
LEARNING WITH PRESERVING FOR CONTINUAL MULTITASK LEARNING

Hanchen David Wang*
Vanderbilt University
Nashville, TN 37235 USA
hanchen.wang.1@vanderbilt.edu

Siwoo Bae*
Vanderbilt University
Nashville, TN 37235 USA
siwoo.bae@vanderbilt.edu

Zirong Chen
Vanderbilt University
Nashville, TN 37235 USA
zirong.chen@vanderbilt.edu

Meiyi Ma
Vanderbilt University
Nashville, TN 37235 USA
meiyi.ma@vanderbilt.edu

ABSTRACT

Artificial intelligence systems in critical fields like autonomous driving and medical imaging analysis often continually learn new tasks using a shared stream of input data. For instance, after learning to detect traffic signs, a model may later need to learn to classify traffic lights or different types of vehicles using the same camera feed. This scenario introduces a challenging setting we term Continual Multitask Learning (CMTL), where a model sequentially learns new tasks on an underlying data distribution without forgetting previously learned abilities. Existing continual learning methods often fail in this setting because they learn fragmented, task-specific features that interfere with one another. To address this, we introduce Learning with Preserving (LwP), a novel framework that shifts the focus from preserving task outputs to maintaining the geometric structure of the shared representation space. The core of LwP is a Dynamically Weighted Distance Preservation (DWDP) loss that prevents representation drift by regularizing the pairwise distances between latent data representations. This mechanism of preserving the underlying geometric structure allows the model to retain implicit knowledge and support diverse tasks without requiring a replay buffer, making it suitable for privacy-conscious applications. Extensive evaluations on time-series and image benchmarks show that LwP not only mitigates catastrophic forgetting but also consistently outperforms state-of-the-art baselines in CMTL tasks. Notably, our method shows superior robustness to distribution shifts and is the only approach to surpass the strong single-task learning baseline, underscoring its effectiveness for real-world dynamic environments.²

1 Introduction

In critical applications such as intelligent driving, a system must continually adapt by learning new tasks from a consistent stream of sensory data. This paradigm is driven by practicality: when a new task is introduced, the cost of retrospectively annotating the entire existing dataset with the new labels is often unsustainable Golatkar, Achille, and Soatto (2020). It is far more efficient to instead leverage the existing data stream by acquiring labels only for the new task as needed. For instance, after a model learns to detect traffic signs, it can later be taught to classify other attributes like scene types using the same camera feed Shaheen et al. (2022); Kang, Kum, and Kim (2024). Similarly, in medical imaging, a model trained for tumor classification can be updated to recognize secondary characteristics such as tissue density or shape, all while using the same underlying patient scans An et al. (2025); Freeman et al. (2021). The central challenge, therefore, is to learn new tasks by acquiring new labels for a shared and potentially evolving input distribution.

*Equal contribution.

²Work accepted at AAAI-2026. The code is available at <https://github.com/AICPS-Lab/lwp>

We term this challenging real-world setting **Continual Multitask Learning (CMTL)**. CMTL combines challenges from both Multitask Learning (MTL) and Continual Learning (CL). In a typical CMTL scenario, a model is presented with a sequence of tasks, T_1, T_2, \dots, T_n . Each task introduces a new label set applied to inputs from the same sensor/input space (though their underlying distribution may differ across tasks). This setting is distinct from standard MTL, where all tasks are known and trained on concurrently, and it presents unique challenges not fully addressed by conventional Task-Incremental Learning (Task-IL) methods Van De Ven, Tuytelaars, and Tolias (2022). The key distinctions are summarized in Table 1.

Table 1: Comparison of Scenarios. CMTL uniquely requires learning new tasks sequentially on a shared data distribution without full access to past task labels.

Characteristic	MTL	Task-IL (Standard CL)	CMTL (Our Setting)
Tasks Arrive Sequentially	X	✓	✓
All Task Data Available Concurrently	✓	X	X
Tasks Share an Input Domain	✓	(Often Not)	✓
Goal: Learn Shared Representation	✓	X	✓
Goal: Mitigate Forgetting	N/A	✓	✓

The primary challenge in CMTL is twofold: the model must 1) retain knowledge from previous tasks to prevent catastrophic forgetting (a core CL goal), and 2) develop robust, shared representations that benefit multiple distinct tasks (a core MTL goal), all without having simultaneous access to the complete labeled data for all tasks. This is especially difficult when the underlying data distribution shifts over time (a non-stationary setting), which can exacerbate task interference.

Although CMTL can be formally categorized as a case of Task-IL, its strong emphasis on building a unified representation from a shared input domain exposes a key weakness in conventional CL methods. These approaches are primarily designed to prevent catastrophic forgetting, often by isolating task-specific knowledge Kirkpatrick et al. (2017); Zenke, Poole, and Ganguli (2017a); Ma et al. (2020). As our experiments confirm (Table 2), this strategy frequently struggles in the CMTL setting, leading to performance below that of even single-task baselines. Handling heterogeneous tasks on shared inputs requires unified representations Jiao et al. (2025); Huang et al. (2023), yet conventional CL methods fail this requirement by design, relying on parameter freezing and replay buffers that isolate rather than integrate task-specific knowledge.

To address these challenges, we introduce **Learning with Preserving (LwP)**, a framework designed specifically for the CMTL setting. Instead of focusing only on task outputs, LwP directly preserves the integrity of the shared representation space throughout sequential training. Its core principles are: **(i)**, a novel regularization term (the DWDP loss) to explicitly maintain the geometric structure of the model’s latent space and prevent representation drift, **(ii)**, a stabilized representation space to preserve the implicit knowledge encoded in the geometric relationships between data points, and **(iii)**, a framework operates without a replay buffer, making it efficient for privacy-constrained applications.

The main contributions of this paper are: **(1)** We formally define and analyze CMTL, and we demonstrate that conventional CL methods are often ill-suited for this context. **(2)** We propose Learning with Preserving (LwP), a novel, replay-free framework whose key innovation is a Dynamically Weighted Distance Preservation (DWDP) loss function that maintains the geometric integrity of the latent representation space, mitigating catastrophic forgetting while promoting knowledge sharing. **(3)** We conduct extensive evaluations on image and time-series benchmarks, showing that LwP consistently and significantly outperforms state-of-the-art baselines and, unlike other methods, surpasses the performance of independently trained single-task models, especially in scenarios with distribution shifts.

2 Problem Formulation: Continual Multitask Learning

CMTL is a sequential learning scenario involving T tasks $\{\mathcal{T}_t\}_{t=1}^T$. Each task \mathcal{T}_t is associated with a label space \mathcal{Y}_t and involves learning a mapping $f_t : \mathcal{X} \rightarrow \mathcal{Y}_t$. At each time step t , we receive a dataset $D_t = \{(x_i, y_i^t)\}_{i=1}^{n_t}$ where the input x_i is drawn from a task-specific distribution $x_i \sim P_X^{(t)}$, and $y_i^t \in \mathcal{Y}_t$ is the corresponding label for task \mathcal{T}_t . Note that for time t , only label y_i^t is available.

In simpler CMTL settings, the distribution is stationary ($P_X^{(t)} = P_X^{(j)}$ for all t, j), while in more challenging settings, it can be non-stationary ($P_X^{(t)} \neq P_X^{(j)}$ for $t \neq j$). Our goal is to find a predictor $\varphi(x; \theta_s, \theta_t) : \mathcal{X} \rightarrow \mathcal{Y}_1 \times \dots \times \mathcal{Y}_T$ parameterized by a set of shared parameters θ_s and task-specific parameters θ_t , such that

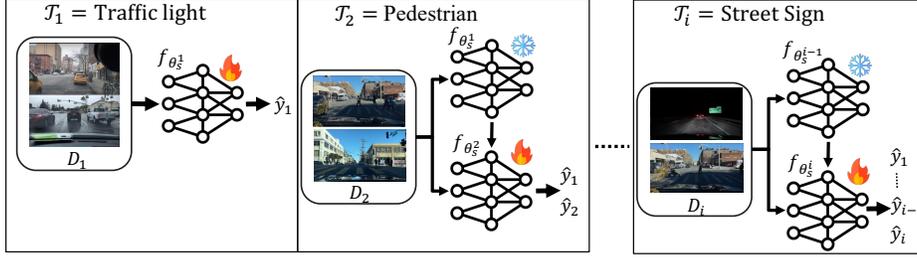


Figure 1: Overview of the LwP framework. For the first task, \mathcal{T}_1 (e.g., Traffic Light), the model is trained on data D_1 . When learning a subsequent task like \mathcal{T}_2 (Pedestrian), the model from \mathcal{T}_1 is frozen as a teacher. This process generalizes for any current task \mathcal{T}_i : the model from the previous step, $f_{\theta_s^{[i-1]}}$, acts as a teacher for the student model, $f_{\theta_s^{[i]}}$, which learns on new data D_i using a supervised loss (\mathcal{L}_{cur}), a distillation loss (\mathcal{L}_{old}), and our geometric preservation loss ($\mathcal{L}_{\text{DWDP}}$).

$$\mathcal{L}(\theta_s, \{\theta_t\}_{t=1}^T) := \sum_{t=1}^T \mathbb{E}_{(\mathbf{x}, y^t) \leftarrow \mathcal{D}_t} [\ell(y^t, \varphi(\mathbf{x}, t; \theta_s, \theta_t))], \tag{1}$$

is minimized for some loss function $\ell(\cdot, \cdot)$.

3 Learning with Preserving

This section details our proposed framework, Learning with Preserving (LwP). We begin with a high-level overview of the architecture and training process in Section 3.1. Then we explore the theoretical motivation for our core contribution, a preservation loss designed to maintain the geometric structure of the latent space, in Section 3.2. Finally, in Section 3.3, we introduce the dynamic weighting mechanism that makes this loss effective for discriminative tasks.

3.1 Overview

We introduce **Learning with Preserving (LwP)**, a framework designed to manage CMTL scenarios by maintaining the structural integrity of the model’s shared representation space across a sequence of tasks. As depicted in Figure 1, LwP uses a shared feature extractor $f_{\theta_s}(\mathbf{x})$ to produce a representation \mathbf{z} . For each task t , a separate, task-specific head $g_{\theta_t}(\mathbf{z})$ (e.g., a linear layer) generates the final prediction.

The training process at a given task t proceeds as follows. First, we duplicate the model from task $t - 1$ and add a new, randomly initialized head, g_{θ_t} , for the new task. The parameters of the previous model (both the feature extractor $f_{\theta_s}^{[t-1]}$ and all previous heads $g_{\theta_o}^{[t-1]}$ for $o < t$) are then **frozen** to serve as a stable teacher.

The current model is trained using a composite loss function, \mathcal{L}_{lwp} , which consists of three key components:

1. A standard supervised loss (\mathcal{L}_{cur}) for the current task t , which trains the new head g_{θ_t} and fine-tunes the shared extractor f_{θ_s} on new data (x_i, y_i^t) .
2. A distillation loss (\mathcal{L}_{old}) that preserves performance on previous tasks. The frozen teacher model generates “pseudolabels” \tilde{y}_o for old tasks $o < t$, and the current model is trained to match them.
3. Our novel preservation loss ($\mathcal{L}_{\text{DWDP}}$), which is the core of LwP. This loss prevents representation drift by ensuring the geometric structure of the current latent space ($z^{[t]}$) remains consistent with the structure of the frozen latent space ($z^{[t-1]}$).

Thus, the total objective is a weighted sum of these three components:

$$\mathcal{L}_{\text{lwp}} = \lambda_c \mathcal{L}_{\text{cur}} + \lambda_o \mathcal{L}_{\text{old}} + \lambda_d \mathcal{L}_{\text{DWDP}} \tag{2}$$

3.2 Preserving Implicit Knowledge

In CMTL, a model must preserve not only explicit knowledge from past tasks but also the *implicitly learned knowledge* encoded in its shared representation \mathbf{z} . We define this implicit knowledge as the geometric structure of the latent space.

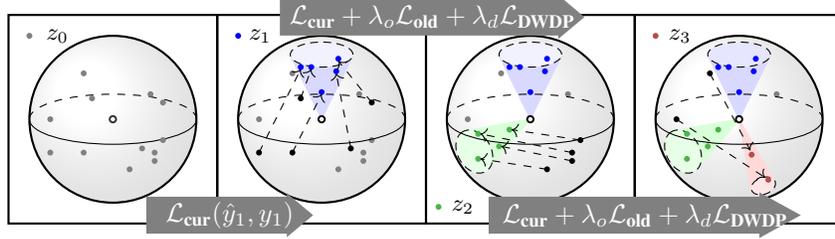


Figure 2: A visualization of the latent representation space (depicted as a sphere) as new tasks are learned sequentially. The points represent data embeddings. The LwP framework organizes representations for new tasks into distinct clusters (colored points) while its preservation loss maintains the geometric structure of prior representations.

To prevent this structure from degrading as new tasks are learned, we introduce a loss function designed to explicitly preserve it.

A direct way to maintain this geometric structure is to ensure that the pairwise distances between data points in the current model’s latent space, $Z' = f_{\theta_s^{[t]}}(X)$, remain close to those from the frozen previous model’s space, $Z = f_{\theta_s^{[t-1]}}(X)$. This leads to a family of preservation losses:

$$\mathcal{L}_{\text{pres}}(Z, Z') = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N (d(\mathbf{z}_i, \mathbf{z}_j) - d(\mathbf{z}'_i, \mathbf{z}'_j))^2, \quad (3)$$

where $d(\cdot, \cdot)$ is a distance or similarity function, which in our primary implementation is the squared Euclidean distance.

This formulation is closely related to kernel methods. Preserving pairwise distances implicitly preserves the structure defined by certain kernels. As we show in Appendix C, the change in the Gaussian kernel value is bounded by the change in the squared Euclidean distance. Therefore, minimizing the difference in distances effectively minimizes the difference between the Gram matrices $K(Z)$ and $K(Z')$, where $K_{ij} = k(\mathbf{z}_i, \mathbf{z}_j)$.

Preserving the Gram matrix ($K(Z') \approx K(Z)$) ensures that the new representation Z' is functionally equivalent to the old representation Z within the Reproducing Kernel Hilbert Space (RKHS) induced by the kernel Yamada (2013); Schölkopf, Herbrich, and Smola (2001). This means that for any function f in this universal function space, its evaluation on a new data point $f(\mathbf{z}'_i)$ can be mapped to an equivalent function f' evaluated on the old data point, $f'(\mathbf{z}_i)$. Formally, this is because preserving the kernel matrix implies an isometry T in the RKHS such that $\phi(\mathbf{z}'_i) = T(\phi(\mathbf{z}_i))$, where ϕ is the feature map. Consequently, any learning problem defined on Z has an equivalent solution on Z' , as shown in Equations 10-13.

In essence, by using a simple and efficient distance preservation loss, we ensure that the representation space remains stable in a high-dimensional feature space, preserving its capability to solve not only previously learned tasks but also potential future ones.

To formalize the alignment objective, we define the loss function $\mathcal{L}_{\text{pres}}$ as the squared Frobenius norm of the difference between the two kernel matrices:

$$\mathcal{L}_{\text{pres}}(Z, Z') = \|K(Z) - K(Z')\|_F^2 \quad (4)$$

By minimizing $\mathcal{L}_{\text{pres}}$ while keeping Z fixed, we align the images of Z and Z' under the feature map ϕ :

$$\langle \phi(\mathbf{z}_i), \phi(\mathbf{z}_j) \rangle_{\mathcal{H}} \approx \langle \phi(\mathbf{z}'_i), \phi(\mathbf{z}'_j) \rangle_{\mathcal{H}}, \quad \forall i, j. \quad (5)$$

This alignment implies that there exists an isometry $T : \mathcal{H} \rightarrow \mathcal{H}$ such that:

$$\phi(\mathbf{z}'_i) = T(\phi(\mathbf{z}_i)), \quad \forall i. \quad (6)$$

For any function $f \in \mathcal{H}$, the Riesz representation theorem states that there exists a unique element $w_f \in \mathcal{H}$ such that $f(\mathbf{z}) = \langle w_f, \phi(\mathbf{z}) \rangle_{\mathcal{H}}$. The evaluation of f at \mathbf{z}'_i becomes:

$$f(\mathbf{z}'_i) = \langle w, \phi(\mathbf{z}'_i) \rangle_{\mathcal{H}} = \langle w, T(\phi(\mathbf{z}_i)) \rangle_{\mathcal{H}}. \quad (7)$$

Because T is an isometry, its adjoint T^* is also an isometry, and we can write:

$$f(\mathbf{z}'_i) = \langle T^* w, \phi(\mathbf{z}_i) \rangle_{\mathcal{H}}. \quad (8)$$

Define $w' = T^* w$ and $f'(\mathbf{z}) = \langle w', \phi(\mathbf{z}) \rangle_{\mathcal{H}}$. Then:

$$f(\mathbf{z}'_i) = f'(\mathbf{z}_i), \quad \forall i. \quad (9)$$

Thus, Z' becomes an alternative representation that is functionally equivalent to Z in terms of any operations performed within the RKHS induced by the Gaussian kernel. Now, consider a learning problem defined on Z :

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n \ell(f(\mathbf{z}_i), y_i) + \Omega(f), \quad (10)$$

and the corresponding problem on Z' :

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n \ell(f(\mathbf{z}'_i), y_i) + \Omega(f). \quad (11)$$

Using the relationship $f(\mathbf{z}'_i) = f'(\mathbf{z}_i)$, the loss terms satisfy $\ell(f(\mathbf{z}'_i), y_i) = \ell(f'(\mathbf{z}_i), y_i)$. Since $\|f\|_{\mathcal{H}} = \|f'\|_{\mathcal{H}}$, the regularization terms are equal: $\Omega(f) = \Omega(f')$. Thus, the risk functionals for the problems on Z and Z' are equivalent when considering f and f' :

$$\frac{1}{n} \sum_{i=1}^n \ell(f(\mathbf{z}'_i), y_i) + \Omega(f) = \frac{1}{n} \sum_{i=1}^n \ell(f'(\mathbf{z}_i), y_i) + \Omega(f'). \quad (12)$$

Because the risk functionals are equivalent, the optimal solutions f^* obtained on Z' correspond to the optimal solutions f'^* on Z via the isometry T^* :

$$f^*(\mathbf{z}'_i) = f'^*(\mathbf{z}_i). \quad (13)$$

This means any model trained on Z can be transformed to a model on Z' with identical performance, and vice versa.

Through empirical observation, we have determined that maintaining the squared Euclidean distance leads to enhanced performance. This is likely because the non-exponentiated distance metric more effectively retains the global structure of the representation. Refer to Section 4.6 for the ablation result. Additionally, in Appendix C, we show that the difference in RBF kernel values is bounded by the difference in the squared L^2 norm.

3.3 Dynamic Weighting

$\mathcal{L}_{\text{pres}}$ is designed to maintain the implicitly learned knowledge of the input data in the representation space. However, in scenarios where there are distinct classes or labels, this loss can conflict with other objectives, such as separating distinct classes.

To address this issue, we introduce the Dynamically Weighted Distance Preservation (DWDP) Loss, $\mathcal{L}_{\text{DWDP}}$, which applies a dynamic mask m_{ij} to deactivate preservation for pairs with different labels, preventing conflicts with separation objectives. Unlike PODNet Douillard et al. (2020), which preserves spatial features uniformly across all pairs, RKD Park et al. (2019a), which maintains all pairwise distances regardless of class, or Asadi et al. Asadi et al. (2023), which preserves only prototype-sample distances, LwP uses dynamic per-batch masking to preserve complete intra-class pairwise structure while avoiding inter-class conflicts. The dynamic mask m_{ij} is defined as follows:

$$m_{ij} = \begin{cases} 1, & \text{if } y_i^{[t]} = y_j^{[t]}, \\ 0, & \text{otherwise,} \end{cases} \quad (14)$$

where $y^{[t]}$ represents the labels of the current task. Thus, the DWDP loss is given by:

$$\mathcal{L}_{\text{DWDP}} = \frac{1}{N^2} \sum_{i,j=1}^N m_{ij} (\Delta d_{ij})^2, \quad (15a)$$

$$\Delta d_{ij} = d(\mathbf{z}_i, \mathbf{z}_j) - d(\mathbf{z}'_i, \mathbf{z}'_j). \quad (15b)$$

Consequently, this modification alleviates the objective conflict issue at the cost of reducing the scope for preservation to intra-class sets of the current task. Our detailed pseudo-code algorithm is presented in Appendix B.

4 Evaluation

Table 2: Accuracy Comparison Across Models, Datasets, and Distribution Shifts

Method Type	Model	No Shift				Shift Scenarios (BDD100k)			
		BDD100k (3 Tasks)	CelebA (10 Tasks)	PhysiQ (3 Tasks)	FairFace (3 Tasks)	Weather Shift	Scene Shift	Time-of-Day Shift	Combined Shift
STL	-	75.123 ± 6.543	72.230 ± 7.297	87.167 ± 10.102	64.435 ± 3.660	76.760 ± 5.210	76.787 ± 5.183	76.418 ± 5.567	76.751 ± 5.180
Naive FT	-	75.572 ± 6.382	70.068 ± 9.941	81.588 ± 15.429	56.291 ± 2.221	75.281 ± 5.893	73.797 ± 8.804	76.417 ± 5.609	76.029 ± 5.895
CL	LwF	76.645 ± 6.577	64.626 ± 10.806	69.952 ± 21.090	61.034 ± 6.162	76.794 ± 5.552	77.499 ± 5.215	76.031 ± 6.062	76.938 ± 5.006
	oEWC	74.873 ± 8.375	69.666 ± 9.019	82.640 ± 12.166	63.604 ± 3.122	73.529 ± 9.222	77.224 ± 5.057	75.885 ± 5.782	74.999 ± 6.449
	ER	69.933 ± 9.112	67.598 ± 7.452	76.798 ± 16.347	63.220 ± 4.730	72.287 ± 7.359	68.533 ± 9.071	68.331 ± 8.110	67.372 ± 11.000
	SI	76.601 ± 5.277	68.735 ± 10.545	83.727 ± 11.828	63.359 ± 3.451	75.848 ± 6.193	77.893 ± 4.239	74.818 ± 5.745	74.567 ± 7.218
	GSS	75.434 ± 4.066	71.680 ± 8.468	85.741 ± 10.950	64.230 ± 3.918	74.049 ± 6.793	74.582 ± 5.554	74.332 ± 5.541	73.520 ± 7.924
	FDR	76.779 ± 5.024	69.514 ± 8.917	71.859 ± 18.687	63.709 ± 3.151	76.098 ± 6.564	73.623 ± 11.168	75.588 ± 6.654	76.200 ± 5.847
	DER	77.183 ± 5.055	70.703 ± 8.388	84.796 ± 11.168	64.114 ± 3.484	76.748 ± 5.519	76.166 ± 6.507	76.727 ± 5.252	75.943 ± 6.326
	DERPP	76.677 ± 5.751	67.693 ± 9.425	82.838 ± 13.775	63.806 ± 3.694	76.582 ± 6.071	77.409 ± 5.078	75.846 ± 6.411	68.581 ± 14.210
	DVC	72.683 ± 4.982	71.441 ± 7.640	85.100 ± 10.381	63.848 ± 3.193	72.011 ± 7.061	69.830 ± 7.905	70.019 ± 6.902	70.661 ± 7.100
OBC	76.993 ± 5.118	70.829 ± 8.267	83.999 ± 11.377	63.872 ± 3.449	72.270 ± 14.364	76.661 ± 5.988	74.835 ± 6.853	73.732 ± 8.306	
CMTL	LwP	78.299 ± 3.828	73.484 ± 8.019	88.242 ± 12.010	66.482 ± 3.138	77.937 ± 4.041	78.198 ± 3.842	76.820 ± 5.331	74.004 ± 11.268

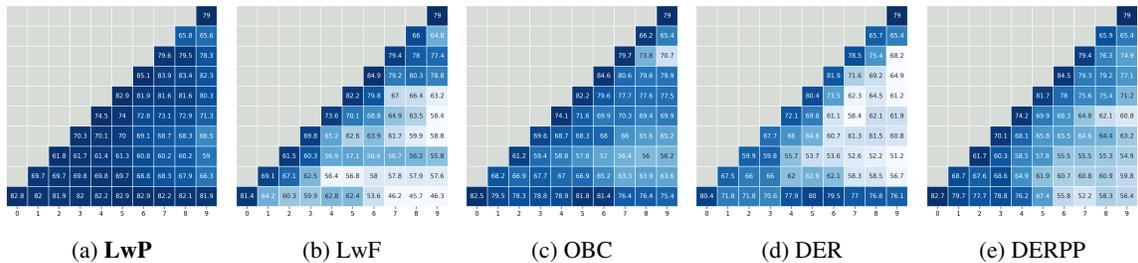


Figure 3: Selected matrices showcasing the accuracy progression for the dataset CelebA. More details in Appendix D.

4.1 Experiment Setup

We validate our approach through a series of experiments designed to demonstrate LwP’s effectiveness. First, we establish LwP’s superior performance and its enhanced robustness to distribution shifts against state-of-the-art CL baselines (Sec. 4.2, 4.3). We then investigate the underlying reason for this success, showing how LwP mitigates catastrophic forgetting by preserving the geometric structure of the representation space (Sec. 4.4). Finally, we validate our specific design choices and scalability with ablation studies and tests on larger models and higher-resolution inputs (Sec. 4.5, 4.6). The Appendix provides further details on our experimental setup, model architectures, and baselines, along with additional results such as MTL comparisons, accuracy progression over time, and evaluation on training tasks from the MTL to the CMTL setting (Appendix D).

Dataset

Dataset We utilize four datasets from two distinct modalities for our task-incremental learning experiments: BDD100K Yu et al. (2020a) (object detection in driving scenes), CelebA Liu et al. (2018) (facial attribute recognition), PhysiQ Wang and Ma (2023); Wang et al. (2025, 2024) (IMU-based exercise quality HAR), and FairFace Karkkainen and Joo (2021) (facial attribute recognition). Each dataset is structured into a series of tasks. Further details on each dataset, including the attributes, data splits, and preprocessing steps, are provided in the Appendix D.1.

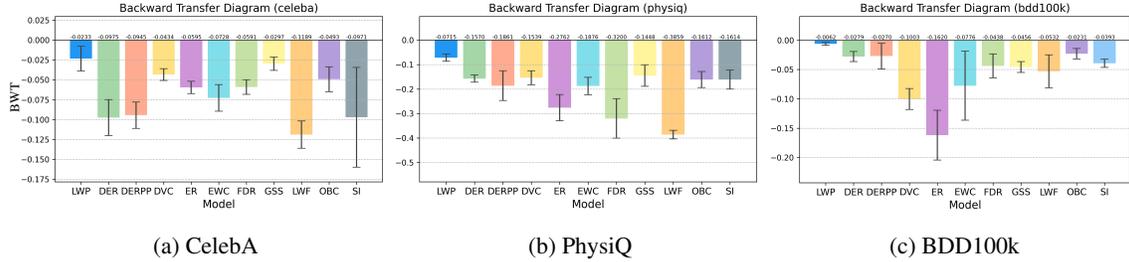


Figure 4: Selected Backward Transfer Diagrams for the Benchmark Datasets. More details in Appendix D.7

Baselines Our primary emphasis is on CL baselines since integrating many MTL methods into CMTL often requires substantial modifications to accommodate the incremental characteristics of CMTL. For CL, we compare against Online Bias Correction (OBC) Chrysakos and Moens (2023), Dual View Consistency (DVC) Gu et al. (2022), Dark Experience Replay (DER) Buzzega et al. (2020), DERPP Boschini et al. (2022), Function Distance Regularization (FDR) Benjamin, Rolnick, and Kording (2019), Experience Replay (ER) Robins (1995); Ratcliff (1990), Gradient-based Sample Selection (GSS) Aljundi et al. (2019), online Elastic Weight Consolidation (oEWC) Kirkpatrick et al. (2017), Synaptic Intelligence (SI) Zenke, Poole, and Ganguli (2017b), and Learning without Forgetting (LwF) Li and Hoiem (2017a). In addition, we compare our approach with MTL methods, which are detailed in Appendix D.4. These include the basic MTL approach of training all tasks simultaneously with different predictors Caruana (1997), as well as more advanced techniques like PCGrad Yu et al. (2020b), Impartial MTL (IMTL) Liu et al. (2021), and NashMTL Navon et al. (2022). We also include a single-task learning (STL) baseline, where each task is learned separately, and a naive fine-tune (FT), where a previously trained model is fine-tuned on the current task. For the choice of distance metric d , we test common options such as Euclidean distance and cosine similarity, as well as loss functions designed to preserve relational knowledge, such as those proposed in RKD Park et al. (2019b) and Co2L Cha, Lee, and Shin (2021).

Model Architectures For the image-based datasets (BDD100K, CelebA, and FairFace), we use a ResNet structure as the shared feature extractor f_{θ_s} . For PhysIQ, which consists of time-series data from IMU sensors, we use a 3-layer 1D-CNN model more suited to that modality. This demonstrates the versatility of our LwP framework across different data types and architectures. For all models, each task is handled by a separate linear projection layer (head) applied to the shared representation z . We evaluate additional architectures and image sizes in Section 4.5.

4.2 Comprehensive Performance in CMTL

In this experiment, we evaluate the performance of our LwP against several state-of-the-art CL methods. All methods, except for LwF and LwP, are provided with a buffer size of 512 for the CelebA and FairFace datasets, and 46 for the PhysIQ dataset, corresponding to approximately 2-3% of the training set for each dataset. Each model is trained five times using different random seeds. The standard training protocol consists of 20 epochs, with a batch size of 256 for image-based datasets and 32 for PhysIQ, coupled with early stopping. For PhysIQ, we only compare the average accuracy across the final task iteration due to the training instability caused by a smaller dataset size. Table 2 reports the average test accuracy, along with the standard deviation over five runs for each method and dataset. Fig. 3 visualizes the progression of task accuracy in task iterations (left to right). Additional results are provided in Appendix D.5

Table 2 highlights that LwP consistently achieves superior performance across all three benchmarks and is the only method to exceed the Single Task Learning (STL) baseline. This suggests that other continual learning methods likely experience significant task interference. Additional results with MTL are provided in Appendix D. Furthermore, our approach is modality-agnostic, as evidenced by LwP’s ability to generalize across different domains. This is demonstrated by the results on the PhysIQ dataset from the IMU sensor domain, which underscores LwP’s robustness against challenges unique to non-image-based tasks. While high final accuracy is a primary goal, the strength of a continual learning method is also measured by its ability to retain knowledge from past tasks. LwP’s superior accuracy, particularly its success in surpassing the strong STL baseline, suggests it is more effective at mitigating the task interference that degrades the performance of other methods. This ability to minimize forgetting is analyzed more directly using the Backward Transfer metric in Section 4.4.

4.3 Robustness to Non-Stationary Task Distributions

For real-world applications, such as autonomous driving, a model must be robust not only to test-time shifts but also to scenarios where the training data itself is non-stationary. We leveraged the BDD100k dataset’s rich annotations to design a CMTL protocol that explicitly simulates this challenge.

Instead of training on the whole, mixed dataset, we created sequential tasks defined by shifting environmental conditions. For instance, in the Weather Shift scenario, the model was trained sequentially on distinct tasks, where the data for each task came from a specific weather condition (e.g., Task 0: clear, Task 1: rainy, etc.). This protocol directly evaluates the model’s ability to handle non-stationary input distributions ($P_X^{(t)} \neq P_X^{(t+1)}$).

The final three columns of Table 2 show the average accuracy of all models trained under these non-stationary protocols. While most models experience a significant performance drop when faced with compounding distribution shifts between tasks, LwP maintains a more significant advantage over the baselines. This demonstrates that LwP learns more generalizable and fundamentally robust representations. By preserving the core latent structure, our method is less susceptible to catastrophic forgetting induced by shifting data distributions, a critical advantage for deploying intelligent systems in dynamic, real-world environments.

4.4 Mitigating Forgetting by Preserving Representation Structure

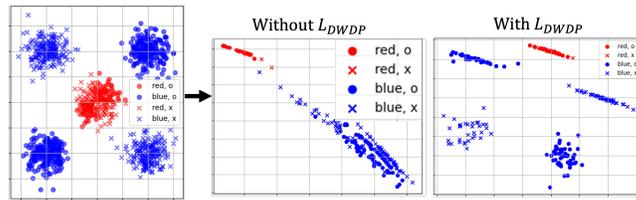


Figure 5: Impact of \mathcal{L}_{DWDP} on a constructed example. After training on a concentric circle task, the representation space without our loss (left) degrades. With \mathcal{L}_{DWDP} (right), the structure required for a subsequent XOR task is preserved.

We demonstrate that LwP effectively mitigates catastrophic forgetting in CMTL settings through both empirical evaluation and controlled demonstrations on a constructed dataset. By preserving the structure of the learned representation space, LwP enables the model to retain performance on previously learned tasks, even as it acquires new ones. This structured preservation also enhances the model’s future adaptability: as it accumulates experience across tasks, previously learned features are retained in a generalized form, facilitating more efficient learning on subsequent tasks that share latent structure. This is supported by improvements in the Backward Transfer (BWT) metric across all benchmarks, as well as by the toy example in Fig. 5, where maintaining latent geometry is critical for success on sequential tasks.

The Backward Transfer Lopez-Paz and Ranzato (2017) is a metric to evaluate the influence of learning the current task on the performance of previous tasks. A positive backward transfer value indicates that, on average, accuracies on the previous tasks have increased during the current task iteration and vice versa. It is defined as:

$$BWT = \frac{1}{T-1} \sum_{i=1}^{T-1} R_{T,i} - R_{i,i}, \quad (16)$$

where T is the index of the current task, i is an index of previous tasks ranging from 1 to $T-1$, $R_{T,i}$ is the accuracy on task i after training up to task T , and $R_{i,i}$ is the accuracy on task i after learning. As illustrated in Fig. 4, we observe that LwP outperforms all baselines in terms of BWT across all benchmarks. This result is consistent with the visualization, where LwP can maintain the accuracy of each task since its initial training.

4.5 Effect of Model Parameters and Image Sizes on Performance

Table 3 illustrates that the LwP method scales effectively with increased input resolution and model size. We find that preserving the Gaussian kernel, as shown in eq. 4, results in improved performance on larger scales, especially with respect to input resolution. In the ResNet50 benchmark utilizing a 224x224 image size, LwP notably surpasses other baselines by achieving an 85% accuracy, which is about 15% percentage points greater than the runner-up. This suggests that, as the input allows the model to create more insightful representations, LwP becomes increasingly

Table 3: Accuracy Percentage Comparison Across Models on CelebA Dataset (10 Task)

Method Type	Model	ResNet50 (32 × 32)	ResNet101 (32 × 32)	ResNet50 (224 × 224)
CL	LwF	59.277 ± 11.920	58.279 ± 11.202	60.012 ± 14.448
	oEWC	66.975 ± 10.110	67.159 ± 10.506	68.511 ± 13.352
	ER	65.335 ± 9.298	65.646 ± 8.784	65.973 ± 14.729
	SI	66.698 ± 10.030	67.456 ± 9.880	67.747 ± 13.754
	GSS	65.926 ± 13.120	65.587 ± 13.142	69.817 ± 18.771
	FDR	61.753 ± 11.943	61.720 ± 12.017	65.225 ± 15.545
	DER	62.105 ± 12.114	63.797 ± 10.774	69.859 ± 12.690
	DERPP	62.814 ± 11.071	62.957 ± 11.577	68.102 ± 13.557
	DVC	67.084 ± 10.380	65.340 ± 11.427	70.921 ± 13.823
OBC	64.220 ± 11.237	66.058 ± 10.370	69.319 ± 13.607	
CMTL	LwP	67.388 ± 11.125	69.432 ± 10.416	85.064 ± 5.388

advantageous because it can maintain these representations. We also note that the bigger models with the same input size are not performing as well as the one with ResNet18. This is because the inputs lack sufficient information to capture generalized patterns, leading to overfitting.

4.6 Ablation Study

Table 4: Ablation study on the components of the LwP framework on the PhysiQ dataset.

Method on PhysiQ	Dynamic Weighting	w/o Dynamic Weighting
LwP (Full Model)	88.2 ± 12.0	86.0 ± 12.3
LwP w/o \mathcal{L}_{old}	87.1 ± 9.44	85.4 ± 12.1
LwP (Cosine)	85.4 ± 13.1	84.1 ± 14.4
LwP (RBF)	84.5 ± 13.7	84.8 ± 14.5
IRD (Co2L)	86.4 ± 11.5	79.9 ± 17.1
RKD	85.1 ± 13.3	85.9 ± 11.9

To validate our framework, we perform two types of ablation studies. First, we evaluate the design of the proposed DWDP loss function by selectively disabling the dynamic weighting feature and comparing different distance metrics. We include Co2L Cha, Lee, and Shin (2021), RKD Park et al. (2019b) (which preserves distances across all pairs), cosine similarity, and the RBF kernel (eq. 4) as alternatives. The results in Table 4 confirm our design choices: the squared Euclidean distance combined with Dynamic Weighting yields the best performance, underscoring the importance of preserving global structure while avoiding conflicts with class-separation objectives.

Second, we analyze the model’s sensitivity to its key hyperparameters. As shown in Figure 6, LwP exhibits remarkable robustness. Its performance remains high and stable across a range of hyperparameter values. Crucially, its entire performance range surpasses that of the best-performing baseline, demonstrating a stable advantage that is not dependent on precise tuning. Detailed results of this hyperparameter sweep for both LwP and baseline models are provided in Appendix D.9.

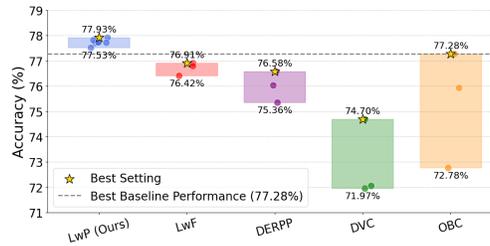


Figure 6: Accuracy on hyperparameter sensitivity analysis on the BDD100k weather shift scenario.

5 Related Work

MTL enhances generalization and computational efficiency by leveraging shared representations across related tasks Caruana (1997); Sener and Koltun (2018); Chen et al. (2022, 2023). However, optimizing multiple objectives often presents conflicting gradients. Approaches like the **Multiple Gradient Descent Algorithm (MGDA)** Sener and Koltun (2018) seek Pareto optimal solutions through convex combinations of task-specific gradients, while Gradient Surgery (PCGrad) Yu et al. (2020b) projects conflicting gradients onto the normal plane of each other to reduce interference. Navon et al. Navon et al. (2022) modeled gradient combination as a cooperative bargaining game to ensure fairness among tasks. Loss balancing is also crucial, with methods like IMTL Liu et al. (2021) incorporating both gradient and loss balancing mechanisms. CL enables sequential task learning without catastrophic forgetting Ratcliff (1990); Robins (1995). Techniques like **MER Riemer et al. (2018)** focus on maximizing knowledge transfer while minimizing interference, while HAL Chaudhry et al. (2021) anchors past knowledge to prevent representation drift. Duncker et al. Duncker et al. (2020) demonstrate that organizing recurrent network dynamics by task computation enables continual learning. Bridging MTL and CL, CMTL aims to manage performance across sequential and concurrent tasks Wu et al. (2023), using methods like MC-SGD Mirzadeh et al. (2020) to enhance CL by leveraging linear mode connectivity. **Task-free CL Aljundi, Kelchtermans, and Tuytelaars (2019)** eliminates task boundaries. More detailed discussions are available in the Appendix A.

6 Conclusion

We identified and analyzed Continual Multitask Learning (CMTL), a challenging and practical setting within continual learning where conventional methods often fail to perform effectively. Our findings show that conventional approaches underperform because they focus on preserving task-specific information while neglecting the broadly applicable, implicit features required to build a unified representation. To address this, we introduced *Learning with Preserving* (LwP) with a dynamically weighted distance preservation function. This approach maintains the structure of the representation space, preserving implicit knowledge without needing replay buffers, making it especially valuable in privacy-sensitive domains like healthcare. Our experiments across various datasets demonstrated that LwP surpasses state-of-the-art baselines and outperforms single-task models, consistently retaining accuracy and mitigating catastrophic forgetting. The results emphasize the importance of preserving implicit knowledge and the effectiveness of our loss function.

Acknowledgments

This work was supported in part by the U.S. National Science Foundation under Grants 2427711 and 2443803, the Institute of Education Sciences under Award Numbers R305C240010. The opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the sponsoring agencies.

References

- Adel, T.; Zhao, H.; and Turner, R. E. 2019. Continual learning with adaptive weights (claw). *arXiv preprint arXiv:1911.09514*.
- Aljundi, R.; Kelchtermans, K.; and Tuytelaars, T. 2019. Task-free continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11254–11263.
- Aljundi, R.; Lin, M.; Goujaud, B.; and Bengio, Y. 2019. Gradient based sample selection for online continual learning. *Advances in neural information processing systems*, 32.
- An, Z.; Moyer, D.; Oguz, I.; Johnson, T. T.; and Ma, M. 2025. ISL: Monitoring Image Segmentation Logic in Medical Imaging Analysis. In *International Conference on Runtime Verification*, 477–496. Springer.
- Asadi, N.; Davari, M.; Mudur, S.; Aljundi, R.; and Belilovsky, E. 2023. Prototype-sample relation distillation: Towards replay-free continual learning. In *International Conference on Machine Learning*, 1098–1116. PMLR.
- Benjamin, A. S.; Rolnick, D.; and Kording, K. 2019. Measuring and regularizing networks in function space. ArXiv:1805.08289 [cs, stat].
- Boschini, M.; Bonicelli, L.; Buzzega, P.; Porrello, A.; and Calderara, S. 2022. Class-Incremental Continual Learning into the eXtended DER-verse. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Buzzega, P.; Boschini, M.; Porrello, A.; Abati, D.; and CALDERARA, S. 2020. Dark Experience for General Continual Learning: a Strong, Simple Baseline. In *Advances in Neural Information Processing Systems*, volume 33, 15920–15930. Curran Associates, Inc.
- Caruana, R. 1997. Multitask learning. *Machine learning*, 28: 41–75.
- Cha, H.; Lee, J.; and Shin, J. 2021. Co²L: Contrastive Continual Learning. ArXiv:2106.14413 [cs].
- Chaudhry, A.; Gordo, A.; Dokania, P.; Torr, P.; and Lopez-Paz, D. 2021. Using hindsight to anchor past knowledge in continual learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 6993–7001.
- Chen, Z.; An, Z.; Reynolds, J.; Mullen, K.; Martini, S.; and Ma, M. 2025. LogiDebrief: A Signal-Temporal Logic based Automated Debriefing Approach with Large Language Models Integration. *arXiv preprint arXiv:2505.03985*.
- Chen, Z.; Li, I.; Zhang, H.; Preum, S.; Stankovic, J. A.; and Ma, M. 2022. CitySpec: An Intelligent Assistant System for Requirement Specification in Smart Cities. In *2022 IEEE International Conference on Smart Computing (SMARTCOMP)*, 32–39.
- Chen, Z.; Li, I.; Zhang, H.; Preum, S.; Stankovic, J. A.; and Ma, M. 2023. CitySpec with shield: A secure intelligent assistant for requirement formalization. *Pervasive and Mobile Computing*, 92: 101802.
- Chen, Z.; Sun, X.; Li, Y.; and Ma, M. 2024. Auto311: A confidence-guided automated system for non-emergency calls. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 21967–21975.
- Chrysakis, A.; and Moens, M.-F. 2023. Online bias correction for task-free continual learning. *ICLR 2023 at Open-Review*.
- Douillard, A.; Cord, M.; Ollion, C.; Robert, T.; and Valle, E. 2020. PODNet: Pooled outputs distillation for small-tasks incremental learning. In *European Conference on Computer Vision*, 86–102. Springer.
- Duncker, L.; Driscoll, L.; Shenoy, K. V.; Sahani, M.; and Sussillo, D. 2020. Organizing recurrent network dynamics by task-computation to enable continual learning. *Advances in neural information processing systems*, 33: 14387–14397.
- Freeman, B.; Hammel, N.; Phene, S.; Huang, A.; Ackermann, R.; Kanzheleva, O.; Hutson, M.; Taggart, C.; Duong, Q.; and Sayres, R. 2021. Iterative Quality Control Strategies for Expert Medical Image Labeling. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, 9(1): 60–71.
- Golatkar, A.; Achille, A.; and Soatto, S. 2020. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9304–9312.

- Gu, Y.; Yang, X.; Wei, K.; and Deng, C. 2022. Not Just Selection, but Exploration: Online Class-Incremental Continual Learning via Dual View Consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7442–7451.
- Guo, X.; Gao, L.; Liu, X.; and Yin, J. 2017. Improved Deep Embedded Clustering with Local Structure Preservation. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 1753–1759. Melbourne, Australia: International Joint Conferences on Artificial Intelligence Organization. ISBN 978-0-9992411-0-3.
- Hinton, G. 2015. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*.
- Huang, T. E.; Liu, Y.; Van Gool, L.; and Yu, F. 2023. Video task decathlon: Unifying image and video tasks in autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8647–8657.
- Jiao, Y.; Qiu, H.; Jie, Z.; Chen, S.; Chen, J.; Ma, L.; and Jiang, Y.-G. 2025. Unitoken: Harmonizing multimodal understanding and generation through unified visual encoding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 3600–3610.
- Kang, D.; Kum, D.; and Kim, S. 2024. Continual Learning for Motion Prediction Model via Meta-Representation Learning and Optimal Memory Buffer Retention Strategy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15438–15448.
- Karkkainen, K.; and Joo, J. 2021. FairFace: Face Attribute Dataset for Balanced Race, Gender, and Age for Bias Measurement and Mitigation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 1548–1558.
- Kingma, D. P.; and Ba, J. 2017. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980*.
- Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; Hassabis, D.; Clopath, C.; Kumaran, D.; and Hadsell, R. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13): 3521–3526.
- Li, Y.; Zhao, L.; Church, K.; and Elhoseiny, M. 2019. Compositional language continual learning. In *International Conference on Learning Representations*.
- Li, Z.; and Hoiem, D. 2017a. Learning without Forgetting. *ArXiv:1606.09282 [cs, stat]*.
- Li, Z.; and Hoiem, D. 2017b. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12): 2935–2947.
- Liao, W.; Xiong, H.; Wang, Q.; Mo, Y.; Li, X.; Liu, Y.; Chen, Z.; Huang, S.; and Dou, D. 2022. Muscle: Multi-task self-supervised continual learning to pre-train deep models for x-ray images of multiple body parts. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 151–161. Springer.
- Lin, Z.; Wang, Z.; and Zhang, Y. 2022. Continual Semantic Segmentation via Structure Preserving and Projected Feature Alignment. In Avidan, S.; Brostow, G.; Cissé, M.; Farinella, G. M.; and Hassner, T., eds., *Computer Vision – ECCV 2022*, volume 13689, 345–361. Cham: Springer Nature Switzerland. ISBN 978-3-031-19817-5 978-3-031-19818-2. Series Title: Lecture Notes in Computer Science.
- Liu, L.; Li, Y.; Kuang, Z.; Xue, J.-H.; Chen, Y.; Yang, W.; Liao, Q.; and Zhang, W. 2021. Towards Impartial Multi-task Learning. In *International Conference on Learning Representations*.
- Liu, Z.; Luo, P.; Wang, X.; and Tang, X. 2018. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, 15(2018): 11.
- Lopez-Paz, D.; and Ranzato, M. 2017. Gradient episodic memory for continual learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’ 17*, 6470–6479. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781510860964.
- Ma, M.; Gao, J.; Feng, L.; and Stankovic, J. 2020. STLnet: Signal temporal logic enforced multivariate recurrent neural networks. *Advances in Neural Information Processing Systems*, 33: 14604–14614.
- Mirzadeh, S. I.; Farajtabar, M.; Gorur, D.; Pascanu, R.; and Ghasemzadeh, H. 2020. Linear mode connectivity in multitask and continual learning. *arXiv preprint arXiv:2010.04495*.
- Navon, A.; Shamsian, A.; Achituve, I.; Maron, H.; Kawaguchi, K.; Chechik, G.; and Fetaya, E. 2022. Multi-task learning as a bargaining game. *arXiv preprint arXiv:2202.01017*.
- Nixon, J.; Dusenberry, M. W.; Zhang, L.; Jerfel, G.; and Tran, D. 2019. Measuring Calibration in Deep Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.
- Park, W.; Kim, D.; Lu, Y.; and Cho, M. 2019a. Relational knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3967–3976.

- Park, W.; Kim, D.; Lu, Y.; and Cho, M. 2019b. Relational Knowledge Distillation. ArXiv:1904.05068 [cs].
- Ratcliff, R. 1990. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, 97(2): 285.
- Rebuffi, S.-A.; Kolesnikov, A.; Sperl, G.; and Lampert, C. H. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2001–2010.
- Riemer, M.; Cases, I.; Ajemian, R.; Liu, M.; Rish, I.; Tu, Y.; and Tesauro, G. 2018. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910*.
- Robins, A. 1995. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2): 123–146.
- Schölkopf, B.; Herbrich, R.; and Smola, A. J. 2001. A generalized representer theorem. In *International conference on computational learning theory*, 416–426. Springer.
- Schwarz, J.; Czarnecki, W.; Luketina, J.; Grabska-Barwinska, A.; Teh, Y. W.; Pascanu, R.; and Hadsell, R. 2018. Progress & compress: A scalable framework for continual learning. In *International conference on machine learning*, 4528–4537. PMLR.
- Sener, O.; and Koltun, V. 2018. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31.
- Shaheen, K.; Hanif, M. A.; Hasan, O.; and Shafique, M. 2022. Continual Learning for Real-World Autonomous Systems: Algorithms, Challenges and Frameworks. *Journal of Intelligent & Robotic Systems*, 105(1): 9.
- Van De Ven, G. M.; Tuytelaars, T.; and Tolias, A. S. 2022. Three types of incremental learning. *Nature Machine Intelligence*, 4(12): 1185–1197.
- van der Maaten, L.; and Hinton, G. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86): 2579–2605.
- Wang, H. D.; Bae, S.; Sun, X.; Thatigotla, Y.; and Ma, M. 2025. EXACT: A Meta-Learning Framework for Precise Exercise Segmentation in Physical Therapy. In *Proceedings of the ACM/IEEE 16th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2025)*, 1–11.
- Wang, H. D.; Khan, N.; Chen, A.; Sarkar, N.; Wisniewski, P.; and Ma, M. 2024. MicroXercise: A Micro-Level Comparative and Explainable System for Remote Physical Therapy. In *2024 IEEE/ACM Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, 73–84. IEEE.
- Wang, H. D.; and Ma, M. 2023. PhysiQ: Off-site Quality Assessment of Exercise in Physical Therapy. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(4): 208:1–208:25.
- Wei, H.; Xie, R.; Cheng, H.; Feng, L.; An, B.; and Li, Y. 2022. Mitigating Neural Network Overconfidence with Logit Normalization. arXiv:2205.09310.
- Wu, Z.; Tran, H.; Pirsiavash, H.; and Kolouri, S. 2023. Is multi-task learning an upper bound for continual learning? In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE.
- Yamada, A. 2013. Inequalities for Gram matrices and their applications to reproducing kernel Hilbert spaces. *Project Euclid*.
- Yu, F.; Chen, H.; Wang, X.; Xian, W.; Chen, Y.; Liu, F.; Madhavan, V.; and Darrell, T. 2020a. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2636–2645.
- Yu, T.; Kumar, S.; Gupta, A.; Levine, S.; Hausman, K.; and Finn, C. 2020b. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33: 5824–5836.
- Zenke, F.; Poole, B.; and Ganguli, S. 2017a. Continual Learning Through Synaptic Intelligence. ArXiv:1703.04200 [cs, q-bio, stat].
- Zenke, F.; Poole, B.; and Ganguli, S. 2017b. Continual learning through synaptic intelligence. In *International conference on machine learning*, 3987–3995. PMLR.

A Related Works

A.1 Multitask Learning

Multitask learning (MTL) has been extensively explored for its ability to leverage shared representations across multiple related tasks, thereby enhancing generalization and computational efficiency. In MTL, models are trained on multiple tasks simultaneously, with the assumption that learning tasks together allows the model to capture commonalities and differences among tasks, leading to better performance than training each task separately.

A central challenge in MTL is the optimization of multiple objectives, which often present conflicting gradients that can impede the convergence and performance of the model. To address this, various gradient balancing approaches have been proposed. Sener and Koltun (2018) introduced the Multiple Gradient Descent Algorithm (MGDA), which seeks Pareto optimal solutions by finding a convex combination of task-specific gradients. Building on this, Yu et al. (2020b) proposed Gradient Surgery (PCGrad), which directly modifies conflicting gradients by projecting them onto the normal plane of each other to reduce negative interference. More recently, Navon et al. (2022) approached MTL from a game-theoretic perspective, modeling the gradient combination step as a cooperative bargaining game and employing the Nash Bargaining Solution to ensure proportional fairness among tasks.

In addition to gradient balancing, loss balancing is crucial for achieving stable and unbiased learning in MTL. Liu et al. (2021) introduced IMTL, which incorporates both gradient and loss balancing mechanisms. Their method, IMTL-G, ensures unbiased updates to task-shared parameters by finding the geometric angle bisector of task gradients. At the same time, IMTL-L automatically learns loss weighting parameters to harmonize the scales of different task losses.

While these MTL methods have advanced the ability to learn multiple tasks simultaneously, they typically assume that all task data is available at training time and can be processed jointly. This assumption does not hold in scenarios where tasks and their associated data arrive sequentially, as is the case in our defined problem, Continual Multitask Learning (CMTL). In such cases, models must learn new tasks without access to all previous data, and ideally, they should utilize new tasks to enhance their performance on prior tasks.

Our work differs from traditional MTL approaches in that it addresses the sequential arrival of tasks and data, where tasks are learned iteratively rather than simultaneously. Unlike MTL methods that focus on balancing gradients and losses across tasks trained together, our approach must handle the challenge of incorporating new tasks without re-training on the data of previous tasks. Furthermore, we introduce mechanisms to utilize new task data to enhance the model’s generalizability on earlier tasks, which is not typically considered in standard multi-task learning (MTL) frameworks.

Rather than creating entirely new frameworks, some research focuses on improving existing models. For instance, Park et al. (2019b) proposes a method for transferring knowledge via the loss function, known as relational knowledge distillation, which enhances the latent space. Similarly, Guo et al. (2017) and Lin, Wang, and Zhang (2022) concentrate on structurally maintaining the embedding space, ensuring its robustness over time.

A.2 Continual Learning (CL)

CL aims to enable models to learn sequentially from a stream of tasks without forgetting previously acquired knowledge, addressing the challenge of catastrophic forgetting Ratcliff (1990); Robins (1995). Various methods have been developed to tackle this problem, broadly categorized into *rehearsal-based methods*, *knowledge distillation*, and *regularization-based techniques*.

Rehearsal-based methods Early works such as Ratcliff (1990); Robins (1995) introduced Experience Replay (ER), where old data samples are mixed with current ones during training. Building upon this concept, Robins (1995) explored pseudorehearsal techniques. More recent methods like Meta-Experience Replay (MER) Riemer et al. (2018) reformulate ER within a meta-learning framework, aiming to enhance knowledge transfer between past and present tasks while reducing interference. Gradient-based Sample Selection (GSS) Aljundi et al. (2019) modifies ER by selecting optimal examples for storage in the memory buffer, improving retention of past knowledge. Lastly, Aljundi et al. Aljundi, Kelchtermans, and Tuytelaars (2019) introduce task-free continual learning, eliminating the need for task boundaries and enabling more flexible adaptation to new tasks without explicit task identifiers.

These methods, while effective in certain scenarios, rely heavily on storing and replaying data from previous tasks, which may not be feasible due to privacy concerns or memory constraints. In contrast, our approach does not require storing raw data from previous tasks. Instead, we utilize pseudolabels generated by the frozen previous model and introduce a novel regularization term to preserve the structure of shared representations. This enables the model to retain and improve upon prior knowledge without explicit rehearsal.

Knowledge Distillation Methods leveraging Knowledge Distillation Hinton (2015) address the issue of forgetting by using a previous iteration of the model as a teacher. Learning Without Forgetting (LwF) Li and Hoiem (2017b) generates a softened version of the model’s current outputs on new data at the onset of each task, minimizing output drift throughout training. iCaRL Rebuffi et al. (2017) combines distillation with replay techniques, using a memory buffer to train a nearest-mean-of-exemplars classifier while applying a self-distillation loss to preserve learned representations across tasks. Moreover, Li et al. Li et al. (2019) proposed a continual learning method tailored for sequence-to-sequence tasks, leveraging compositionality to enable knowledge transfer and prevent catastrophic forgetting. Their approach extends traditional label prediction continual learning methods to handle more complex tasks like machine translation and instruction learning.

While these methods use knowledge distillation to maintain performance on old tasks, they typically focus on preserving output logits or feature representations without considering the underlying relational structure between data points. Our method extends this idea by not only preserving the output predictions via pseudolabels but also maintaining the pairwise relationships in the representation space through our Dynamically Weighted Distance Preservation (DWDP) loss. This helps in better retaining the learned structure and prevents the model from drifting away from previously acquired knowledge.

Regularization-based techniques These methods modify the loss function to include a penalty that restricts changes to the model’s parameters. Examples include Elastic Weight Consolidation (EWC) Kirkpatrick et al. (2017), its online variant (oEWC) Schwarz et al. (2018), and Synaptic Intelligence (SI) Zenke, Poole, and Ganguli (2017b). Complementary to parameter-based regularization, Duncker et al. Duncker et al. (2020) demonstrate how organizing network dynamics by task computation can enable continual learning in recurrent networks. Moreover, Adel et al. Adel, Zhao, and Turner (2019) introduced Continual Learning with Adaptive Weights (CLAW), which employs a probabilistic modeling approach to adaptively identify which parts of the network should be shared across tasks in a data-driven manner. This method balances between modeling each task separately to prevent catastrophic forgetting and sharing components to allow transfer learning and reduce model size.

Our approach differs from these methods as we do not rely on parameter regularization or expanding architectures. Instead, we focus on preserving the learned representations and their relational structure between tasks through the DWDP loss, which provides a more scalable solution without incurring additional memory overhead.

A.3 Continual and Multitask Learning

Our work distinguishes itself from existing approaches in CMTL by introducing a new problem domain where new tasks and their associated datasets arrive incrementally. In this setting, the model is not only required to adapt to new tasks but also to utilize these new datasets to enhance its performance on previous tasks. Specifically, when new data for additional tasks becomes available, it is used to further train the existing model. This training process enables the model to reinforce and improve its understanding of prior tasks, effectively allowing it to remember and perform better on both past and current tasks.

Building upon the extensive research in multitask learning (MTL) Caruana (1997); Sener and Koltun (2018); Yu et al. (2020b); Navon et al. (2022); Li and Hoiem (2017b) and continual learning (CL) Ratcliff (1990); Robins (1995); Riemer et al. (2018); Aljundi et al. (2019); Chaudhry et al. (2021); Li and Hoiem (2017b), the emerging field of continual multitask learning seeks to bridge the two paradigms to effectively manage performance across sequential and concurrent tasks Wu et al. (2023); Chen et al. (2025, 2024).

One of the most related works to ours, Mirzadeh et al. Mirzadeh et al. (2020), focus on the linear mode connectivity between solutions obtained through sequential and simultaneous training. While they demonstrate that a linear path of low error exists for more than twenty tasks and introduce algorithms like Mode Connectivity SGD (MC-SGD) to enhance continual learning, their work does not address the use of *new tasks* to improve performance on previous ones, particularly using a similar setup to traditional continual learning, which means their works fit more on the realm of CL.

Similarly, Liao et al. Liao et al. (2022) propose MUSCLE, a multitask self-supervised continual learning framework designed to pre-train deep models on diverse X-ray datasets. This work, similar to ours, operates in the domain of medical imaging to process classification and segmentation in different body areas. However, their work differs from ours because their focus is on pre-training the model on different tasks for better generalization, which they refer to as “multitask continual learning.” We specifically differentiate our CMTL approach from theirs in that our tasks are seen *iteratively*; we do not have access to all tasks at the same time, and the tasks themselves could be orthogonal to previously seen tasks.

In summary, our approach introduces a novel aspect to CMTL by leveraging new tasks and their data not only to learn the new tasks but also to generalize on prior tasks, all within an iterative framework where tasks arrive sequentially and are potentially unrelated. This sets our work apart from existing CMTL methods, which typically do not utilize new tasks to enhance previous ones in this manner.

B Learning with Preserving Algorithm Overview

In this section, we present the pseudocode for our algorithm presented in Section 3.

Algorithm 1 Learning with Preserving (LwP)

```

1: Input: Sequence of tasks  $\{D_t\}_{t=1}^T$ , hyperparameters  $\lambda_c, \lambda_o, \lambda_d$ 
2: Output: Final model parameters  $\theta^{[T]}$ 
3: Initialize model parameters  $\theta^{[0]}$ 
4: for  $t = 1$  to  $T$  do
5:   Set  $\theta^{[t]} \leftarrow \theta^{[t-1]}$ 
6:   Add new task-specific head  $g_{\theta_t}$  for task  $t$ 
7:   Freeze all parameters from the previous step ( $\theta^{[t-1]}$ )
8:   for each mini-batch  $\{(\mathbf{x}_i, y_i^t)\}_{i=1}^N$  in  $D_t$  do
9:     // Compute representations and outputs
10:     $\mathbf{z}_i^{[t]} \leftarrow f_{\theta_s^{[t]}}(\mathbf{x}_i)$  // current representation
11:     $\hat{y}_i^t \leftarrow g_{\theta_t^{[t]}}(\mathbf{z}_i^{[t]})$  // current task output
12:     $\mathbf{z}_i^{[t-1]} \leftarrow f_{\theta_s^{[t-1]}}(\mathbf{x}_i)$  // frozen representation
13:    // Compute outputs for previous tasks
14:    for  $o = 1$  to  $t - 1$  do
15:       $\hat{y}_i^o \leftarrow g_{\theta_o^{[t]}}(\mathbf{z}_i^{[t]})$  // current old task output
16:       $\tilde{y}_i^o \leftarrow g_{\theta_o^{[t-1]}}(\mathbf{z}_i^{[t-1]})$  // pseudolabel
17:    end for
18:    // Compute losses
19:     $\mathcal{L}_{\text{cur}} \leftarrow \mathcal{L}_{\text{cur}}(y_i^t, \hat{y}_i^t)$ 
20:     $\mathcal{L}_{\text{old}} \leftarrow \sum_{o=1}^{t-1} \mathcal{L}_{\text{old}}(\tilde{y}_i^o, \hat{y}_i^o)$ 
21:     $\mathcal{L}_{\text{DWDP}} \leftarrow \mathcal{L}_{\text{DWDP}}(\mathbf{z}_i^{[t]}, \mathbf{z}_i^{[t-1]}, y_t)$ 
22:    // Combine all losses into LwP loss

$$\mathcal{L}_{\text{lwp}} = \lambda_c \mathcal{L}_{\text{cur}} + \lambda_o \mathcal{L}_{\text{old}} + \lambda_d \mathcal{L}_{\text{DWDP}}.$$

23:    // Update parameters
24:     $\theta^{[t]} \leftarrow \theta^{[t]} - \eta \nabla_{\theta^{[t]}} \mathcal{L}_{\text{lwp}}$ 
25:  end for
26: end for

```

C On Using Euclidean Distance 3.2

Here, we show that preserving the squared Euclidean distances between the data points in Z and Z' is sufficient to achieve the same effect.

Squared Euclidean Distance Preservation We define the squared Euclidean distance between two points z_i and z_j as:

$$D_{ij}(Z) = \|z_i - z_j\|^2. \quad (17)$$

Similarly, we compute $D_{ij}(Z')$ for Z' .

Our goal is to minimize the difference between the squared distances in Z and Z' , which we formalize with the following loss function:

$$\mathcal{L}_{\text{dist}}(Z, Z') = \sum_{i=1}^n \sum_{j=1}^n (\|z_i - z_j\|^2 - \|z'_i - z'_j\|^2)^2. \quad (18)$$

Minimizing $\mathcal{L}_{\text{dist}}$ with respect to Z' encourages the squared distances between all pairs of points in Z' to match those in Z :

$$\|z'_i - z'_j\|^2 \approx \|z_i - z_j\|^2, \quad \forall i, j. \quad (19)$$

Since the exponential function is Lipschitz continuous on compact subsets, small changes in the squared distance result in small changes in the kernel value. Specifically, if the squared distances are preserved within a small error $\epsilon > 0$:

$$\left| \|z_i - z_j\|^2 - \|z'_i - z'_j\|^2 \right| < \epsilon, \quad (20)$$

then the difference in the kernel values can be bounded:

$$\begin{aligned} & \left| \|z_i - z_j\|^2 - \|z'_i - z'_j\|^2 \right| < \epsilon \\ \implies & \left| k(z_i, z_j) - k(z'_i, z'_j) \right| \\ & = \left| \exp\left(-\frac{\|z_i - z_j\|^2}{2\sigma^2}\right) - \exp\left(-\frac{\|z'_i - z'_j\|^2}{2\sigma^2}\right) \right| \\ & \leq \frac{1}{2\sigma^2} \exp\left(-\frac{\min(\|z_i - z_j\|^2, \|z'_i - z'_j\|^2)}{2\sigma^2}\right) \\ & \quad \times \left| \|z_i - z_j\|^2 - \|z'_i - z'_j\|^2 \right| \\ & \leq \frac{1}{2\sigma^2} \epsilon \\ & \leq L_k \epsilon. \end{aligned} \quad (21)$$

where L_k is a Lipschitz constant dependent on σ .

Therefore, preserving the squared Euclidean distances between Z and Z' implies that the Gaussian kernel matrices $K(Z)$ and $K(Z')$ are approximately equal:

$$k(z_i, z_j) \approx k(z'_i, z'_j), \quad \forall i, j. \quad (22)$$

D Additional Details on Experiments

D.1 Experiments Setup Cont.

Datasets We utilize four datasets from two distinct modalities, each structured for task-incremental learning. In this setting, each task is only exposed to a subset of training samples:

The BDD100K dataset Yu et al. (2020a) is a large-scale, diverse driving dataset containing over 100,000 images. For our purposes, we focus on three object detection tasks: "traffic sign", "person", and "truck." These classes were selected to minimize imbalance. The input images are resized to 224x224. A key feature of this dataset is its rich annotation of environmental conditions, which we leverage to validate our model's robustness to distribution shifts across different weather types (e.g., clear, rainy, snowy) and scene types (e.g., city street, highway), as well as their combinations.

The CelebA dataset Liu et al. (2018), consisting of 200,000 images with 40 facial attributes. For our work, we focus on 10 of the most balanced attributes. The train dataset is equally subdivided for each task, leading to 20,000 images per task. For simplicity, input images are resized to 32x32.

The PhysiQ dataset Wang and Ma (2023), which contains approximately 4,500 samples collected using inertial measurement units (IMUs) to capture the quality of physical exercises. The data is collected on accelerometer and gyroscope modality at a 50 Hz sampling rate for 31 participants with three attributes. Each task corresponds to one of these attributes with around 1,500 samples.

The Fairface dataset Karkkainen and Joo (2021), which includes 100,000 images with three attributes. Following the same subdivision procedure, the dataset results in approximately 33,333 images with a resolution of 128x128 per task. Not only do the tasks differ from those of CelebA, but also the images are not resized in order to show that our approach is scalable.

D.2 Hyperparameters

In the following section, we provide an extensive description of the hyperparameters utilized during the training phase. Across all datasets and models, the Adam optimizer Kingma and Ba (2017) was employed universally. For the CelebA and FairFace datasets, a consistent learning rate of 0.0001 was maintained, coupled with a batch size configuration of 256. In contrast, for the PhysiQ dataset, a higher learning rate of 0.01 was utilized alongside a smaller batch size of 32. Furthermore, we adhered to fixed model-specific hyperparameters for all datasets and models to ensure uniformity and consistency, including the LwP parameters. In the case of LwP, the parameters set as follows: λ_n as a value of 1, λ_o as a value of 1, and λ_d with a value of 0.01. Additionally, the 10 tasks used for CelebA are wearing lipsticks, smiling, mouth slightly open, high cheekbones, attractive, heavy makeup, male, young, wavy hair, and straight hair. PhysiQ dataset includes three attributes assessing exercise quality: stability, range of motion, and exercise variation.

Details of all models and their hyperparameter selection have been documented in the codebase at [ANONYMOUS LINK].

D.3 Representation Space Visualization via t-SNE

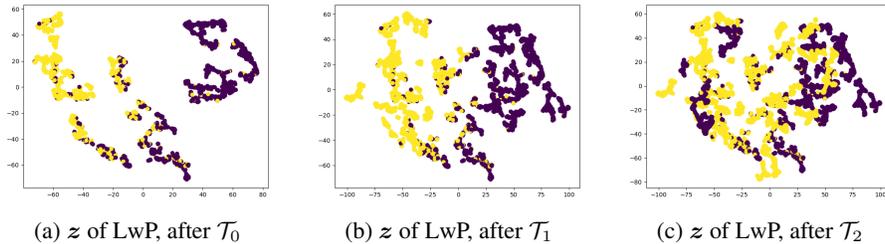


Figure 7: Representation space progression over task iteration. Colors indicate different label values.

Fig. 7 shows the 2D visualization of the representation z for each model trained on the PhysiQ dataset. It was constructed using the dimensionality reduction algorithm t-SNE van der Maaten and Hinton (2008) on the PhysiQ test dataset at the end of each task iteration. x and y axes represent the two new dimensions created by the algorithm to project the high-dimensional data onto a 2D plane. Note these dimensions do not have an intrinsic meaning and rather constructed to reflect the relative distances between data points in the high-dimensional space. As demonstrated, the z produced by LwP maintains coherent cluster formations as it progressively learns new tasks without introducing considerable distortions when compared to the baseline model. This behavior is comparable to the example provided with the toy dataset depicted in Fig. 5.

D.4 Comparison with MTL methods

All MTL approaches utilize the same model architecture as LwP. Despite being supplied with all labels for every input data point, the amount of training samples for MTL models matches that seen by CL models per task iteration. Aligning with earlier studies, MTL approaches frequently represent the upper bound for all CL models. An interesting discovery is that all MTL models deliver nearly identical performance on the benchmark.

Table 5: Comparison of Accuracy across Different Models and Datasets

Method Type	Model	CelebA	PhysiQ	FairFace
STL	-	72.230 ± 7.297	87.167 ± 10.102	64.435 ± 3.660
MTL	MTL	76.526 ± 7.616	93.536 ± 5.739	71.418 ± 4.169
	PCGrad	75.506 ± 8.146	91.910 ± 8.491	70.061 ± 4.892
	IMTL	76.280 ± 7.248	92.661 ± 6.617	71.399 ± 3.887
	NashMTL	75.506 ± 8.146	91.518 ± 7.118	71.607 ± 3.577
CMTL	LwP	73.484 ± 8.019	88.242 ± 12.010	68.545 ± 4.454

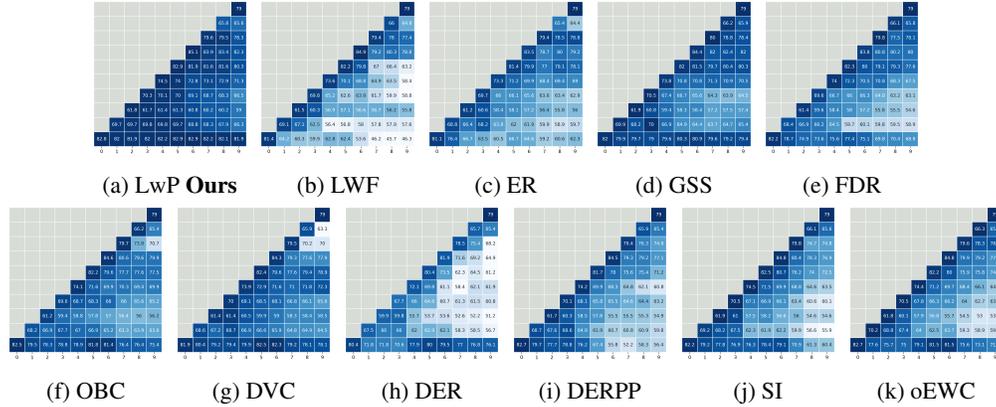


Figure 8: Matrices showcasing the accuracy progression for various models for the dataset CelebA. Each column corresponds to an iteration of the task, arranged sequentially from left to right. We generate confusion matrices normalized across all tasks for all models to ensure consistency.

D.5 Accuracy Progression for Each Task Iteration

Figures 9, 10, and 11 illustrate that the application of LwP reduces the issue of catastrophic forgetting in the PhysiQ and FairFace datasets as well. This effect is particularly pronounced when applied to datasets with a large number of samples, such as Fairface, CelebA, and BDD100K, in comparison to smaller datasets such as PhysiQ. These observations imply that LwP is a scalable and effective solution to mitigate catastrophic forgetting in continual multitask learning models. We further investigate the effect of the number of training samples on performance in D.6.

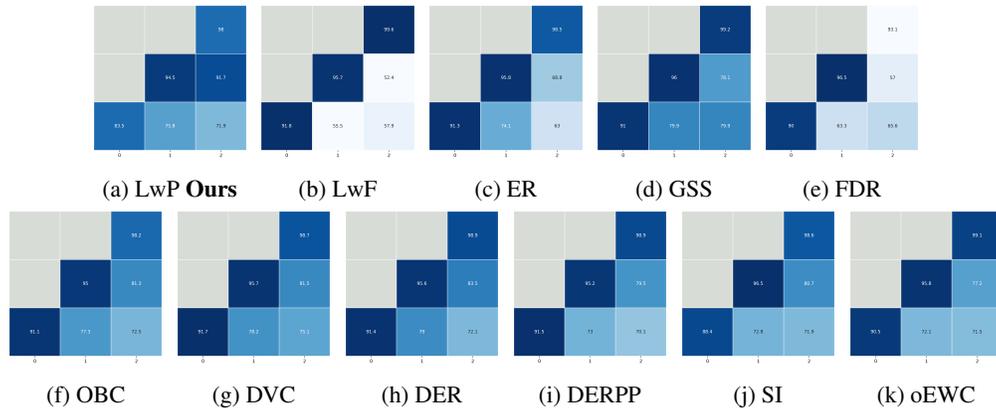


Figure 9: Confusion matrices for different models on the PhysiQ dataset

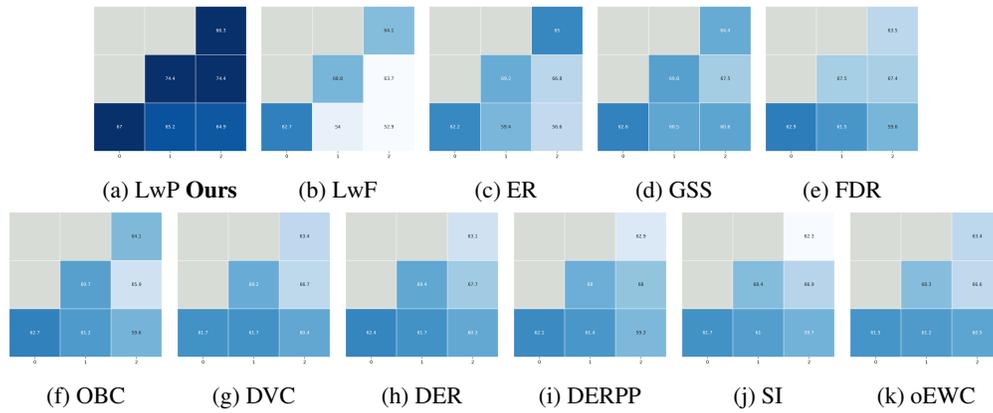


Figure 10: Confusion matrices for different models on the FairFace dataset

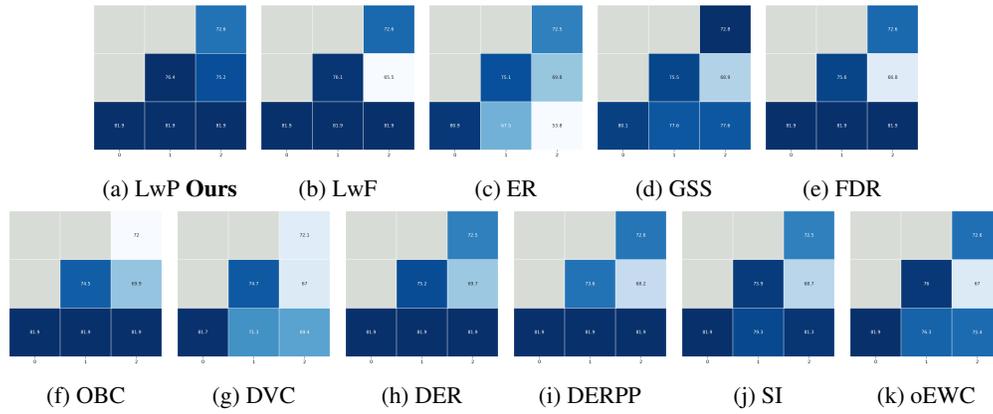


Figure 11: Confusion matrices for different models on the BDD100K dataset

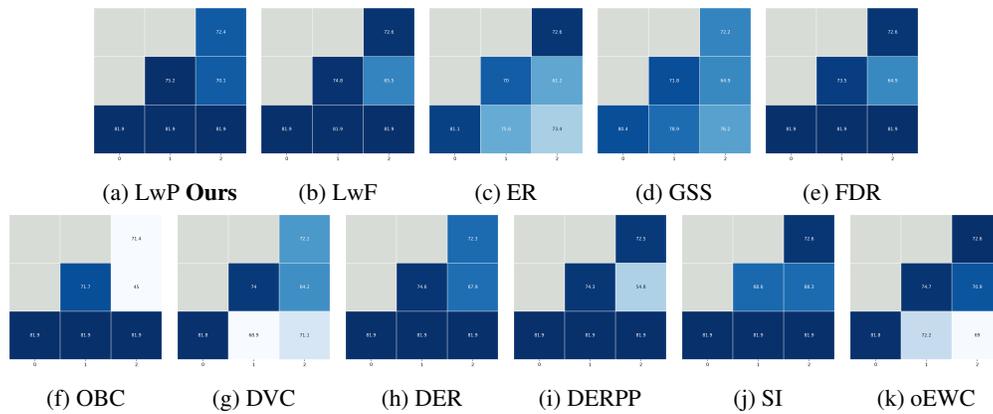


Figure 12: Confusion matrices for different models on the BDD100K dataset under a **weather-based distribution shift**.

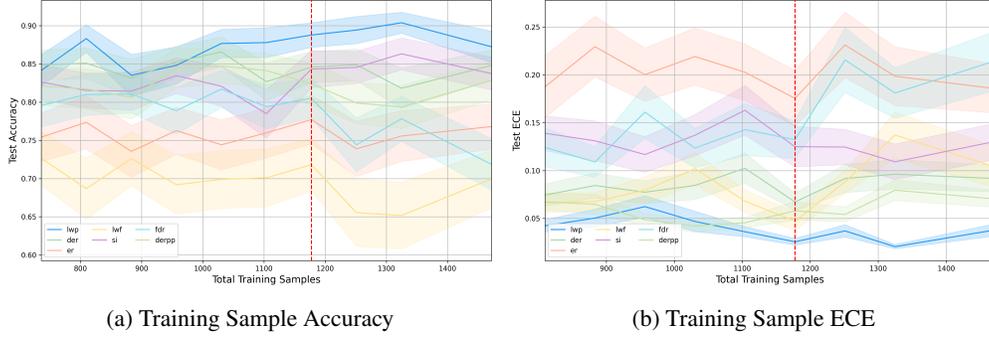


Figure 13: Training Sample Accuracy and ECE. Noted, the red line represents the buffer size is greater or equal than the batch size, since the buffer size of replay buffer methods is determined by a percentage of the batch size.

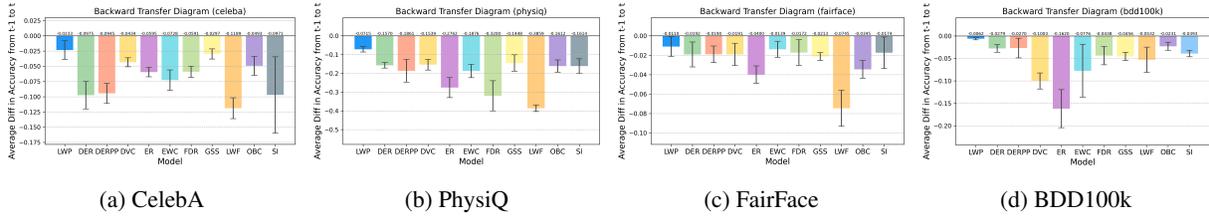


Figure 14: The Backward Transfer Diagrams for All Four Benchmark Datasets

D.6 Influence of Training Sample

We include experiment results on the influence of number of training samples to the performance, as shown in Fig. 13a. It shows that our approach outperforms others from 1000 labels and onward, when trained and tested on the PhysiQ dataset.

Fig. 13b illustrates the Expected Calibration Error (ECE) Nixon et al. (2019) for each model in relation to the number of training samples. The ECE quantifies how much confidence a model deviates from the actual output distribution, with a lower ECE indicating more accurate confidence assignments for a given classification target. This metric is particularly crucial in safety-critical settings, where the model must provide reliable confidence output. As neural network models frequently demonstrate overconfidence Wei et al. (2022), monitoring ECE becomes essential. The figure reveals that LwP not only maintains the lowest variance across different seeds but also achieves the lowest ECE value when the training sample size exceeds roughly 1000.

D.7 Performance Improvement per Iteration

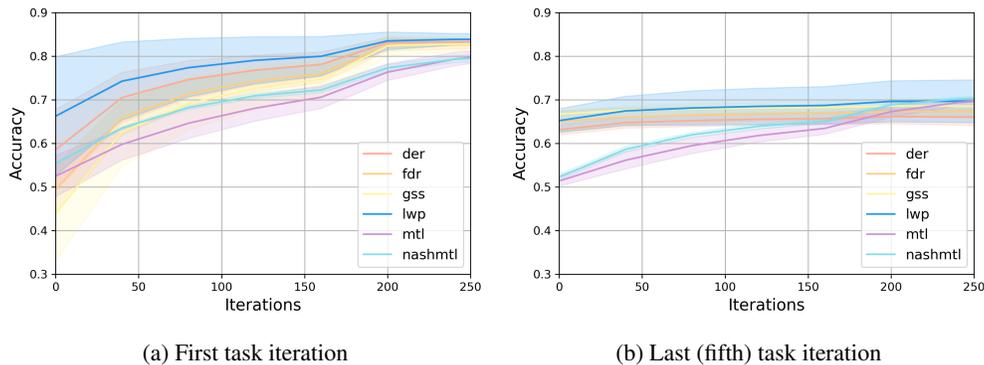


Figure 15: Average accuracy progression per iteration for CelebA with 5 tasks

We show that LwP demonstrates a faster improvement per iteration given the same batch size as other CL and MTL models. Here, we plot the evolution of average accuracy across all tasks seen on the test set over training iteration for the top performing CL and MTL baselines along with LwP. We used the CelebA dataset with 5 task splits. To make a fair comparison with CL models, MTL models were trained on the amount of train data that CL models saw in each iteration with access to all 5 tasks. The accuracies of MTL models are calculated up to what CL models have learned so far.

Fig. 15a shows how quickly each CL and MTL model learns the first task. This can be understood as the speed at which the models acquire knowledge when they have no prior information to “recall”. As shown, LwP learns consistently faster per iteration compared to CL and MTL baselines. As MTL models simultaneously learn multiple labels, their convergence per iteration is generally slower compared to CL models in this setting.

Conversely, Fig. 15b illustrates a case where MTL models are trained from the beginning with labels available for all t tasks, whereas CL models, having been pretrained on $t - 1$ tasks, must now incrementally learn the t^{th} task while maintaining performance on old tasks. This configuration is crucial in real-world scenarios where the cost of labeling data typically exceeds that of data collection, prompting the decision to gather more partially labeled data rather than re-labeling existing data. Analogous to the prior scenario, the progression of test accuracy over iterations demonstrates that LwP consistently exceeds other CL models and exhibits performance that is competitive with MTL models, which are considered the upper bound for continual learning. This highlights the comparative benefit of LwP when users face the choice between relabeling existing data and obtaining new data with different labels.

D.8 Training from MTL to CMTL

We initially train the model on the first five tasks using a MTL setting, employing ResNet18 as the encoder with input images of size $64 \times 64 \times 3$. After completing the MTL phase, we extract the encoder and freeze its weights. This frozen encoder is then used to train classifiers for the first five tasks in a CMTL setting with various models. Subsequently, we train the entire model for the last five tasks under the same CMTL setting, utilizing the same frozen encoder on the remaining tasks. This approach enables us to evaluate the effectiveness of our method in transitioning from MTL to CMTL setting while maintaining consistent performance across all tasks.

Our method significantly outperforms other models in terms of accuracy on the CelebA dataset. Specifically, as shown in Table 6, LwP achieves an average accuracy of 83.652%, surpassing all other CL methods tested. The closest competitors, oEWC and SI, achieve accuracies of 82.250% and 82.194%, respectively. This demonstrates the effectiveness of our approach in leveraging an MTL pre-trained encoder for subsequent CMTL tasks.

The superior performance of LwP suggests that initializing the encoder with MTL on the first five tasks provides a robust foundation for learning new tasks. Our method effectively mitigates catastrophic forgetting by preserving essential features discovered during the MTL phase while adapting to new tasks, given that the continual tasks are shorter now. This balance between stability and plasticity still allows LwP to maintain high accuracy in the continual learning tasks.

Table 6: Accuracy Percentage Comparison Across Models on CelebA Dataset, Trained on MTL on the first five tasks, then CL on the last five tasks

Method Type	Model	ResNet18 (64×64)
CL	LwF	74.057 ± 11.364
	oEWC	82.250 ± 6.362
	ER	77.245 ± 8.434
	SI	82.194 ± 6.460
	GSS	80.563 ± 8.239
	FDR	81.271 ± 7.738
	DER	81.010 ± 8.674
	DERPP	78.177 ± 9.532
	DVC	81.387 ± 7.821
	OBC	80.516 ± 8.446
CMTL	LwP	83.652 ± 7.069

Moreover, the lower standard deviation in LwP’s performance indicates consistent results across different runs, highlighting the reliability of our approach. The results confirm that combining MTL pre-training with our proposed CL strategy enhances the model’s ability to generalize and adapt to new tasks without compromising performance on previously learned tasks.

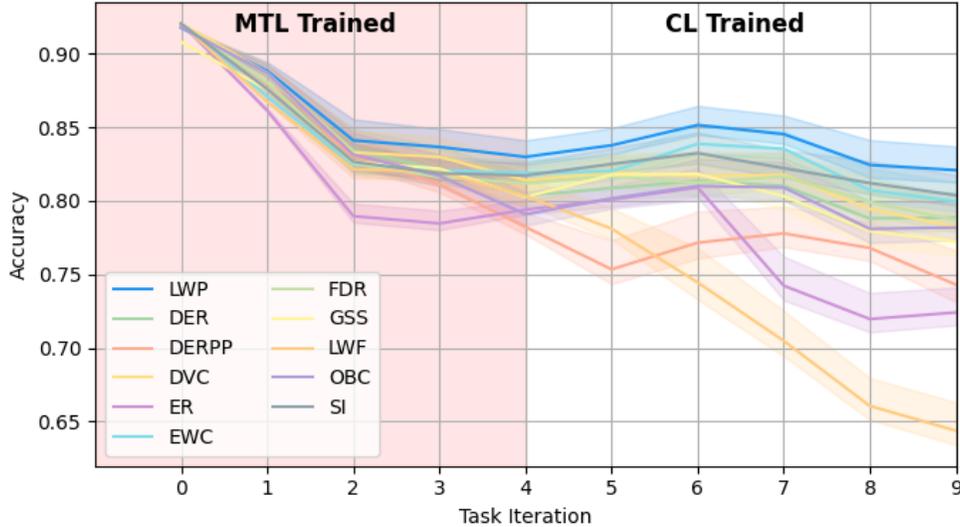


Figure 16: Train the first five tasks in the MTL method (meaning all tasks are trained simultaneously using the basic MTL model), then apply the encoder to CL method models. We used the encoder to further train in the CMTL setting with different models with ResNet 18, with an image size of 64.

Similarly, in Figure 16, we averaged the results across task iterations to evaluate performance over time. Our method, LwP, demonstrates minimal accuracy loss when training on new tasks, highlighting its performance against forgetting. The standard deviation—represented as 20% of the total for visualization purposes—remains low, indicating consistent performance. Although there is a slight increase in standard deviation during later tasks, suggesting a potential drop in accuracy due to forgetting, LwP still preserves knowledge at a superior level compared to other baselines. Even with the first five tasks trained in a multitask setting, our method maintains the best overall accuracy, outperforming other models in preserving learned information.

D.9 Hyperparameter testing

We test the sensitivity of LwP with respect to the hyperparameter values. The experiments are conducted using BDD100k under the weather shift scenario. The results show that different hyperparameter settings do not significantly affect the performance of LwP. Additionally, we provide the hyperparameter tuning results for the two most comparable baselines.

Table 7: Hyperparameter sensitivity analysis on BDD100k (weather shift scenario). The best-performing setting for each method is shown in bold.

Method	Parameter Tested	Accuracy
LwP (Ours)	$\lambda_{dwdp} = 0.01$ (fixed $\lambda_{old} = 1$)	77.71±4.30
	$\lambda_{dwdp} = 0.05$ (fixed $\lambda_{old} = 1$)	77.73±4.20
	$\lambda_{dwdp} = 0.1$ (fixed $\lambda_{old} = 1$)	77.83±4.09
	$\lambda_{dwdp} = 1$ (fixed $\lambda_{old} = 1$)	77.93±4.04
	$\lambda_{old} = 0.1$ (fixed $\lambda_{dwdp} = 0.05$)	77.72±4.23
	$\lambda_{old} = 0.5$ (fixed $\lambda_{dwdp} = 0.05$)	77.75±4.23
	$\lambda_{old} = 1$ (fixed $\lambda_{dwdp} = 0.05$)	77.53±4.48
LwF	$\lambda = 0.01$	76.42±6.08
	$\lambda = 0.1$	76.91±5.32
	$\lambda = 1$	76.79±5.55
DERPP	$\beta = 0.01$ ($\alpha = 0.1$)	76.04±6.84
	$\beta = 0.1$ ($\alpha = 0.1$)	76.58±6.07
	$\beta = 1$ ($\alpha = 0.1$)	75.36±9.67
DVC	$\lambda = 0.01$	71.97±5.23
	$\lambda = 0.1$	72.07±7.59
	$\lambda = 1$	74.70±5.26
OBC	$\lambda = 0.01$	75.94±7.11
	$\lambda = 0.1$	72.78±13.23
	$\lambda = 1$	77.28±4.78