For evaluation on any domain $\mathcal{D}_t$, only the corresponding domain-specific α_t and $\mathcal{G}_t$ get activated in the forward pass. In effect, our model has multiple domain-specific paths with a large degree of parameter sharing.

Algorithm 1 Training protocol in the t^{th} incremental step

```
Require:
  $\mathcal{D}_t$: new data of current step $t$
  $M_{t-1}$: model from previous step $t-1$
Initialize:
  $M_t \leftarrow$ add new DS layers $W_t$ to $M_{t-1}$ (for learning $\mathcal{D}_t$)
  $init_{W_t}$: $W_t$ in $M_t \leftarrow W_{t-1}$ in $M_{t-1}$
Freeze: DS weights of all previous domains; $W_{t-i}, 0 < i < t$
for epochs do
    for mini-batch do
        Forward pass $M_t(x_t, t)$ via $W_t$
        Compute task-specific loss $L_{CE_t}$ for $\mathcal{D}_t$ by Eq. 1
        Forward pass $M_t(x_t, t-1)$ via $W_{t-1}$, Eq. 2
        Forward pass $M_{t-1}(x_t, t-1)$ via $W_{t-1}$ of $M_{t-1}$, Eq. 3
        Compute KLD loss $L_{KLD}$ by Eq. 4
        Compute $L_{W_s}$ by Eq. 5
        Update:
            $L_{CE_t}$ on $W_t$ at standard network learning rate $lr$
            $L_{W_s}$ on $W_s$ at a lower learning rate $dlr$
    end for
end for
Discard training data $\mathcal{D}_t$
```

Experimental Setting. Consider a two-task IL setting where the goal is to take a model trained on geographical domain D_A and incrementally learn on another domain D_B. Multi-domain incremental semantic segmentation consists of two scenarios: (i) Case 1: D_A and D_B have a domain shift, but aligned label spaces, i.e. $D_A \neq D_B, \mathcal{Y}_A = \mathcal{Y}_B$; (ii) Case 2: D_A and D_B have a domain shift as well as non-overlapping label spaces, i.e. $D_A \neq D_B, \mathcal{Y}_A \neq \mathcal{Y}_B$. Our proposed approach for MDIL tackles both these scenarios.

4. Experiments and Results

Datasets. We perform IL over three highly diverse, large-scale urban driving datasets collected from different geographic locations. The Cityscapes dataset (CS) [7] is a standard autonomous driving dataset of daytime images collected from urban streets of 50 European cities. It contains 19 labels, captured in 2975 training and 500 validation images. The Berkeley Deep Drive dataset (BDD) is a widespread collection of road scenes spanning diverse weather conditions and times of the day in the United States [55]. It covers residential areas and highways along with urban streets. It has 7000 training and 1000 validation images. The Indian Driving Dataset (IDD) has unconstrained road environments collected from Indian cities [45]. These road environments captured in 6993 training and 781 validation images are highly unstructured with unique labels like billboard, auto-rickshaw, animal, etc. In our experiments, we adhere to the default label spaces of these datasets, i.e., we use the 19 labels in Cityscapes and

IL Step	Step 1 CS		Step 2: $D_A \neq D_B, \mathcal{Y}_A = \mathcal{Y}_B$ $CS \rightarrow BDD$			Step 2: $D_A \neq D_B, \mathcal{Y}_A \neq \mathcal{Y}_B$ $CS \rightarrow IDD$		
Methods	CS ↑	$\Delta_m\%$ ↓	CS ↑	BDD ↑	$\Delta_m\%$ ↓	CS ↑	IDD ↑	$\Delta_m\%$ ↓
Single-task	72.55		72.55	54.1		72.55	61.97	
Multi-task	72.55		69.42	57.69	1.16% (↑)	71.11	60.85	1.89%
FT	72.55	0.0%	40.05 (-32.5)	52.74	23.66%	36.81 (-35.74)	61.56	24.96%
FE	72.55	0.0%	72.55 (-0.00)	42.93	10.32%	72.55 (-0.00)	45.69	13.14%
FT (Single-Head)	72.55		47.42 (-25.13)	50.89	20.29%	36.82 (-35.73)	53.79	31.22%
LwF [24]	72.55		58.66 (-13.89)	43.26	19.59%	62.63 (-9.92)	42.89	22.23%
ILT [32]	72.55		56.84 (-15.71)	32.97	30.36%	54.37 (-18.18)	25.07	42.30%
Ours	71.82	1.01%	65.21 (-7.34)	55.73 (+1.63)	**3.55%**	64.58 (-7.97)	59.11 (-2.86)	**7.80%**

Table 2: *Results of 2-task incremental settings.* We report performance on all datasets, after incrementally learning on the current dataset D_t in step t. Arrows indicate order of learning. Parenthesis show drop/gain in performance w.r.t single-task baseline for the corresponding dataset. Lower $\Delta_m\%$ indicates better stability-plasticity trade-off and overall performance.

IL Step	Step 3: $D_A \neq D_B, \mathcal{Y}_A \neq \mathcal{Y}_B$ $CS \rightarrow BDD \rightarrow IDD$			
Methods	CS ↑	BDD ↑	IDD ↑	$\Delta_m\%$ ↓
Single-task	72.55	54.1	61.97	
Multi-task	69.37	58.13	59.37	0.38%
FT	30.49 (-42.06)	32.05 (-22.05)	60.65	33.62%
FE	72.55 (-0.00)	42.93 (-11.17)	46.09	15.42%
Ours	59.19 (-13.36)	49.66 (-4.44)	59.16	**10.39%**
	$CS \rightarrow IDD \rightarrow BDD$			
Methods	CS ↑	IDD ↑	BDD ↑	$\Delta_m\%$ ↓
Single-task	72.55	61.97	54.1	
Multi-task	69.37	59.37	58.13	0.38%
FT	36.19 (-36.36)	26.3 (-35.67)	53.37	36.34%
FE	72.55 (-0.00)	45.69 (-16.28)	43.06	15.56%
Ours	62.55 (-10.0)	53.85 (-8.12)	55.90	**7.85%**

Table 3: *Results of 3-task incremental learning settings.* $CS \rightarrow BDD \rightarrow IDD$ model was trained on CS in step 1, on BDD in step 2. Performance is reported on all 3 datasets after it is incrementally trained on IDD in step 3. Similarly, we report results on the $CS \rightarrow IDD \rightarrow BDD$ setting.

BDD100k, and IDD level 3, which has 26 labels.

Evaluation Metrics. We use the mean Intersection-over-Union (mIoU) metric to evaluate the semantic segmentation performance of a model on each dataset, following standard practice. Similar to [19], we quantify the overall IL performance of a model m, as the average per-task drop in semantic segmentation performance (mIoU) with respect to the corresponding single-task baseline b:

$$\Delta_m\% = \frac{1}{T}\sum_{t=1}^{T} \frac{mIoU_{m,t} - mIoU_{b,t}}{mIoU_{b,t}} \tag{6}$$

where $mIoU_{m,t}$ is the segmentation accuracy of model m on task t. $\Delta_m\%$ quantifies the stability-plasticity trade-off to give an overall score of IL performance.

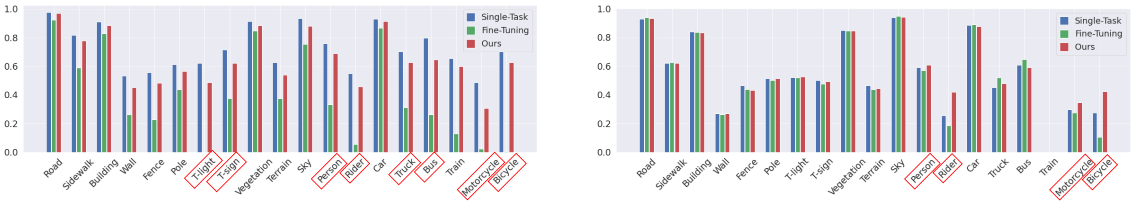
Implementation Details. We use ERFNet [39] as the backbone for implementing this work, as it allows the dynamic addition of our modules seamlessly. Similar to [18, 48, 5], we report mIoU on the standard validation sets of these datasets. More details are provided in the supplementary.

IL Step	Step 1 IDD		Step 2: $D_A \neq D_B, \mathcal{Y}_A \neq \mathcal{Y}_B$ $IDD \rightarrow BDD$		
Methods	IDD ↑	$\Delta_m\%$ ↓	IDD ↑	BDD ↑	$\Delta_m\%$ ↓
Single-task	61.97		61.97	54.1	
Multi-task	61.97		61.05	56.05	1.06% (↑)
FT	61.97	0.0	27.33	52.88	29.08%
FE	61.97	0.0	61.97	46.23	7.27%
Ours	62.60	1.02 (↑)	57.36	55.73	**2.21%**
	BDD		$BDD \rightarrow IDD$		
Methods	BDD ↑	$\Delta_m\%$ ↓	BDD ↑	IDD ↑	$\Delta_m\%$ ↓
Single-task	54.1		54.1	61.97	
Multi-task	54.1		56.05	61.05	1.06% (↑)
FT	54.1	0.0	30.72	59.9	23.28%
FE	54.1	0.0	54.1	47.24	11.88%
Ours	52.1	3.70	50.92	57.21	**6.78%**

Table 4: Results of domain ordering on $IDD \rightarrow BDD$ and $BDD \rightarrow IDD$ 2-task incremental settings.

Results. We show detailed analysis on two, 2-task settings $CS \rightarrow BDD$ and $CS \rightarrow IDD$ to compare the two possible cases. We also show results on 3-task settings including, $CS \rightarrow BDD \rightarrow IDD$ and $CS \rightarrow IDD \rightarrow BDD$.

Incremental Learning Baselines: We compare our proposed approach with four standard IL baselines. The single-task baseline denotes datasets trained independently on separate models, which one can consider as the gold standard or the upper bound for IL performance. This is used to compute catastrophic forgetting and overall evaluation score $\Delta_m\%$ across our experiments. The multi-task model gives the joint training performance, where a single multi-decoder model is trained offline on all the datasets together (note that this violates the IL setting, but is shown for completeness). Fine-tuning (FT) is a standard baseline in IL, where a model is fine-tuned on the newer domain without any explicit effort to mitigate forgetting. This can be considered as a lower bound for our experiments. In feature extraction (FE), we freeze all encoder weights and only train the new domain's decoder weights. Fine-tuning gives the maximum plasticity and minimum stability, while feature extraction exhibits maximum stability and minimum plasticity. We also compare our method against an existing class-



(a) Cityscapes

(b) BDD100k

Figure 4: Analysis of class-wise accuracy (IoU) on (a) CS and (b) BDD after incrementally learning from CS → BDD. Our model is able to show significant improvement for the categories marked in red.

IL method for segmentation, ILT [32]. This is a single-head architecture comparison along with the fine-tuning (single-head). We also compare against learning without forgetting LwF [24] implemented as a multi-head.

Cityscapes → BDD100k : In this setting, we start by learning a model on Cityscapes (CS) in step 1, followed by incrementally learning the same model on BDD100k (BDD) in step 2. CS and BDD datasets have a common label space of 19 labels. Hence, there is a domain shift when going from CS → BDD, but their label spaces are aligned. As shown in Table 2, using our model, forgetting on CS has been mitigated by 25.16% (w.r.t the fine-tuning baseline) and is only 7.34% below the single-task upper limit. A comparison of the class-wise performance of our approach with the fine-tuning baseline is given in Figure 4. Our model mitigates forgetting in all 19 classes, and retains performance by a significantly large margin ($\geq$ 30%) on *safety-critical classes* such as traffic light, traffic sign, person, rider, truck, bus and bicycle. Importantly, we observe that our proposed model has surpassed the single-task model performance on BDD by 1.63%. We hypothesize that this forward transfer is achieved since our model captures the domain-specific characteristics of the dataset distributions of CS and BDD in the domain-specific parameters. Class-wise analysis are explained in detail in supplementary material.

Cityscapes → IDD : As presented in Table 2, we first learn a model on CS in step 1, then incrementally learn on IDD in step 2. CS has a label space of 19 labels, and IDD has a label space of 26 labels such that a subset of 17 road classes is common. 2 classes are exclusive to CS, while 8 classes are exclusive to IDD. This scenario includes both a domain shift as well as label misalignment. Forgetting on CS is mitigated by 27.77% by our model. This shows that despite the label misalignment, our approach can retain old task performance in CS → IDD almost as well as it does in the CS → BDD setting (forgetting on CS is 7.97% after learning on IDD as compared to the 7.34% after learning on BDD).

3-Task Incremental Settings : In Table 3, we show results for Cityscapes → BDD100k → IDD and Cityscapes → IDD → BDD100k settings. We also explore different sequences of domain ordering in Table 4. These results show that our model is generalizable with respect to domain ordering.

Methods	L_{KLD}	*dlr*	$init_{W_t}$	DAU	CS	BDD
Single-task	×	×	×	×	72.55	54.1
DAU-FT	×	×	×	✓	1.34	49.96
DAU-FT-*dlr*1	×	×	✓	✓	8.41	54.51
DAU-FT-rinit	×	✓	×	✓	46.4	54.50
DAU-FT-*dlr*	×	✓	✓	✓	58.4	57.03
Ours	✓	✓	✓	✓	65.21	55.73

Table 5: Ablation studies on the contribution of each component of our proposed model for the $CS \rightarrow BDD$ setting. mIoU on CS, BDD is reported after learning on BDD.

More exhaustive permutations are given in supplementary.

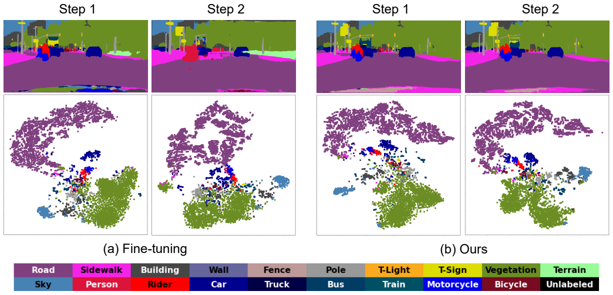
5. Ablation Studies and Analysis

Significance of optimization strategies. In this section, we study the significance of the optimization and initialization strategy we use for attaining a stability-plasticity trade-off in our IL setting. Table 5 shows these results. In IL step t, the W_s and W_{t-1} weights in current model M_t are initialized from the corresponding layers in M_{t-1} (for all experiments). The DS weights of current domain W_t can either be randomly initialized or initialized from the DS weights of previous domain W_{t-1}. We call the latter as $init_{W_t}$. In DAU-FT model, we perform vanilla fine-tuning of the W_s and randomly initialize W_t weights using the standard learning rate on the task-specific loss L_{CE_t}. This performs poorly on both old and new domains, catastrophically forgetting the previous domain (1.34% mIoU).

Next, we define a differential learning rate *dlr* as $\frac{LR_{W_t}}{LR_{W_s}}$. If we use *dlr=1* and fine-tune both W_s and W_t using same LR, the W_s parameters learn on the new domain (BDD) and forget the previously learned representation on CS (DAU-FT-*dlr1* in table). All *dlr* models use $init_{W_t}$ unless stated otherwise. As we decrease the LR of W_s w.r.t W_t, stability of the model w.r.t the previous domain increases and plasticity w.r.t. the new domain decreases. We find that a *dlr* value of 100 gives a good stability-plasticity trade-off. This model is referred to as DAU-FT-*dlr* in the table. Applying L_{KLD} on the shared weights W_s of DAU-FT-*dlr* model further mitigates forgetting by 7.91% and is our proposed model (*ours*). It is important that the DS weights W_t not be randomly initialized. If they are randomly initialized, there is a performance drop as given by DAU-FT-rinit.

IL Step	Step 1 CS		Step 2: $D_A \neq D_B, \mathcal{Y}_A = \mathcal{Y}_B$ $CS \rightarrow BDD$			Step 2: $D_A \neq D_B, \mathcal{Y}_A \neq \mathcal{Y}_B$ $CS \rightarrow IDD$			Step 3: $D_A \neq D_B, \mathcal{Y}_A \neq \mathcal{Y}_B$ $CS \rightarrow BDD \rightarrow IDD$			
Methods	CS ↑	$\Delta_m\%$ ↓	CS ↑	BDD ↑	$\Delta_m\%$ ↓	CS ↑	IDD ↑	$\Delta_m\%$ ↓	CS ↑	BDD ↑	IDD ↑	$\Delta_m\%$ ↓
Single-task	72.55		72.55	54.1		72.55	61.97		72.55	54.1	61.97	
RCM-NFI [19]	1.98	97.27%	1.98	1.32	97.42%	1.98	1.13	97.72%	1.98	1.32	1.13	97.67%
RCM-I [19]	63.13	12.98%	63.13	47.94	12.19%	63.13	55.66	11.58%	**63.13**	47.94	55.66	11.52%
RAS-I [36]	61.65	15.02%	61.65	48.05	13.10%	61.65	54.09	13.87%	61.65	48.05	54.09	12.97%
RAP-I [37]	58.43	19.46%	58.43	46.34	16.90%	58.43	50.97	18.61%	58.43	46.34	51.27	17.02%
Ours	**71.82**	1.01%	**65.21**	**55.73**	**3.55%**	**64.58**	**59.11**	**7.80%**	59.19	**49.66**	**59.16**	**10.39%**

Table 6: Comparison with other residual adapter-based architectures. Lower score $\Delta_m\%$ indicates better overall performance.



(a) Fine-tuning

(b) Ours

Figure 5: Consider the CS → BDD setting. (a) t-SNE of CS features for the *fine-tuning* model in step 1 and step 2. (b) t-SNE of CS features for *our* proposed model in step 1 and step 2. Fine-tuning distorts the latent space representation of CS learned in step 1. **Our model preserves the latent space of CS after learning on BDD in step 2.**

Comparison with other residual adapter-based architectures. We show a comparison of our approach against three state-of-the-art residual adapter-based methods in Table 6. RAP-I [37] denotes the parallel residual adapter with W_s weights frozen from pre-training on ImageNet. Our proposed optimization strategy outperforms this by a large margin. RAS-I is the series residual adapter [36]. While RAP and RAS contain layer-level residual adapters α in the shared encoder, RCM (Reparameterized Convolutions for Multi-task learning) [19] is a block-level adapter, wherein a 1×1 convolution is added to each residual block in series. Only the task-specific adapter layers are trained in each of these models, while the W_s weights are frozen to Imagenet pre-training. RCM-NFI applies normalized feature fusion to the output of RCM layers in the RCM-I model (given as RCM-NFF in [19]). This model does not perform well in our setting. The RAP is a plug-and-play residual adapter that can easily be plugged into existing segmentation models. The RAS and RCM are series adapters and need to be included when ResNet is pre-trained on Imagenet for best performance. We find that the RAP adapter is better suited when fine-tuning the shared weights W_s (please see supplementary material).

Latent space visualization. Figure 5 shows a t-SNE [44] visualization of the features extracted from the last layer of the encoder $\mathcal{F}(.)$. We show the latent space of features extracted from a Cityscapes sample image before and after incrementally learning over the next domain BDD (CS → BDD setting). In (a), we train a single-task baseline on CS in step 1 and fine-tune it on BDD in step 2. In (b), we train *our* model on CS in step 1 and incrementally learn on BDD in step 2. The latent space of CS gets significantly distorted after fine-tuning the model on BDD (see (a) Step 2). On the other hand, our model can preserve the learned feature representation even after adding BDD to the model. While the smaller classes (with fewer pixels) had distinct separation in step 1, inter-cluster separation has suffered in step 2 for the fine-tuning model. It can be observed that classes like rider, motorcycle, traffic light, bicycle and truck have moved in a small space towards the center and are indistinguishable, causing confusion in predicting these classes. Our model in step 2 is able to preserve distinct clusters and maintain inter-class separation on the old domain CS.

6. Conclusion

We define the problem of multi-domain incremental semantic segmentation and present a parameter-isolation based dynamic architecture that leads to a significant improvement over the baselines. Our model allows domain-specific paths for different domains, while having a large degree of parameter sharing (78.83%) in a universal model. We compare two scenarios of fully overlapping and partially overlapping label spaces to understand the challenges involved in multi-domain incremental semantic segmentation. From the CS → BDD and CS → IDD cases, we infer that: (i) our approach works equally well in mitigating forgetting in the two scenarios; (ii) if the labels spaces are aligned, forward transfer can occur; (iii) misalignment of label spaces is likely to cause some domain interference on the new domain, although our method provides promising performance across the domains. We demonstrate through visualizations how the proposed method maintains the latent space of classes across domains. This enables learning novel domains while preserving the representations of the previous domains at the same time.

Acknowledgments. This work was partly funded by IHub-Data at IIIT Hyderabad, and DST (IMPRINT program).

References

[1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory Aware Synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 139–154, 2018.

[2] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert Gate: Lifelong Learning with a Network of Experts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3366–3375, 2017.

[3] Adrian Bulat, Jean Kossaifi, Georgios Tzimiropoulos, and Maja Pantic. Incremental Multi-Domain Learning with Network Latent Tensor Factorization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10470–10477, 2020.

[4] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-End Incremental Learning. In *Proceedings of the European conference on Computer Vision (ECCV)*, pages 233–248, 2018.

[5] Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulo, Elisa Ricci, and Barbara Caputo. Modeling the Background for Incremental Learning in Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9233–9242, 2020.

[6] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian Walk for Incremental Learning: Understanding Forgetting and Intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 532–547, 2018.

[7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016.

[8] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. Continual learning: A comparative study on how to defy forgetting in classification tasks. *arXiv preprint arXiv:1909.08383*, 2(6), 2019.

[9] Matthias Delange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. IEEE, 2009.

[11] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without Memorizing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5138–5146, 2019.

[12] Arthur Douillard, Yifu Chen, Arnaud Dapogny, and Matthieu Cord. PLOP: Learning without Forgetting for Continual Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4040–4050, 2021.

[13] Sayna Ebrahimi, Franziska Meier, Roberto Calandra, Trevor Darrell, and Marcus Rohrbach. Adversarial Continual Learning. *arXiv preprint arXiv:2003.09553*, 1, 2020.

[14] Yunhui Guo, Yandong Li, Liqiang Wang, and Tajana Rosing. Depthwise Convolution Is All You Need for Learning Multiple Visual Domains. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8368–8375, 2019.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[16] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. *stat*, 1050:9, 2015.

[17] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a Unified Classifier Incrementally via Rebalancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 831–839, 2019.

[18] Tarun Kalluri, Girish Varma, Manmohan Chandraker, and CV Jawahar. Universal Semi-Supervised Semantic Segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5259–5270, 2019.

[19] Menelaos Kanakis, David Bruggemann, Suman Saha, Stamatios Georgoulis, Anton Obukhov, and Luc Van Gool. Reparameterizing Convolutions for Incremental Multi-Task Learning without Task Interference. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 689–707. Springer, 2020.

[20] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

[21] Marvin Klingner, Andreas Bär, Philipp Donn, and Tim Fingscheidt. Class-Incremental Learning for Semantic Segmentation Re-Using Neither Old Data Nor Old Labels. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–8. IEEE, 2020.

[22] Divya Kothandaraman, Athira M Nambiar, and Anurag Mittal. Domain Adaptive Knowledge Distillation for Driving Scene Semantic Segmentation. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV Workshops)*, pages 134–143, 2021.

[23] Jogendra Nath Kundu, Rahul Mysore Venkatesh, Naveen Venkat, Ambareesh Revanur, and R Venkatesh Babu. Class-Incremental Domain Adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 53–69, 2020.

[24] Zhizhong Li and Derek Hoiem. Learning without Forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(12):2935–2947, 2017.

[25] Xialei Liu, Hao Yang, Avinash Ravichandran, Rahul Bhotika, and Stefano Soatto. Multi-Task Incremental Learning for Object Detection. *arXiv e-prints*, pages arXiv–2002, 2020.

[26] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a Single Network to Multiple Tasks by Learning to Mask Weights. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 67–82, 2018.

[27] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7765–7773, 2018.

[28] Massimiliano Mancini, Elisa Ricci, Barbara Caputo, and Samuel Rota Bulo. Adding New Tasks to a Single Network with Weight Transformations using Binary Masks. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 180–189, 2018.

[29] Michael McCloskey and Neal J Cohen. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

[30] Ke Mei, Chuang Zhu, Jiaqi Zou, and Shanghang Zhang. Instance Adaptive Self-Training for Unsupervised Domain Adaptation. *arXiv preprint arXiv:2008.12197*, 2020.

[31] Martial Mermillod, Aurélia Bugaiska, and Patrick Bonin. The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects. *Frontiers in psychology*, 4:504, 2013.

[32] Umberto Michieli and Pietro Zanuttigh. Incremental Learning Techniques for Semantic Segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 3205–3212, 2019.

[33] Umberto Michieli and Pietro Zanuttigh. Continual Semantic Segmentation via Repulsion-Attraction of Sparse and Disentangled Latent Representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1114–1124, 2021.

[34] Oleksiy Ostapenko, Mihai Puscas, Tassilo Klein, Patrick Jahnichen, and Moin Nabi. Learning to Remember: A Synaptic Plasticity Driven Framework for Continual Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11321–11329, 2019.

[35] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual Lifelong Learning with Neural Networks: A Review. *Neural Networks*, 113:54–71, 2019.

[36] S-A Rebuffi, H. Bilen, and A. Vedaldi. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[37] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Efficient parametrization of multi-domain deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8119–8127, 2018.

[38] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. iCaRL: Incremental Classifier and Representation Learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2001–2010, 2017.

[39] Eduardo Romera, José M Alvarez, Luis M Bergasa, and Roberto Arroyo. ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 19(1):263–272, 2017.

[40] Amir Rosenfeld and John K Tsotsos. Incremental Learning Through Deep Adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 42(3):651–663, 2018.

[41] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive Neural Networks. *arXiv preprint arXiv:1606.04671*, 2016.

[42] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual Learning with Deep Generative Replay. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[43] Pravendra Singh, Vinay Kumar Verma, Pratik Mazumder, Lawrence Carin, and Piyush Rai. Calibrating CNNs for Lifelong Learning. *Advances in Neural Information Processing Systems (NIPS)*, 33, 2020.

[44] Laurens Van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of machine learning research*, 9(11), 2008.

[45] Girish Varma, Anbumani Subramanian, Anoop Namboodiri, Manmohan Chandraker, and CV Jawahar. IDD: A Dataset for Exploring Problems of Autonomous Navigation in Unconstrained Environments. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1743–1751. IEEE, 2019.

[46] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. ADVENT: Adversarial Entropy Minimization for Domain Adaptation in Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2517–2526, 2019.

[47] Haoran Wang, Tong Shen, Wei Zhang, Ling-Yu Duan, and Tao Mei. Classes Matter: A Fine-grained Adversarial Approach to Cross-domain Semantic Segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 642–659. Springer, 2020.

[48] Li Wang, Dong Li, Yousong Zhu, Lu Tian, and Yi Shan. Cross-Dataset Collaborative Learning for Semantic Segmentation. *arXiv preprint arXiv:2103.11351*, 2021.

[49] Chenshen Wu, Luis Herranz, Xialei Liu, Yaxing Wang, Joost van de Weijer, and Bogdan Raducanu. Memory Replay GANs: learning to generate images from new categories without forgetting. In *Conference on Neural Information Processing Systems (NIPS)*, 2018.

[50] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large Scale Incremental Learning. In *Proceedings of the IEEE/CVF Confer-*

ence on Computer Vision and Pattern Recognition (CVPR), pages 374–382, 2019.

[51] Zuxuan Wu, Xin Wang, Joseph E Gonzalez, Tom Goldstein, and Larry S Davis. ACE: Adapting to Changing Environments for Semantic Segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2121–2130, 2019.

[52] Markus Wulfmeier, Alex Bewley, and Ingmar Posner. Incremental Adversarial Domain Adaptation for Continually Changing Environments. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4489–4495. IEEE, 2018.

[53] Yanchao Yang and Stefano Soatto. FDA: Fourier Domain Adaptation for Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4085–4095, 2020.

[54] Jaehong Yoon, Eunho Yang, Jungtae Lee, and Sung Ju Hwang. Lifelong Learning with Dynamically Expandable Networks. In *Sixth International Conference on Learning Representations*. ICLR, 2018.

[55] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2636–2645, 2020.

[56] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual Learning Through Synaptic Intelligence. In *International Conference on Machine Learning*, pages 3987–3995. PMLR, 2017.

[57] Sicheng Zhao, Bo Li, Xiangyu Yue, Yang Gu, Pengfei Xu, Runbo Tan, Hu, Hua Chai, and Kurt Keutzer. Multi-source Domain Adaptation for Semantic Segmentation. In *Advances in Neural Information Processing Systems (NIPS)*, 2019.