Continual Self-Supervised Learning with Knowledge Distillation

Essam Sleiman Xiangbo Li Saad Ali

Twitch (Amazon)

Abstract

Self-supervised representation learning offers a practical way to use large unlabeled datasets to build reusable embeddings for downstream tasks, but deployed data distributions are not static. As new data arrives, retraining from scratch can be expensive, while continual training on only the newest data can degrade representations learned from earlier distributions. This work studies continual self-supervised learning with knowledge distillation. Starting from a SimCLR encoder trained on an initial data distribution, we continue training on a later distribution while distilling the pairwise similarity structure of the original embedding space. Unlike supervised Learning Without Forgetting, which preserves class-probability outputs, our objective preserves relationships between unlabeled examples. We evaluate this approach on temporal Twitch stream datasets and two downstream classification tasks, with an additional ImageNet-to-CelebA validation setting to test a stronger distribution shift. On Twitch, we observe limited downstream forgetting under the sampled temporal split, suggesting substantial semantic overlap between time periods for the tasks studied. Under the stronger ImageNet-to-CelebA shift, similarity-preserving distillation reduces the increase in original-distribution SimCLR loss by approximately 9% relative to regular continual training. These results suggest that knowledge distillation can help preserve self-supervised representations during continual updates, while also showing that the benefits depend strongly on the amount of distribution shift.

1. Introduction

Self-supervised representation learning has become a practical way to exploit large unlabeled datasets. Instead of collecting task-specific labels for every downstream problem, a model can first learn a reusable embedding space from unlabeled data and then be fine-tuned with smaller labeled datasets. This is especially useful in production settings where unlabeled data is abundant, downstream tasks change over time, and labeling every new task is expensive.

A core limitation of this setup is that real data distributions are not fixed. New content, behaviors, formats, and visual patterns can appear over time, while older patterns may become less frequent. A representation model trained once may therefore become stale as the deployment distribution changes. One solution is to periodically retrain the model from scratch on all historical and newly collected data, but this can be computationally expensive and may require retaining large amounts of old data. A cheaper alternative is to continue training the existing model on newly collected data. However, naive continual training can lead to catastrophic forgetting, where performance on the earlier distribution degrades after adapting to the new one.

Knowledge distillation provides one way to reduce forgetting. In supervised continual learning, methods such as Learning Without Forgetting preserve the outputs of an earlier model while training on new data (Li & Hoiem, 2018). However, self-supervised representation learning does not produce supervised class probabilities that can be directly preserved. The relevant object is not a class-logit distribution, but the structure of the learned embedding space. This motivates a distillation objective that preserves relationships between examples rather than labels or logits.

In this work, we study continual self-supervised learning with knowledge distillation. Starting from a SimCLR encoder trained on an initial data distribution, we continue training on a later distribution while encouraging the updated model to preserve the pairwise similarity structure of the original embedding space. This adapts the central idea of Learning Without Forgetting to a self-supervised setting: the teacher model defines relationships among unlabeled examples, and the student is penalized for changing those relationships while learning from new data.

We evaluate this approach in two settings. First, we use temporal datasets of Twitch stream frames to study continual updates in a production platform setting, with downstream evaluation on IP violation detection and game

stream classification. Second, we use an ImageNet-to-CelebA validation setting to test the method under a larger distribution shift. The Twitch experiments show limited downstream forgetting under the sampled temporal split, suggesting substantial semantic overlap between the two time periods for the tasks studied. The ImageNet-to-CelebA experiment exposes a stronger forgetting effect and shows that similarity-preserving distillation reduces the increase in original-distribution SimCLR loss by approximately 9% relative to regular continual training.

This work makes three contributions. First, we motivate continual self-supervised representation learning as a practical alternative to repeatedly retraining reusable embeddings from scratch. Second, we adapt knowledge distillation to self-supervised learning by preserving pairwise embedding similarities rather than supervised class outputs. Third, we evaluate the approach on both temporal production data and a stronger public distribution-shift setting, showing that the benefit of distillation depends on the degree of distribution shift.

2. Background and Related Work

2.1. Self-Supervised Representation Learning

Self-supervised representation learning aims to learn useful embeddings from unlabeled data by constructing supervisory signals from the data itself. This makes it attractive when unlabeled data is abundant but task-specific labels are expensive. A common strategy is to train an encoder on a pretext objective and then reuse the representation for downstream linear evaluation or fine-tuning.

This work uses SimCLR as the base objective. SimCLR samples two augmented views of the same image, passes both through a shared encoder and projection head, and optimizes a contrastive loss that pulls together representations of the same image while pushing apart representations of different images in the batch (Chen et al., 2020). Other self-supervised methods use clustering, prediction, redundancy reduction, self-distillation, or masked reconstruction objectives (Grill et al., 2020; Caron et al., 2020; Zbontar et al., 2021; Bardes et al., 2022; Caron et al., 2021; He et al., 2022). Although these objectives differ, they share the goal of learning transferable representations from unlabeled data.

2.2. Continual Learning and Catastrophic Forgetting

Continual learning studies settings where a model learns from sequentially arriving data rather than from a single fixed training distribution. In such settings, naively updating a model on new data can degrade performance on earlier data, a phenomenon known as catastrophic forgetting. This is especially relevant for reusable representations: if an embedding model is continually updated to fit a new distribution, it may lose structure that made it useful for earlier downstream tasks.

A common way to reduce forgetting is to constrain updates so that the new model remains compatible with the old model. Some methods do this through replay, where examples or exemplars from earlier distributions are mixed into later training. Others use regularization or distillation, where the updated model is penalized for changing behavior that was learned previously. Learning Without Forgetting is a representative distillation-based approach: it trains on new-task data while preserving the old model's output distribution, allowing a model to learn new tasks without directly retraining on the full old dataset (Li & Hoiem, 2018).

However, supervised continual learning methods usually assume that the previous model produces class logits or class probabilities. In self-supervised representation learning, there is no fixed label space to preserve. The object to preserve is instead the representation space: the geometry of embeddings and the relationships among examples induced by the encoder.

2.3. Knowledge Distillation for Representation Preservation

Knowledge distillation was introduced as a way to transfer knowledge from a teacher model into a student by matching the teacher's softened output distribution (Hinton et al., 2015). In continual learning, the same idea can preserve behavior from a previous model while training on new data.

For representation learning, matching final class probabilities is not sufficient or may not be available. Several distillation approaches therefore preserve intermediate representations or relationships between representations. Similarity-preserving knowledge distillation trains a student so that pairs of inputs that produce similar or dissimilar activations in the teacher also produce similar or dissimilar activations in the student (Tung & Mori, 2019). Relational distillation methods similarly emphasize distances, angles, or other relations among examples rather than only matching individual outputs.

This paper follows the same principle in a continual self-supervised setting. Instead of preserving supervised logits, we preserve pairwise similarity structure in the embedding space. The teacher encoder defines a set of relationships among examples from the earlier distribution, and the student is encouraged to retain those relationships while continuing to learn from later data.

2.4. Continual Self-Supervised Learning

Recent work has begun to study continual learning in self-supervised representation learning directly. This setting differs from supervised continual learning because the model is preserving a representation that must remain useful for future downstream tasks, not only task accuracy. Co2L combines contrastive representation learning with rehearsal and self-supervised distillation (Cha et al., 2021). CaSSLe converts self-supervised objectives into continual distillation mechanisms by adding a predictor that maps current representations to their previous state (Fini et al., 2022). Other work studies continuous self-supervised learning under non-i.i.d. streams and highlights the difficulty of learning from sequentially arriving unlabeled data without losing earlier representation quality (Purushwalkam et al., 2022).

Our work is complementary to this emerging literature. Rather than benchmarking a large set of algorithms, we study a simple similarity-preserving distillation objective in a production temporal-data setting. The goal is to understand whether a self-supervised encoder can be updated on later unlabeled data while preserving useful representation structure from an earlier distribution.

3. Method

We study continual self-supervised representation learning across two unlabeled data distributions. Let D_1 denote an earlier distribution and D_2 denote a later distribution. The goal is to update an encoder on D_2 while preserving useful representation structure learned from D_1. This setting captures the practical case where new unlabeled data becomes available over time, but repeatedly retraining from scratch on all historical data is expensive.

3.1. Problem Setup

Let f_θ be an encoder trained with a self-supervised objective. We first train f_θ on D_1 using SimCLR. After this initial training stage, we freeze a copy of the encoder as the teacher, denoted f_{θ_T}. A student encoder f_θ is initialized from the teacher and then continually trained on D_2.

The regular continual training baseline updates the student only with the SimCLR objective on D_2. Our method adds a distillation term that anchors the student to the teacher's representation structure on reference examples from D_1. Thus, the student learns from new data while being penalized for changing the relationships among examples from the earlier distribution.

3.2. Base Self-Supervised Objective

We use SimCLR as the base self-supervised learning objective. For each input, SimCLR samples two stochastic augmentations, encodes both views using a shared encoder and projection head, and applies the normalized temperature-scaled cross-entropy loss. The objective encourages two augmented views of the same example to have similar projected representations while separating representations from different examples in the batch.

During continual training, the student continues to learn from later data using the SimCLR objective. For a batch $B_2 \sim D_2$, we write this loss as

$$\mathcal{L}_{\mathrm{SSL}}(B_2;\theta) = \mathcal{L}_{\text{NT-Xent}}(B_2;\theta). \tag{1}$$

This term gives the model plasticity: it allows the representation to adapt to new examples from the later distribution.

3.3. Similarity-Preserving Knowledge Distillation

In supervised Learning Without Forgetting, the old model's knowledge is represented by its class-probability outputs. The student is trained to match those outputs while learning new tasks. In self-supervised representation learning, there are no supervised class probabilities to preserve. The relevant object is instead the geometry of the embedding space: which examples are close, which examples are far, and how examples are arranged relative to one another.

We therefore preserve pairwise similarities among examples from the earlier distribution. For a reference batch $B_1 = \{x_1, \ldots, x_M\}$ sampled from D_1, define the pairwise similarity matrix induced by encoder f_θ:

$$S_\theta(B_1)_{m,n} = \mathrm{sim}\left(f_\theta(x_m), f_\theta(x_n)\right), \tag{2}$$

where $\mathrm{sim}(\cdot,\cdot)$ denotes cosine similarity. The frozen teacher defines the original similarity structure $S_{\theta_T}(B_1)$, and the student defines the updated similarity structure $S_\theta(B_1)$. We penalize the student when its pairwise similarity matrix moves away from the teacher's:

$$\mathcal{L}_{\mathrm{distill}}(B_1;\theta) = \left\| S_\theta(B_1) - S_{\theta_T}(B_1) \right\|_F^2. \tag{3}$$

This objective does not require labels. It only requires reference examples from the earlier distribution and the frozen teacher encoder. The distillation term preserves relationships between examples rather than preserving individual embedding coordinates or class logits.

3.4. Continual Training Objective

The final continual training objective combines self-supervised learning on the later distribution with similarity-preserving distillation on the earlier distribution:

$$\mathcal{L}_{\mathrm{continual}} = \mathcal{L}_{\text{NT-Xent}}(B_2;\theta) + \lambda \left\| S_\theta(B_1) - S_{\theta_T}(B_1) \right\|_F^2, \tag{4}$$

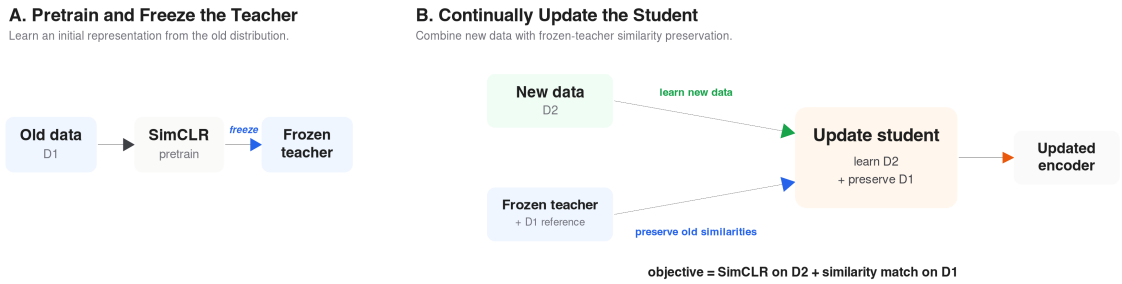

Figure 1. Similarity-preserving distillation for continual self-supervised learning. First, a teacher encoder is trained on the initial distribution D_1 and frozen. During continual training, the student learns from later data D_2 while matching the frozen teacher's similarity structure on reference examples from D_1.

where $B_2 \sim D_2$, $B_1 \sim D_1$, and λ controls the strength of the distillation penalty.

The first term adapts the encoder to the later data distribution. The second term discourages the encoder from changing the teacher's representation geometry on earlier data. This produces a simple stability-plasticity tradeoff: larger λ favors preserving the earlier representation, while smaller λ allows more adaptation to the later distribution.

3.5. Training Procedure

The full procedure is:

1. Train a SimCLR encoder on D_1.
2. Freeze the trained encoder as the teacher f_{θ_T}.
3. Initialize the student encoder f_θ from the teacher weights.
4. During continual training, sample a new-data batch B_2 from D_2 and a reference batch B_1 from D_1.
5. Update the student using the combined SimCLR and similarity-preserving distillation objective.
6. Use the updated encoder for downstream evaluation.

The method avoids full retraining on the union of old and new data. However, it is not fully data-free: the distillation term requires reference samples from the earlier distribution. In this work, those reference samples are used only to preserve teacher-defined representation relationships, while the self-supervised learning signal comes from the later distribution.

4. Datasets

We evaluate continual self-supervised learning in two settings: a temporal production-data setting based on Twitch stream frames, and a public distribution-shift setting based on ImageNet and CelebA. The Twitch data is the main setting of interest, while the public datasets provide a stronger semantic shift for validating whether the distillation objective reduces forgetting when the source and target distributions differ more substantially.

4.1. Temporal Twitch Stream Data

We collected two unlabeled datasets of Twitch stream frames from different time periods. The first dataset, denoted D_1, contains frames from January–June 2020. The second dataset, denoted D_2, contains frames from July–December 2021. Each dataset contains approximately 150,000 frames sampled from Twitch video-on-demand content.

This temporal split approximates a continual update setting: an encoder is first trained on an earlier slice of platform data and later adapted to a newer slice. The goal is not only to learn from the later data, but also to preserve representation structure learned from the earlier period. In the Twitch experiments, D_1 and D_2 are each split into 90% training and 10% test data for self-supervised evaluation.

Frames were sampled from stream metadata and processed into image datasets for SimCLR training. The same collection pipeline can be applied to later time periods, making the setup representative of a recurring production update process where new unlabeled data becomes available over time.

4.2. Downstream Evaluation Tasks

To evaluate whether the continually updated representations remain useful for practical downstream tasks, we use two labeled Twitch classification datasets.

The first task is **IP violation detection**, a binary classification task for distinguishing images with and without IP violations. The evaluation set contains 1,995 labeled examples, with 974 examples in the no-violation class and 1,021 examples in the violation class.

The second task is **game stream classification**, a 10-way classification task for predicting the game being streamed. The evaluation set contains 8,913 labeled examples across ten game categories: Apex Legends, COD Warzone, CSGO, DOTA 2, Fortnite, GTA 5, League of Legends, Minecraft, Resident Evil Village, and Valorant.

These downstream datasets are used to measure transfer after continual self-supervised training. In each case, the learned encoder is frozen and a lightweight MLP head is trained for the downstream task. This isolates the quality of the representation learned by the self-supervised encoder.

4.3. Public Distribution-Shift Validation

The Twitch temporal datasets may share substantial semantic content across time periods. To evaluate the method under a stronger distribution shift, we also use a public validation setting with ImageNet and CelebA. In this setting, ImageNet is used as the initial distribution D_1, and CelebA is used as the later distribution D_2.

We subsample ImageNet so that the two public datasets are roughly matched in scale, with approximately 200,000 images per dataset. Labels are not used for self-supervised training or continual evaluation. Instead, the public setting is used to measure whether continual training on CelebA degrades the original ImageNet self-supervised representation, as measured by SimCLR loss on the ImageNet test set.

This setting is not intended to replace the Twitch temporal evaluation. Rather, it provides a controlled stress test where the source and target data distributions are more semantically distinct, making catastrophic forgetting easier to observe.

4.4. Dataset Summary

Table 1. Summary of datasets used in the experiments.

Dataset	Role	Size	Labels
Twitch D_1	Initial SSL	$\sim$150k	No
Twitch D_2	Continual update	$\sim$150k	No
IP violation	Downstream binary	1,995	Yes
Game stream	Downstream 10-way	8,913	Yes
ImageNet subset	Public D_1	$\sim$200k	Unused
CelebA	Public D_2	$\sim$200k	Unused

5. Experimental Setup

All experiments use a ResNet-18 encoder trained with SimCLR on unlabeled data. We evaluate whether continual self-supervised training on a later distribution preserves representation quality on the earlier distribution and maintains downstream transfer performance.

The experiments are designed to measure representation preservation rather than state-of-the-art performance. In each setting, the encoder is first trained on an initial distribution D_1, then continually updated on a later distribution D_2. We compare regular continual training against the similarity-preserving distillation objective described in Equation (4).

5.1. Compared Methods

We compare two continual training procedures.

Regular continual training first trains a SimCLR encoder on the initial distribution D_1, initializes the continual model from that encoder, and then continues training on the later distribution D_2 using only the SimCLR objective.

Similarity-preserving distillation uses the same initial encoder and the same continual training data, but adds the pairwise similarity distillation penalty described in Section 3. The teacher is the frozen encoder trained on D_1, and the student is updated on D_2 while preserving the teacher's similarity structure on reference batches from D_1.

This comparison isolates the effect of the distillation term. Both methods start from the same D_1-trained encoder and adapt to the same D_2 data, but only the distillation method is explicitly constrained to preserve earlier representation geometry.

5.2. Unlabeled Retention Evaluation

The first evaluation measures whether continual training changes the representation learned from the original unlabeled distribution. We train an encoder on D_1, evaluate its SimCLR loss on the D_1 test set, continue training on D_2,

and then re-evaluate on the same D_1 test set.

Because this evaluation uses unlabeled data, we use SimCLR loss as a proxy for representation retention. Lower post-update loss on D_1 indicates that the updated encoder better preserves the contrastive structure of the original distribution under this metric.

5.3. Downstream Transfer Evaluation

The second evaluation measures whether continual training affects downstream task performance. After initial SimCLR training on D_1, we freeze the encoder, attach a lightweight MLP head, and train the head on a downstream classification task. We then continually train the encoder on D_2 using either regular continual training or similarity-preserving distillation. After continual training, we freeze the updated encoder, attach a new MLP head, and train and evaluate the head on the same downstream task.

This protocol evaluates whether the updated representation remains useful for labeled downstream tasks after adapting to the later unlabeled distribution. We report precision, recall, F1, and accuracy for the Twitch downstream tasks.

5.4. Public Distribution-Shift Evaluation

The third evaluation uses ImageNet as D_1 and CelebA as D_2 to test the method under a stronger semantic shift. We train a SimCLR encoder on ImageNet, evaluate its SimCLR loss on the ImageNet test set, continually train on CelebA using either regular continual training or similarity-preserving distillation, and then re-evaluate on the ImageNet test set.

This setting is used as a stress test for catastrophic forgetting. Since the source and target distributions are more distinct than the Twitch temporal slices, it provides a clearer measure of whether the distillation term preserves the original representation under distribution shift.

6. Results

Across the experiments, the main finding is that similarity-preserving distillation improves original-distribution retention, but the size of the benefit depends on the amount of distribution shift. On the Twitch temporal split, both regular continual training and distillation preserve downstream performance, suggesting limited measured forgetting for the tasks studied. Under the stronger ImageNet-to-CelebA shift, the benefit of distillation is clearer.

Table 2 summarizes the main empirical effects before the detailed task-level results.

Table 2. Summary of the main empirical effects of similarity-preserving distillation.

Setting	Metric	Distill. effect
Twitch D_1	SimCLR loss	-187.33 loss
IP violation	Accuracy	+0.01
Game stream	Accuracy	+0.01
ImageNet→CelebA	Loss increase	-9.3% loss increase

6.1. Original-Distribution Retention on Twitch

Table 3 reports SimCLR loss on the original Twitch D_1 test set before and after continual training on D_2. Lower loss indicates better preservation of the contrastive representation under this metric.

Table 3. Twitch D_1 SimCLR loss before and after continual training. Lower is better.

Method	Initial	After D_2	Change
Regular	46122.72	46085.47	-37.25
Distill.	46122.72	**45898.14**	**-224.58**

Both methods maintain the original-distribution objective after continual training, and neither shows an increase in D_1 loss. This suggests that the sampled Twitch temporal split does not expose severe forgetting under the SimCLR-loss metric. However, similarity-preserving distillation achieves a lower post-update loss than regular continual training, reducing the D_1 test loss by 224.58 compared with 37.25 for the regular baseline. This indicates that the distillation term better preserves, or reinforces, the original contrastive structure while the model adapts to later data.

6.2. Downstream Transfer After Temporal Update

Tables 4 and 5 report downstream classification performance after continual training. These experiments evaluate whether the updated encoder remains useful when frozen and used as a feature extractor for labeled Twitch tasks.

Table 4. Downstream IP violation detection performance after continual training. Method columns report precision/recall/F1.

Class	N	Regular P/R/F1	Distill. P/R/F1	ΔF1
No violation	974	.81/.88/.85	.88/.79/.84	-0.01
Violation	1021	.88/.81/.84	.82/.90/.86	+0.02
Accuracy	1995	–/–/.84	–/–/**.85**	+0.01
Macro avg.	1995	.85/.85/.84	.85/.85/**.85**	+0.01
Weighted avg.	1995	.85/.84/.84	.85/.85/**.85**	+0.01

On IP violation detection, the two methods perform similarly. Distillation improves overall accuracy from 0.84 to 0.85 and macro F1 from 0.84 to 0.85. The class-level effects are mixed: F1 decreases slightly for the no-violation class and improves for the violation class. This suggests a small aggregate benefit, but not a large downstream shift.

Table 5. Downstream game stream classification performance after continual training. Method columns report precision/recall/F1.

Class	N	Regular P/R/F1	Distill. P/R/F1	ΔF1
Apex Legends	871	.74/.78/.76	.75/.81/**.78**	+0.02
COD Warzone	891	.80/.69/.74	.77/.76/**.76**	+0.02
CSGO	941	.83/.76/.79	.77/.81/.79	0.00
DOTA 2	855	.91/.94/.92	.94/.94/**.94**	+0.02
Fortnite	894	.83/.75/.79	.83/.77/**.80**	+0.01
GTA 5	880	.82/.83/.82	.83/.83/**.83**	+0.01
LOL	815	.89/.89/.89	.90/.88/.89	0.00
Minecraft	929	.81/.87/.84	.83/.86/.84	0.00
REV	907	.77/.88/.83	.83/.84/.83	0.00
Valorant	930	.84/.85/**.85**	.87/.82/.84	-0.01
Accuracy	8913	–/–/.82	–/–/**.83**	+0.01
Macro avg.	8913	.82/.82/.82	.83/.83/**.83**	+0.01
Weighted avg.	8913	.82/.82/.82	.83/.83/**.83**	+0.01

The game classification results show the same pattern. Distillation improves overall accuracy from 0.82 to 0.83 and macro F1 from 0.82 to 0.83, with small gains for several classes and no change or slight degradation for others. Taken together, the downstream results indicate that the Twitch temporal update does not substantially harm transfer performance for the tasks studied. The benefit of distillation is positive but modest, consistent with the hypothesis that the sampled Twitch time periods share substantial semantic structure.

6.3. Stronger Distribution Shift: ImageNet to CelebA

The Twitch experiments evaluate a realistic temporal production setting, but the observed downstream forgetting is limited. To test the method under a stronger shift, we use ImageNet as D_1 and CelebA as D_2. Table 6 reports ImageNet test loss before and after continual training on CelebA.

Table 6. ImageNet test loss after continual training on CelebA. Lower is better.

Method	Initial	After D_2	Increase
Regular	31856.58	39943.75	8087.17
Distill.	31856.58	**39188.78**	**7332.20**

In this setting, continual training produces a clear increase in loss on the original distribution. Regular continual training increases ImageNet test loss by 8087.17, while similarity-preserving distillation increases it by 7332.20. This corresponds to a 9.3% relative reduction in the loss increase:

$$\frac{8087.17 - 7332.20}{8087.17} \approx 9.3\%. \tag{5}$$

This result supports the core motivation of the method: when the later distribution differs substantially from the earlier one, preserving teacher-defined similarity structure can reduce loss-based forgetting of the original self-supervised representation.

6.4. Summary

The results suggest three conclusions. First, on the Twitch temporal split, regular continual training does not cause severe measured forgetting under the metrics used here. Second, similarity-preserving distillation provides small but consistent aggregate improvements on Twitch retention and downstream transfer. Third, under a stronger ImageNet-to-CelebA shift, the same objective produces a clearer reduction in original-distribution loss increase. Overall, the benefit of knowledge distillation in continual self-supervised learning appears to depend strongly on the degree of distribution shift between the initial and later data.

7. Discussion

The results highlight an important property of continual self-supervised learning: the value of a forgetting-mitigation method depends on the amount and kind of distribution shift between the original and later data. Similarity-preserving distillation anchors the earlier representation, but its measured benefit is largest when the later distribution creates enough pressure to move the encoder away from its original structure.

7.1. Distribution Shift Determines the Observed Benefit

The Twitch temporal split represents a realistic production update setting, but it does not produce severe measured forgetting in the downstream tasks studied. Both regular continual training and similarity-preserving distillation maintain similar downstream accuracy and F1 after the update from D_1 to D_2. This suggests that the sampled Twitch time periods retain substantial shared structure for these tasks.

This is an important empirical point: temporal separation alone does not guarantee a hard continual learning problem. A dataset can be collected at different times while preserving enough semantic overlap that forgetting is limited. In contrast, the ImageNet-to-CelebA experiment acts as a stronger distribution-shift stress test; there, regular continual training causes a clear increase in original-distribution

loss, and distillation reduces that increase.

7.2. Why Preserve Relationships?

The method is motivated by the difference between supervised outputs and self-supervised representations. In supervised Learning Without Forgetting, the old model's behavior can be summarized by class-probability outputs. In self-supervised learning, there is no fixed label space and no supervised output distribution to preserve. The learned object is the representation space itself.

Preserving pairwise similarities is therefore a natural distillation target. The objective does not force the student to keep each embedding coordinate unchanged. Instead, it asks the student to preserve which examples are close or far under the teacher representation, allowing adaptation to new data while discouraging changes that would destroy earlier relational structure.

7.3. Implications for Deployed Representation Systems

For production representation systems, the results suggest that continual learning should be treated as an update policy rather than a fixed schedule. New unlabeled data may improve a shared encoder, but the need for distillation, replay, or full retraining depends on how much the new data shifts the representation and which downstream tasks are sensitive to that shift.

A practical system should therefore monitor both self-supervised retention metrics and downstream task performance. If new data is close to the old distribution, regular continual training may be sufficient. If new data introduces a stronger shift, a distillation objective can help preserve earlier representation structure without retraining from scratch on the full historical dataset.

8. Limitations and Future Work

This study has several limitations. First, the Twitch experiments use approximately 150,000 frames per time period, which is modest for contrastive self-supervised pretraining. Larger temporal slices, longer time gaps, and more update periods would provide a stronger test of continual representation learning. Second, the unlabeled retention metric is based on SimCLR loss on the original distribution. This is useful for measuring preservation of the contrastive objective, but it is only a proxy for downstream representation quality. Third, the downstream evaluation covers two Twitch tasks, so the results may not capture forgetting for tasks that are more sensitive to emerging or disappearing content categories.

The method also requires reference batches from the earlier distribution to compute the similarity-preserving distillation loss. This avoids full retraining on all historical data, but it is not fully data-free. Future work should study how much reference data is needed, whether compact exemplars or stored embeddings can provide the same benefit, and how the method compares against replay-based and predictor-based continual self-supervised learning methods. More broadly, continual update policies should be evaluated across stronger backbones, larger unlabeled datasets, additional downstream tasks, and multiple sequential updates rather than a single $D_1 \rightarrow D_2$ transition.

9. Impact Statement

This work studies methods for maintaining reusable representations as data distributions change over time. The main potential benefits are lower labeling costs, more efficient model updates, and better preservation of performance across temporal shifts. In deployed systems, however, representation updates can affect downstream applications unevenly across classes, tasks, or user communities. Continual learning methods should therefore be paired with monitoring for distribution shift, downstream regressions, bias, and unintended degradation before deployment.

10. Conclusion

This work studies continual self-supervised learning with knowledge distillation. Starting from a SimCLR encoder trained on an initial unlabeled distribution, we continually update the encoder on later data while preserving the pairwise similarity structure induced by the original model. This adapts distillation to a self-supervised setting where there are no supervised class logits to preserve, and where the relevant object is instead the geometry of the learned representation space.

Across temporal Twitch data, regular continual training and similarity-preserving distillation both maintain downstream transfer performance on the tasks studied, suggesting limited measured forgetting under the sampled time split. Under a stronger ImageNet-to-CelebA distribution shift, similarity-preserving distillation reduces the increase in original-distribution SimCLR loss by approximately 9% relative to regular continual training.

The main lesson is that continual self-supervised updates should be evaluated through both representation-retention metrics and downstream task performance. Distillation can help preserve earlier representation structure, but its benefit depends on the amount of shift between the original and later data. For deployed representation systems, this makes continual learning an update-policy problem as much as a modeling problem: the right strategy depends on measured drift, compute cost, and downstream risk.

References

Bardes, A., Ponce, J., and LeCun, Y. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2022.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments. In *Advances in Neural Information Processing Systems*, 2020.

Caron, M., Touvron, H., Misra, I., Jegou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. In *International Conference on Computer Vision*, 2021.

Cha, H., Lee, J., and Shin, J. Co2L: Contrastive continual learning. In *International Conference on Computer Vision*, 2021.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1597–1607, 2020.

Fini, E., da Costa, V. G. T., Alameda-Pineda, X., Ricci, E., Alahari, K., and Mairal, J. Self-supervised models are continual learners. In *Conference on Computer Vision and Pattern Recognition*, 2022.

Grill, J.-B., Strub, F., Altche, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z. D., Azar, M. G., Piot, B., Kavukcuoglu, K., Munos, R., and Valko, M. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, 2020.

He, K., Chen, X., Xie, S., Li, Y., Dollar, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *Conference on Computer Vision and Pattern Recognition*, 2022.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. In *NeurIPS Deep Learning and Representation Learning Workshop*, 2015.

Li, Z. and Hoiem, D. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2018.

Purushwalkam, S., Morgado, P., and Gupta, A. The challenges of continuous self-supervised learning. In *European Conference on Computer Vision*, 2022.

Tung, F. and Mori, G. Similarity-preserving knowledge distillation. In *International Conference on Computer Vision*, 2019.

Zbontar, J., Jing, L., Misra, I., LeCun, Y., and Deny, S. Barlow Twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, 2021.