Continually Learn to Map Visual Concepts to Large Language Models in Resource-constrained Environments

Clea Rebillard
Bordeaux INP - ENSEIRB-MATMECA
clea.rebillard@gmail.com

Julio Hurtado
CAMaCS, University of Warwick
julio.hurtado@warwick.ac.uk

Andrii Krutsylo
Institute of Computer Science Polish Academy of Sciences
andrii.krutsylo@ipipan.waw.pl

Lucia Passaro, Vincenzo Lomonaco
University of Pisa
{lucia.passaro, vincenzo.lomonaco}@unipi.it

ABSTRACT

Learning continually from a stream of non-i.i.d. data is an open challenge in deep learning, even more so when working in resource-constrained environments such as embedded devices. Visual models that are continually updated through supervised learning are often prone to overfitting, catastrophic forgetting, and biased representations. On the other hand, large language models contain knowledge about multiple concepts and their relations, which can foster a more robust, informed and coherent learning process. This work proposes Continual Visual Mapping (CVM), an approach that continually ground vision representations to a knowledge space extracted from a fixed Language model. Specifically, CVM continually trains a small and efficient visual model to map its representations into a conceptual space established by a fixed Large Language Model. Due to their smaller nature, CVM can be used when directly adapting large visual pre-trained models is unfeasible due to computational or data constraints. CVM overcome state-of-the-art continual learning methods on five benchmarks and offers a promising avenue for addressing generalization capabilities in continual learning, even in computationally constrained devices.

Keywords Continual Learinng · Multi-Modal · Generalization

1 Introduction

Increasing the amount of training data has repeatedly shown significant positive impacts on the performance of deep learning models [1, 2]. Generally speaking, as the quality and diversity of the training set expands, the performance of predictive models increases. This trend is particularly pronounced with *Transformer* architectures [3], which has uncovered new models capable of achieving state-of-the-art performance in multiple tasks, such as textual [2], visual [4] and multiple modalities [5], provided that the necessary amount of data is available.

The Transformer architecture has shown remarkable performance, fostering the creation of a vast collection of large pre-trained models. These models can be used for multiple tasks, especially for individuals or entities lacking the necessary resources to train such large models. Since a considerable amount of data and computational power is required to fine-tune these models, much effort has been focused on enhancing their knowledge transferability [6, 7].
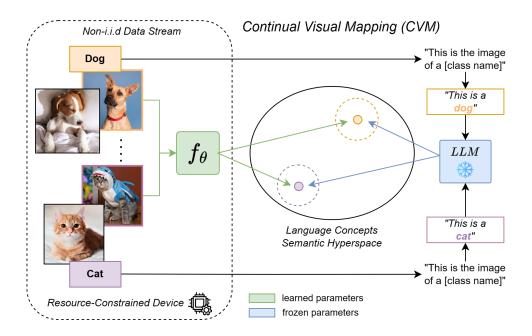
Figure 1: Conceptual image of our Continual Visual Mapping (CVM) method. A small and efficient parametric model f_θ is continually learned to map visual concepts into semantically richer embeddings seldomly extracted from frozen and pre-trained Large Language Models (LLMs).

Large pre-trained models have become popular in Continual Learning (CL). By fixing pre-trained weights and using their internal knowledge, many proposed methods have achieved state-of-the-art results in various settings [8, 9, 10]. However, continually adapting or fine-tuning large pre-trained models is computationally expensive and becomes impractical or even *not possible* in resource-constrained environments. This issue extends to inference time since the time these models take to make a prediction is noticeably superior to smaller models.

The difficulty of using these large models in resource-constrained environments creates the following question: *Can we still use knowledge extracted from large pre-trained models without using them directly in the continual learning and inference processes?*

We propose to use knowledge extracted from a *frozen* Large Language Model (LLM) to train a smaller visual model, without using it directly in the training process. This strategy allows us to train and use this knowledge in resource-constrained environments. We decided to focus on language models for three main reasons: i) LLMs have shown to define inherently rich and robust semantic spaces that can be re-used across several tasks [11]; ii) LLMs are rapidly evolving and generally more available (open-source) than visual or multi-modal models [12]; iii) LLMs allow for a direct connection between the textual labels available in visual supervised datasets, and the concepts associated with them in language.

Hence, our approach focuses on training small visual models that can map their representations to a conceptual space established by a LLM, replacing the classifier with a distance function. Similar to CLIP [5], we propose a multi-modal approach to train a visual model. However, instead of continually training the text and the visual transformer together, we obtain the representation from a frozen LLM and train only a small visual model to map the visual representations to this knowledgeable latent space. The visual model will learn the semantic information that lies in the fixed latent space. We call this approach *Continual Visual Mapping (CVM)*, and a visual representation can be found in Figure 1.

This approach is highly relevant in CL for two main reasons. Firstly, substituting the linear classifier with a stable set of anchor vectors eliminates interference generated in the classification layer, a recognized contributor to forgetting [13, 14]. Secondly, we expect the model to learn more general visual representations that will transfer better to another task, which can increase forward transfer capabilities.

Along with showing the performance of the proposed method in class-incremental and domain-incremental learning, we show that there are scenarios where even directly adapting large visual pre-trained models such as ViT can result in worse performance than CVM due to data scarcity or the specialized knowledge required. By training a smaller visual

model, CVM also becomes significantly faster in inference time, making it practical for real-time applications. The main contributions of this work can be summarised as follows:

- We propose a CL method that trains a small visual model by mapping its representations to knowledge space created by a frozen LLM (Sec. 3.2). Improving performance and mitigating forgetting in standard benchmarks (Sec. 4.2).
- We study the generalization and transfer capabilities of our proposal, showing favourable results against other CL methods (Sec. 5).
- We perform experiments in fine-grained datasets where a method based on a large pre-trained model fails. We show that CVM can achieve comparable performances (with lower inference time) with methods directly leveraging large models in training (Sec. 6).

2 Related Work

Continual Learning Learning continually in deep learning [15, 16, 17] mainly aimed at addressing *Catastrophic Forgetting* by enabling predictive models to learn new tasks without forgetting previously learned ones. Most CL methods can be categorized by different, non-exclusive computational approaches including: subdividing model parameters into subspaces for each new task [18, 19]; imposing constraints on the learned gradients [20, 21]; and using meta-learning to learn reusable weights for all tasks [22, 23]. Out of these categories, memory-based methods such as Experience Replay (ER) [24, 25] provide a straightforward solution that achieves good results. Memory-based methods address catastrophic forgetting by incorporating data from previous tasks into the training process for the current task [26, 27], and have been the state-of-the-art on most benchmarks since their ability to mitigate forgetting due to the constant repetition of previous tasks.

While using pre-trained models is not new in CL, the use of large pre-trained models has naturally shown better performance in several downstream tasks [9, 28, 8, 10], since they can take advantage of the pre-trained representations that can significantly help the mitigation of catastrophic forgetting. These models are trained on vast amounts of data, allowing them to obtain valuable representations for different modalities and inputs [4, 29], making them very useful as a starting point for CL solutions. However, their massive amount of parameters and the need to access a considerable amount of data make them impractical to train (continually) as new distribution arrives [30].

Learning Multi-Modal Representations Training predictive models able to connect different data modalities (vision, text, speech, others) have been shown to provide richer representations and more reusable representations in deep learning [2]. One of the most popular multi-modal models is CLIP [5]. Here, the authors train visual and language models to generate a latent space where visual and textual representations of the same input are close to each other. This fundamental insight has been further refined and expanded [31, 32, 33] achieving state-of-the-art performance in multiple tasks [34].

Pioneering works [35, 36, 37] have explored training a visual model while using a static language representation (i.e., before the advent of contextualized embeddings and large language models). However, unlike previous works, we take advantage of extracting contextual information. Specifically, we build the knowledge space upon pre-trained LLM, expecting our visual model to benefit from the semantic knowledge contained within the LLM. Additionally, we explore how this approach can help reduce forgetting in CL scenarios by including a retention loss that mitigates the forgetting of the semantic knowledge of the previously learned tasks in the optimisation function. Such a solution provides a resource-efficient alternative to using large pre-trained models.

3 Strategy

3.1 Background

In supervised CL, we consider a stream of T tasks. Each task t consists of a new data distribution $D^t = (X^t, Y^t)$, where X^t denotes the input instances and Y^t denotes the instance labels. The goal is to train a classification model $f_\Theta : X \longrightarrow Y$ using data from a sequence of T tasks: $D = \{D^1, ..., D^T\}$. During each task, model f_Θ will minimize the objective function $\mathcal{L}$ using data D^t.

$$\mathcal{L}(D^t) = \frac{1}{N^t} \sum_{t=1}^{N^t} \mathcal{L}_t(f_\Theta(x_i^t), y_i^t) \tag{1}$$
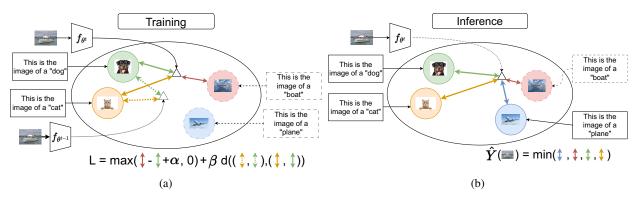
Figure 2: Diagram of how CVM works at Training and Inference. During training (a), when adding new classes (plane and boat), two losses are involved: the triplet loss with a margin α (Equation 4) and the Semantic distance loss restrained by coefficient β (Equation 5). At inference (b), the prediction $\hat{Y}$ is obtained by selecting the minimum distance between the image embedding and the seen classes in the latent space.

Each task is presented sequentially to the model and trained for E epochs. This work focuses on the traditional *Class-Incremental (Class-IL)* setting. This scenario is often intended as the most challenging in CL [38]. Unlike *Task Incremental Learning (Task-IL)*, in Class-IL, a task descriptor is only available during training. To verify the generality of our proposal, we also run experiments in a *Domain Incremental (Domain-IL)* setting. In this last setting, each task considers all classes, but the distribution of each class changes over time.

In memory-based methods, together with minimizing Equation 1, model f_Θ needs to minimize $\mathcal{L}$ using the data available in the buffer memory M at time t. The buffer M^t comprises $|M|$ samples from previous distributions, meaning that at task t, the buffer will contain a limited set of samples collected from tasks $t' < t$.

$$\mathcal{L}_M(D^t, M^t) = \mathcal{L}(D^t) + \frac{1}{|M|} \sum_{i=1}^{|M|} \mathcal{L}_t(f_\Theta(m_i^t, y_i^t) \tag{2}$$

Instead of concatenating the dataset D^t and M^t, which could neglect the representativeness of the memory, some works have adapted the concatenation of current data and memory at the batch level [39]. In this work, we follow this approach, which has been shown to increase the performance of memory-based methods.

3.2 Continual Visual Mapping (CVM)

In a classic formulation of a classification problem, a model f_Θ is composed of a feature extractor f_θ and a classifier c_ω, been θ and ω the trainable weights of the model f_Θ. During training, the learning strategy simultaneously tries to learn robust, semantic representations of the input (f_θ) and a classifier (c_ω) that uses these features to discriminate between the classes correctly, ideally generalising to samples not seen during training.

In CL, the constant modification of the weights θ and ω results in ongoing interference between tasks. This interference can induce forgetting, which causes a decrease in the overall performance. We hypothesise that this interference is primarily due to the biases and boundaries the model learns during a task, knowledge that becomes unsuitable for future distributions. These biases are reinforced by the limited information the model receives through the input, which forces it to construct mappings based on limited visual information. Collectively, these factors strongly restrict the transferability of the learned representations to future tasks due to bias, which makes them prone to overfitting and less reusability.

Our proposed method replaces the classifier c_ω with a set of *anchor vectors* representing the classes or concepts. These vectors are extracted from a pre-trained LLM, creating a fixed and knowledgeable latent space. The vectors establish a fixed latent space that the visual model can use as external knowledge. With this modification, we expect to (1) leverage the knowledgeable space created by the anchor vector to guide the training of the visual model and (2) increase the transferability of the visual representations.

Using a pre-trained LLM, we extract the current task D^t representations, creating a latent space imbued with a broad understanding of the world [40]. Given the absence of textual descriptions for images in most visual datasets, we generate a vector C_c for each class c, using the prompt *"This is an image of "* + *label*, where *label* represents the

4

text associated with each class (i.e., the class name). Importantly, the LLM is not employed during training, as our focus is solely on obtaining fixed vectors to replace the classifier. This strategy can be easily implemented in resource-constrained environments as a *one-time query* to an external service.

Once the representations of classes C at task t are generated, we can obtain the corresponding class with a distance function. As shown in Equation 3, we compare the visual representations of the image x_i with the known vectors in the latent space and select the nearest class. It is important to note that at time t, C is composed only by seen classes. A visual representation of how the model works in inference can be found in Figure 2b.

$$\hat{Y}_i = min(d(f_\theta(x_i), C))$$ (3)

We use a Triplet Loss [41] to train the visual model. As shown in Equation 4, the objective is to minimise the distance between the feature vector $f_\theta(x_i)$ and the representations of the corresponding class in the latent space C_i^P, while concurrently increasing the distance between $f_\theta(x_i)$ and negative classes. In all our experiments, we employ the cosine distance. The ultimate goal is to have the visual model map its representations to a fixed latent space created with knowledge gleaned from the LLM, thereby aligning both sources of information. This integration forms the basis of our approach, which we term Continual Visual Mapping (CVM).

$$L_m = \frac{1}{N} \sum_i^N max(0, d(f_\theta(x_i), C_i^P) - d(f_\theta(x_i), C_i^N) + \alpha)$$ (4)

In addition to mapping visual representations to the knowledge space, the visual model must grasp semantic information, specifically, the relationships between concepts. This knowledge acquisition is anticipated to enhance the generalization capabilities of the visual model, facilitated by the knowledge of the anchor vectors. However, the loss function L_m operates under the assumption that all classes exhibit the same semantic distance, overlooking potential similarities between certain classes that share common characteristics and knowledge, such as the similarity between dogs and cats compared to aeroplanes.

To address this issue, we introduce a novel loss function designed to consider previous similarities learned by the model. Our proposal involves preserving the distances between classes encountered in the past. To achieve this, we calculate the distance between the representations of an input x_i and its corresponding class C^{t-1} using both the current model f_{θ^t} and the model at time $t-1$. As shown in Equation 5, our objective is to minimize the discrepancy between these distances, thereby incorporating past similarities into the learning process.

$$L_d = \frac{1}{N} \sum_i^N d(d(f_{\theta^t}(x_i), C^{t-1}), d(f_{\theta^{t-1}}(x_i), C^{t-1}))$$ (5)

It is worth noting that Equation 5 only compares against classes seen before the current experiences to maintain the alignment in learned space.

We combine the losses according to Equation 6. We add a β coefficient to control how much the model needs to remember previously learned distance between classes. Figure 2a visually represents the training loss.

$$L_t = L_m + \beta L_d$$ (6)

4 Experiments

To showcase the effectiveness of CVM, we have empirically assessed it in various scenarios and benchmarks traditionally used in CL and following standard practices. We tested our method in a Class-IL scenario using CIFAR100 [42] and Tiny-ImageNet [43] datasets. We split these datasets into 10 experiences. We also run experiments in a Domain-IL scenario and higher-dimensional input spaces using the more realistic CORe50 dataset [44], designed explicitly for embedded continual object recognition applications. Here, the setting is composed of 10 domestic objects, and each new experience imposes a new class distribution (e.g., background, illumination, occlusion, scale, among others).

Table 1: Accuracy and forgetting for CIFAR100, Tiny-ImageNet, and CORe50. CVM outperforms all methods regarding average accuracy due to the combination of L_m and the semantic loss. It is worth noting that we cannot measure the forgetting metric in CORe50 since it is designed with a unique and fixed test set.

| | Class-IL | | | | Domain-IL |
| | CIFAR100 | | Tiny-ImageNet | | CORe50 |
	Acc	For	Acc	For	Acc
Naive	9.0% $_{\pm 1.7}$	78.6% $_{\pm 0.8}$	7.4% $_{\pm 0.1}$	73.9% $_{\pm 0.3}$	23.1% $_{\pm 1.0}$
AGEM	9.3% $_{\pm 0.9}$	77.8% $_{\pm 0.3}$	7.5% $_{\pm 0.1}$	74.7% $_{\pm 0.3}$	26.0% $_{\pm 1.3}$
LWF	8.8% $_{\pm 1.5}$	76.3% $_{\pm 1.0}$	7.4% $_{\pm 0.0}$	73.0% $_{\pm 0.4}$	31.7% $_{\pm 2.2}$
EWC	8.6% $_{\pm 0.2}$	77.7% $_{\pm 0.7}$	7.3% $_{\pm 0.1}$	70.7% $_{\pm 0.6}$	24.0% $_{\pm 3.4}$
ER	21.0% $_{\pm 0.1}$	65.0% $_{\pm 0.7}$	13.1% $_{\pm 0.2}$	65.8% $_{\pm 0.2}$	33.0% $_{\pm 0.4}$
+ L_m	20.9% $_{\pm 0.7}$	64.7% $_{\pm 0.5}$	10.7% $_{\pm 0.3}$	66.2% $_{\pm 0.6}$	32.8% $_{\pm 1.3}$
ER + EWC	19.2% $_{\pm 0.4}$	64.8% $_{\pm 0.7}$	17.7% $_{\pm 1.0}$	55.5% $_{\pm 0.6}$	31.9% $_{\pm 0.1}$
+ L_m	25.2% $_{\pm 0.6}$	51.9% $_{\pm 0.2}$	11.5% $_{\pm 1.2}$	58.7% $_{\pm 0.8}$	32.2% $_{\pm 2.3}$
DER	19.1% $_{\pm 0.8}$	65.3% $_{\pm 0.1}$	3.9% $_{\pm 0.4}$	64.1% $_{\pm 0.7}$	34.8% $_{\pm 2.3}$
+ L_m	21.0% $_{\pm 1.2}$	64.4% $_{\pm 0.6}$	11.7% $_{\pm 0.4}$	64.9% $_{\pm 0.5}$	33.1% $_{\pm 0.6}$
iCarl	29.1% $_{\pm 1.1}$	23.2% $_{\pm 0.2}$	24.3% $_{\pm 0.2}$	16.2% $_{\pm 0.1}$	17.8% $_{\pm 1.8}$
CVM	**32.9%** $_{\pm 0.4}$	40.1% $_{\pm 0.8}$	**27.0%** $_{\pm 0.8}$	33.9% $_{\pm 0.8}$	**37.6%** $_{\pm 1.1}$

4.1 Implementation Details

We use a reduced version of a ResNet architecture proposed in [45] for the visual model to simulate a resource-constrained environment. The only difference between the model used in CVM and other baselines is that in CVM, we remove the classifier. This modification causes CVM to train fewer parameters. For the text model, we use SentenceBERT [46] to extract the textual representation for the classes. We ran all the experiments with three different seeds, each inducing a different ordering of the sequences of tasks.

We compare our proposal to different types of methods. First, we compare our method against some classic regularization-based CL methods like AGEM [47], EWC [20], and LwF [48]. We also compare against memory-based methods like ER [39], DER++ [49] and iCarl [45]. iCarl is especially interesting to our case since it also removes the classifier at inference time. However, the methods differ mainly in training, as iCarl uses a cross-entropy loss with labels instead of the distance to knowledge anchor vectors. The way iCarl generates the vector per class is also different since the vectors are generated based on samples in the memory.

In all experiments, we use an SGD optimizer with a learning rate of 0.1 and a batch size of 32. CIFAR100 and TinyImagenet train for 50 epochs per experience; for CORe50, we train for 30 epochs. We use the implementation provided by Avalanche [50]. We used the hyper-parameters proposed by the authors in the corresponding paper. The code of CVM will become available upon acceptance.

We evaluate the average accuracy (Acc) and forgetting (For) over the T tasks after the sequential learning, proposed in [21]. Equations 7 and 8 show the formula for the accuracy and forgetting, respectively, where $Acc_{i,j}$ is the accuracy of task i after training task j.

$$Acc = \frac{1}{T} \sum_{i=1}^{T} Acc_{T,i} \tag{7}$$

$$For = \frac{1}{T-1} \sum_{i=1}^{T-1} Acc_{T,i} - Acc_{i,i} \tag{8}$$

4.2 Results

The results for Class-IL benchmarks are provided in Table 1. As previous studies have shown, regularization methods are suboptimal in this scenario. However, performance gains can be achieved by leveraging access to previous tasks stored in a buffer, exemplified by ER and DER++. Additionally, combining EWC with classical ER only seems to increase performance in Tiny-ImageNet.

Table 2: We use the CIFAR100 benchmark to test the performance with different memory sizes. Methods scale differently; iCarl does not scale well, increasing only 10%, DER increases performance quite a bit (31%). CVM still outperform in every scenario.

	Memory Size				
	500	1000	2000	3000	5000
ER	21.0%	28.2%	36.8%	40.5%	47.3%
+ L_m	20.9%	27.9%	37.7%	41.8%	47.3%
ER + EWC	19.2%	26.1%	35.0%	39.4%	43.6%
+ L_m	25.2%	26.1%	35.5%	39.1%	46.8%
DER	19.1%	26.6%	39.3%	45.3%	50.9%
+ L_m	21.0%	28.2%	39.6%	43.1%	47.5%
iCarl	29.1%	36.4%	37.7%	38.8%	39.6%
CVM	**32.9%**	**38.7%**	**43.5%**	**47.5%**	**51.4%**

To boost the efficacy of memory-based approaches, replacing Cross-Entropy Loss with the mapping loss (Equation 4) proves to be a powerful learning strategy. This strategy significantly increases accuracy, particularly in ER + EWC, where the accuracy increases 6%. The model gains valuable insights by incorporating additional information from latent representations, which is particularly advantageous in CL scenarios where knowledgeable representations can enhance generalization. However, it is noteworthy that this transferability is most pronounced in simple benchmarks like CIFAR100. In contrast, on more complex benchmarks such as Tiny-ImageNet and CORe50, L_m does not contribute to performance improvement. This is primarily attributed to higher forgetting, indicating that complex benchmarks need extra encouragement to mitigate forgetting.

To mitigate the forgetting, we propose the novel loss L_d. This loss is specifically crafted to influence the distances between classes, acknowledging that certain classes are more similar than others, which is not considered in the traditional triplet loss. Combining this loss with L_m creates CVM, which aims to extract valuable knowledge from the latent space while preserving the inherent class relationships. Our experimental results, as detailed in Table 1, confirm that CVM outperforms previous methods, including iCarl, which also removes the classifier during inference. These findings highlight that *relying solely on information extraction from an LLM is insufficient*, emphasizing the necessity of incorporating additional mechanisms in CL scenarios.

Similar results can be seen in CORe50, as shown in Table 1. Most regularization techniques find it challenging to perform reasonably, and memory-based methods produce the best results. iCarl, despite being a memory-based method, also faces difficulties due to the constant shift of the class distribution. Here, introducing L_m does not aid ER nor DER++ in enhancing their outcomes. However, CVM still surpasses its predecessors by encouraging the reuse of previous distances.
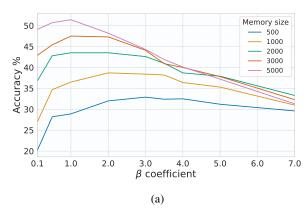
4.3 Ablation

In CL, preserving the plasticity of the model is fundamental to learning new tasks and distributions. However, the trade-off between learning new classes and forgetting the previous one is the primary concern that most CL methods face.

In memory-based methods, increasing the memory size improves the representativeness of the buffer, which should increase performance by decreasing forgetting. As shown in Table 2, DER++ is the method that scales the best as we increase the buffer size, significantly increasing its performance; however, it still achieves worse results than CVM, which consistently outperforms previous methods. On the other hand, iCarl scales poorly.

Something that explains the low performance of iCarl is plasticity. As shown in Table 1, iCarl obtained a lower forgetting value. These results can be explained by the significant restrictions that iCarl imposes when training new tasks. These constraints become more evident as we increase the memory size. iCarl can maintain a lower level of forgetfulness but at the cost of having low plasticity and being unable to scale up when we increase memory size. In contrast, CVM has no problem scaling up and maintaining a high level of plasticity.

We present two compelling arguments affirming the effectiveness of CVM and the importance of the proposed learning strategy for CL. Firstly, our approach involves mapping visual representations to a stable and well-informed space, removing the need for a classifier. Previous studies highlight the classifier as a critical contributor to model overfitting, particularly in challenging CL environments [13]. This finding gains further support from the insights in Table 2, where enhancing memory representativeness enhances overall performance and mitigates the risk of overfitting to
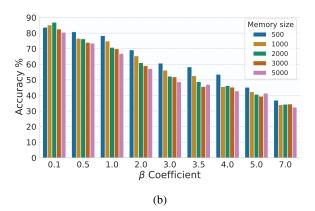
Figure 3: Ablation study over the β coefficient. Intuitively, β controls how much to consider past distances, impacting performance. In 3a, it is shown that the optimal coefficient tends to decrease as we increase the memory size since memory provides a better representation of past distributions. On the other side, β controls the plasticity of a model. Figure 3b shows the accuracy of the last task. Increasing β encourages the model to maintain previous semantic knowledge while decreasing plasticity.

Table 3: The knowledgeable space created by the LLM is critical for CVM. As the performance of these models grows, our method should obtain a better approximation of a semantic space. Here, we test different LLMs from where to extract the world representation.

	SentenceBERT	Text CLIP	BERT
Acc	$32.9\%_{\pm 0.4}$	$31.0\%_{\pm 0.2}$	$29.8\%_{\pm 0.6}$

limited memory capacities. Secondly, the strategic alignment of concepts in the latent space is crucial. The semantic information within the space generated by a frozen LLM discerns the similarities among diverse classes. This concept is actively promoted by incorporating the loss described in Equation 5.

CVM components A critical component of CVM is the β coefficient, which, as shown in Figure 3a, the optimal value is strongly related to the memory size. As we increase the memory size, the model can repeat more concepts presented in the buffer, which decreases the need to remember the exact distance of previous tasks, meaning that CVM can work with a smaller β. On the other hand, if we decrease the memory size, the model must depend mainly on the loss function to maintain semantic knowledge learned in the past.
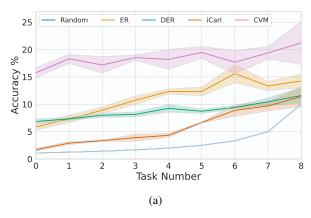
The second experiment about β concerns the plasticity of the model. As shown in figure 3b, the accuracy of the last task will decrease as we increase the β coefficient. This is explained by discouragement in moving the weights when training the new task, reducing the plasticity. With a high value of β, the loss promotes fewer modifications on the model weights, reducing forgetting but decreasing the final accuracy.

Another critical component of CVM is the LLM from where to create the fixed latent space. Maintaining the prompt fixed, we compare SentenceBERT with the textual model of CLIP and BERT. Results are shown in Table 3. These results show that using an appropriate model to generate the latent space is essential. However, future work will explore using more detailed descriptions or concepts extracted from the image instead of directly using the class.

5 Generalization Capabilities

For a continual learner to be efficient, it should reuse previous knowledge when learning new tasks. It is hypothesised that reusing previous knowledge could mitigate forgetting since the model would have no incentive to modify that knowledge. On this note, an essential feature in CL models is the generalization ability due to the transfer capability to *future tasks*.

There are two distinct approaches to assessing the transfer capability of a model. The first approach involves freezing the model and training a linear classifier (linear probing) on unseen classes. This evaluation method measures the generality of the representations acquired by the model for future tasks. The second approach focuses on evaluating the zero-shot capabilities of the model. In the case of CVM, this involves incorporating an anchor vector of unseen classes and verifying the accuracy exclusively for those classes while keeping the visual model frozen. This method provides insights into the model's ability to classify classes that are not part of the training set.

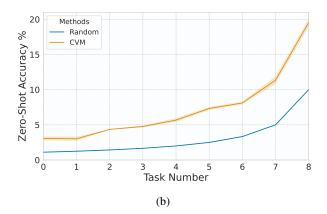(a)   (b)

Figure 4: Using CIFAR100, we evaluate the approaches over unseen tasks, meaning that after Task 0, we have 90 unseen classes. Figure 4a shows the accuracy when freezing the model and learning a linear classifier with the unseen classes after training each task. Random denotes a random assignation of classes. Figure 4b shows the accuracy of the zero-shot performance of CVM compared to a random classification. After training task t, we test the zero-shot capabilities on classes of tasks $t' > t$ and compare them with randomly assigned classes: $1/|unseen|$.

Table 4: The FW_{Score} is a metric that helps compare the transfer capacity of different CL methods. CVM outperforms previous methods of learning representations that learn unseen classes using two approaches.

	Random	ER	DER	iCarl	CVM
FW_{Score}	3.1%	11.2%	8.9%	5.9%	18.4%
Zero-shot Acc.	3.1%	-	-	-	7.9%

In the first approach, reusable representations indicate how prepared a model is to learn new tasks. A low value in accuracy indicates that the model should highly modify its representation to learn new tasks; on the other hand, a higher accuracy value indicates that minor modifications are needed to adjust to new tasks. The results of applying the linear probing are shown in Figure 4a. The results indicate that the learned representations of CVM have the highest transfer capabilities, ranging from 16% when comparing with 90 unseen classes (after training the first task) to about 22% when only 10 classes remain. In contrast, iCarl fails to generate good representations, as it has close to random values.

The zero-shot capabilities take the previous observation to an extreme. If a model can learn to identify a class without modifying its weights, it learns the new distribution without forgetting. However, methods with a classifier are limited to the classes seen due to constraints in how the classifier is optimized and structured. In contrast, CVM holds a distinct advantage. By relying on a distance metric instead of a trained classifier, CVM can classify classes not encountered during training by adding an anchor vector that represents the textual description of the new class. In this context, the zero-shot accuracy performance of the model is computed on unseen classes for tasks $t' > t$ when t is the current task. Results are illustrated in Figure 4b. It is noteworthy that methods with a classifier exhibit a performance of 0% due to the inherent limitations of the classifier.

Unlike accuracy, where the final value represents how well methods perform, the transfer capabilities are better measured over the complete sequence. We developed a simple scoring system to compare the transfer capabilities between various methods easily. This score is obtained by averaging the accuracy achieved by each metric after completing each task. For instance, in the case of linear probing, we train a linear classifier (f_γ^t) after each task (t), using only data from unseen tasks ($t' > t$), i.e., $X^{t'>t}$ and $Y^{t'>t}$. This score helps us understand how well the method performs on unseen classes. We average all these performances, as shown in Equation 9.

$$FW_{score} = \frac{1}{T-1} \sum_{t=1}^{T-1} Acc(f_\gamma^t, X^{t'>t}, Y^{t'>t}) \tag{9}$$

A higher value means that the representations learnt during the sequence are more capable of being used in future tasks. The results of the score for each method can be found in Table 4.

Table 5: Two CL benchmarks were created with granular datasets. In these benchmarks, visual models outperform CL methods based on large pre-trained models in terms of accuracy and inference time.

	GTSRB		Aircraft	
	Acc	Inf. Time	Acc	Inf. Time
L2P	32.4% $_{\pm 0.4}$	~ 188	34.0% $_{\pm 3.7}$	~ 100
L2P + ER	74.9% $_{\pm 4.1}$	~ 188	34.9% $_{\pm 1.3}$	~ 100
ER	79.4% $_{\pm 0.2}$	~ 14	35.0% $_{\pm 1.6}$	~ 65
DER	83.6% $_{\pm 1.4}$	~ 14	36.1% $_{\pm 2.3}$	~ 65
CVM	**84.8%** $_{\pm 1.3}$	~ 14	**36.5%** $_{\pm 1.1}$	~ 65

6   Limitations of Pre-trained Models

Even in scenarios where we have ample computational power, other issues can make methods based on large visual pre-trained models unsuitable, e.g., the need for more data, the specificity or even the granularity of the problem.

We introduce two Continual Learning (CL) benchmarks utilizing the GTSRB [51] and Aircraft [52] datasets. For the GTSRB dataset, we delineate 10 tasks, while the Aircraft dataset is partitioned into 5 tasks, both adhering to a Class-IL scenario.

The primary challenge confronted by methods relying on pre-trained models arises from the granularity of the dataset. This fine-grained nature necessitates additional information not present in the weights of large pre-trained models. Consequently, pre-trained models require assistance to learn the sequence, leading to low performance. In Table 5, we present the performance of L2P [9] both with and without memory. Notably, these approaches fail to outperform methods employing simpler models. Although pre-trained models may perform similarly to simpler and smaller models, their longer inference times can adversely impact overall performance.

7   Conclusions

We presented Continual Visual Mapping (CVM), which uses a conceptual latent space created by a frozen Large Language Model (LLM) to train a small visual model. By learning to map the visual representations to a knowledgeable space, the visual model can learn new concepts based on the knowledge extracted through pre-learned textual information. We demonstrate the performance of CVM in class-incremental and domain-incremental learning using various benchmarks, achieving state-of-the-art results within resource-constrained environments. We also reveal scenarios where simple and small models can be more efficient (and surprisingly, even more effective) than methods based on large pre-trained models. In particular, these are scenarios where the low transferability of these methods affects the performance. Our learning strategy opens new research directions for CL methods to asynchronously take advantage of the knowledge of large pre-trained models efficiently and effectively mapping new knowledge into existing ones.

Acknowledgments

References

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[2] R OpenAI. Gpt-4 technical report. *arXiv*, pages 2303–08774, 2023.

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[5] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[6] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, 2021.

[7] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021.

[8] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision*, pages 631–648. Springer, 2022.

[9] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149, 2022.

[10] Andrés Villa, Juan León Alcázar, Motasem Alfarra, Kumail Alhamoud, Julio Hurtado, Fabian Caba Heilbron, Alvaro Soto, and Bernard Ghanem. Pivot: Prompting for video continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24214–24223, 2023.

[11] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022.

[12] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

[13] Felipe del Rio, Julio Hurtado, Cristian Buc, Alvaro Soto, and Vincenzo Lomonaco. Studying generalization on memory-based methods in continual learning. *The Second Workshop on Spurious Correlations, Invariance and Stability at ICML*, 2023.

[14] MohammadReza Davari, Nader Asadi, Sudhir Mudur, Rahaf Aljundi, and Eugene Belilovsky. Probing representation forgetting in supervised and unsupervised continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16712–16721, 2022.

[15] 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.

[16] 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*, 2021.

[17] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *arXiv preprint arXiv:2302.00487*, 2023.

[18] 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.

[19] 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*, 2018.

[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*, 2017.

[21] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in neural information processing systems*, 2017.

[22] Jathushan Rajasegaran, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Mubarak Shah. itaml: An incremental task-agnostic meta-learning approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[23] Julio Hurtado, Alain Raymond, and Alvaro Soto. Optimizing reusable knowledge for continual learning via metalearning. *Advances in Neural Information Processing Systems*, 34, 2021.

[24] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[25] Jeffrey S. Vitter. Random sampling with a reservoir. *ACM Trans. Math. Softw.*, 11(1):37–57, 1985.

[26] Sayna Ebrahimi, Suzanne Petryk, Akash Gokul, William Gan, Joseph E. Gonzalez, Marcus Rohrbach, and trevor darrell. Remembering for the right reasons: Explanations reduce catastrophic forgetting. In *International Conference on Learning Representations*, 2021.

[27] Pietro Buzzega, Matteo Boschini, Angelo Porrello, and Simone Calderara. Rethinking experience replay: a bag of tricks for continual learning. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 2180–2187. IEEE, 2021.

[28] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision*, pages 709–727. Springer, 2022.

[29] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[30] Andrea Cossu, Tinne Tuytelaars, Antonio Carta, Lucia Passaro, Vincenzo Lomonaco, and Davide Bacciu. Continual pre-training mitigates forgetting in language and vision. *arXiv preprint arXiv:2205.09357*, 2022.

[31] Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Qiang Liu, et al. Language is not all you need: Aligning perception with language models. *arXiv preprint arXiv:2302.14045*, 2023.

[32] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. *CoRR*, abs/2102.05918, 2021.

[33] Karan Desai and Justin Johnson. Virtex: Learning visual representations from textual annotations. *CoRR*, abs/2006.06666, 2020.

[34] Mert Bülent Sariyildiz, Julien Perez, and Diane Larlus. Learning visual representations with caption annotations. *CoRR*, abs/2008.01392, 2020.

[35] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc' Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[36] Jason Weston, Samy Bengio, and Nicolas Usunier. Large scale image annotation: learning to rank with joint word-image embeddings. *Machine learning*, 81:21–35, 2010.

[37] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. Zero-shot learning through cross-modal transfer. *Advances in neural information processing systems*, 26, 2013.

[38] Gido M Van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019.

[39] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and M Ranzato. Continual learning with tiny episodic memories. *International Conference on Machine Learning*, 2019.

[40] Christopher D Manning. Human language understanding & reasoning. *Daedalus*, 151(2):127–138, 2022.

[41] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[42] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *http://www.cs.toronto.edu/ kriz/cifar.html*, 2009.

[43] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

[44] Vincenzo Lomonaco and Davide Maltoni. Core50: a new dataset and benchmark for continuous object recognition. In Sergey Levine, Vincent Vanhoucke, and Ken Goldberg, editors, *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pages 17–26. PMLR, 13–15 Nov 2017.

[45] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017.

[46] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, 2019.

[47] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018.

[48] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

[49] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020.

[50] Vincenzo Lomonaco, Lorenzo Pellegrini, Andrea Cossu, Antonio Carta, Gabriele Graffieti, Tyler L. Hayes, Matthias De Lange, Marc Masana, Jary Pomponi, Gido van de Ven, Martin Mundt, Qi She, Keiland Cooper, Jeremy Forest, Eden Belouadah, Simone Calderara, German I. Parisi, Fabio Cuzzolin, Andreas Tolias, Simone Scardapane, Luca Antiga, Subutai Amhad, Adrian Popescu, Christopher Kanan, Joost van de Weijer, Tinne Tuytelaars, Davide Bacciu, and Davide Maltoni. Avalanche: an end-to-end library for continual learning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2nd Continual Learning in Computer Vision Workshop, 2021.

[51] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In *International Joint Conference on Neural Networks*, number 1288, 2013.

[52] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013.