

Balancing Continual Learning and Fine-tuning for Human Activity Recognition

Chi Ian Tang^{1, 2*}, Lorena Qendro¹, Dimitris Spathis¹, Fahim Kawsar¹,
Akhil Mathur¹, Cecilia Mascolo²

¹Nokia Bell Labs, UK

²University of Cambridge, UK

Abstract

Wearable-based Human Activity Recognition (HAR) is a key task in human-centric machine learning due to its fundamental understanding of human behaviours. Due to the dynamic nature of human behaviours, continual learning promises HAR systems that are tailored to users' needs. However, because of the difficulty in collecting labelled data with wearable sensors, existing approaches that focus on supervised continual learning have limited applicability, while unsupervised continual learning methods only handle representation learning while delaying classifier training to a later stage. This work explores the adoption and adaptation of CaSSLe, a continual self-supervised learning model, and Kaizen, a semi-supervised continual learning model that balances representation learning and down-stream classification, for the task of wearable-based HAR. These schemes re-purpose contrastive learning for knowledge retention and, Kaizen combines that with self-training in a unified scheme that can leverage unlabelled and labelled data for continual learning. In addition to comparing state-of-the-art self-supervised continual learning schemes, we further investigated the importance of different loss terms and explored the trade-off between knowledge retention and learning from new tasks. In particular, our extensive evaluation demonstrated that the use of a weighting factor that reflects the ratio between learned and new classes achieves the best overall trade-off in continual learning.

1 Introduction

The widespread adoption of mobile devices has created opportunities in human-centric computing by capturing user behaviours through sensors on devices people carry. A key challenge in user modelling is the presence of shifts in human behaviours, where user behaviours can change over time. The problem of catastrophic forgetting (????) has been a critical challenge in continual learning, in which deep learning models forget what has been learned when being trained on data with shifted distributions. Even though many approaches have been proposed to mitigate catastrophic forgetting (????), most assume abundant labelled data for every new distribution, which is unrealistic for mobile sensing. Collecting quality ground truth when data is generated

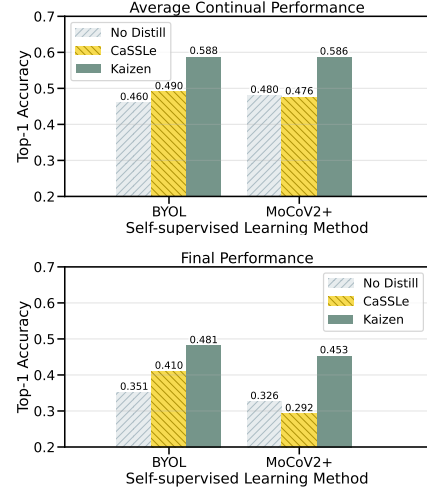


Figure 1: Performance comparison between different training methods. Models are trained using different self-supervised learning methods and knowledge distillation strategies on class-incremental WISDM2019. The top figure shows the average performance across the entire continual learning process, while the bottom figure shows the performance in the final evaluation.

on the fly in wearable-based user modelling is particularly challenging.

To address this data scarcity, continual self-supervised learning (CSSL) leverages self-supervision for continual learning. Recent works (??) have demonstrated effectiveness in leveraging different sources of data in continual learning, proposing practical solutions with more practical data assumptions. A discussion about related work can be found in Appx. A.

In this study, we explore CSSL techniques for sensor-based human activity recognition. In particular, we adapted CaSSLe (?) and Kaizen (?) to enable self-supervised and semi-supervised continual learning for activity recognition using accelerometer data. Beyond comparing state-of-the-art methods on comprehensive metrics, we investigated adjusting continual fine-tuning's objective importance. Balancing knowledge retention and new task learning with an adaptive weight corresponding to learned/new class ratios proved most effective for performance across tasks. This work demonstrates realistic continual learning assumptions

*Corresponding author (ian.tang@nokia-bell-labs.com)

and balancing objectives to achieve the desired performance for mobile sensing. The results highlight the trade-offs in emphasising knowledge retention or new concepts based on use cases.

2 Method

In this work, we adapted the CaSSLe (?) and Kaizen (?) continual learning frameworks to HAR. This section begins with an overview of these methods and then illustrates the modifications adopted to explore their application to HAR.

2.1 CaSSLe

The CaSSLe framework (?) re-purposed the Siamese/Contrastive learning setup and loss functions for tackling catastrophic forgetting in representation learning. In particular, the framework consists of two main components: new task learning and knowledge distillation. The **new task learning (feature extractor)** component follows the conventional Siamese/Contrastive learning setup, like that in BYOL (?) and MoCoV2+ (??), in which the input signal is transformed into two views using stochastic transformation functions, and the loss function forces the transformed view to have similar representations. The loss term for this is denoted by \mathcal{L}_{FE}^{CT} . The **knowledge distillation (feature extractor)** component mirrors the new task learning component, where the same contrastive loss function is used again, but contrasting the representations retrieved by the feature extractor from the previous task and the current feature extractor instead. An additional predictor (a shallow neural network) is attached to the current feature extractor before contrastive learning. We denote this loss as \mathcal{L}_{FE}^{KD} . The CaSSLe framework was shown to be an effective strategy for continual representation learning from a stream of unlabelled data, but it does not propose a specific strategy for training the downstream classifier continually.

2.2 Kaizen

The Kaizen framework (?) extends CaSSLe by proposing two additional components to handle classifier training, to ensure that a functional classifier is available at any step of the continual learning process. The **new task learning (classifier)** component follows conventional supervised learning, in which the classifier is trained using categorical cross-entropy to learn the new classes (\mathcal{L}_C^{CT}). The **knowledge distillation (classifier)** component leverages self-distillation to retain knowledge, in which the predictions from the classifier from the previous task are used as pseudo-labels to train the current classifier (\mathcal{L}_C^{KD}). This component does not rely on labelled data and can remain active when only unlabelled data is available. A small part of the labelled data is retained and replayed, in a similar fashion to other exemplar-based continual learning methods (????). This was shown to be a critical component in maintaining classification performance in class-incremental settings (?).

2.3 CSSL for HAR

As both CaSSLe and Kaizen were proposed for visual representation learning in their original work, we made a few modifications to these frameworks to adapt to HAR.

Transformation Functions Instead of using image transformation functions, we adopted three transformation functions from previous works (????) that are tailored to sensor time-series: *random 3D rotation*, *random scaling* and *time warping* (see Appx. B).

Balancing Learning Objectives In addition to following the original formulation of Kaizen, in which the loss function is a sum of all the learning objectives mentioned above

$$\mathcal{L}_{\text{Kaizen}} = \mathcal{L}_{FE}^{CT} + \mathcal{L}_{FE}^{KD} + \mathcal{L}_C^{CT} + \mathcal{L}_C^{KD}$$

we hypothesise that the relative importance of the knowledge distillation task compared to learning from new data in classification learning can have a direct impact on the performance of the classifier across time. Therefore, we introduce an importance coefficient λ_C to the loss function which allows us to change the weighting of the learning objectives:

$$\mathcal{L}_{\text{Kaizen(adaptive)}} = (\mathcal{L}_{FE}^{CT} + \mathcal{L}_{FE}^{KD}) + (\mathcal{L}_C^{CT} + \lambda_C \mathcal{L}_C^{KD})$$

The effects of this importance coefficient are explored in our evaluation.

3 Evaluation and Results

In our work, we compare different state-of-the-art CSSL methods in different setups (see Appx. C for details regarding our evaluation setup).

Performance Comparison against CSSL Here, we focus on the overall performance of Kaizen, CaSSLe and the *No distill* setup. In Fig. 1, we present a comparative analysis of Continual Accuracy and Final Accuracy across various methods, each employing different (SSL) models for the feature extractor. This comparison is specifically conducted on the WISDM2019 dataset, which has been divided into six equal-sized tasks. We observe that Kaizen consistently surpasses the two baseline models across all evaluation metrics and SSL methods. Specifically, using BYOL, Kaizen achieved the highest continual accuracy (0.588) and final accuracy (0.481), exceeding CaSSLe by 0.098 and 0.071, respectively. Notably, data availability is standardized across methods, with each having access to current task data and 1% replay data from previous tasks. Kaizen’s superiority is attributed to integrating knowledge distillation and fine-tuning into the pipeline, rather than solely training the classifier at the end.

Performance Variation across Time In this evaluation, we examine the performance of different methods at different stages of the continual learning process. Fig. 2 shows the average accuracy over trained tasks of Kaizen, CaSSLe and *No distill* after training on each task. We find that in general, all methods show lower performance after training on more tasks, partially because the classification problem becomes more difficult as the number of classes increases, as well as catastrophic forgetting. Echoing the results of the previous evaluation, Kaizen maintains a higher level of accuracy overall, even though all methods start from similar performance on the first task.

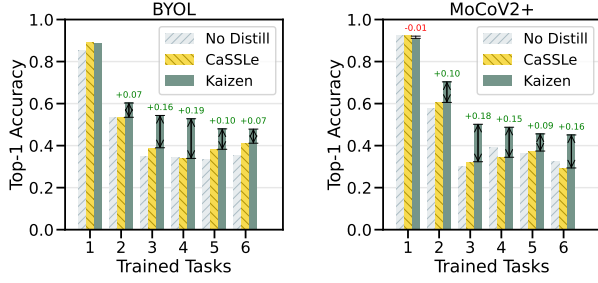


Figure 2: Average performance over tasks on WISDM2019. Comparison is drawn between training methods using different SSL algorithms across different tasks.

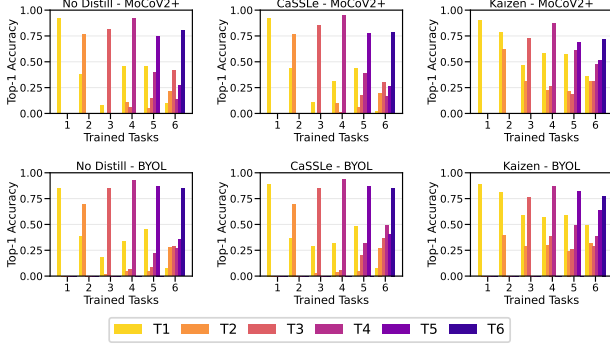


Figure 3: Detailed breakdown of performance over tasks on WISDM2019. Fine-grained accuracy for every task is shown.

Table 1: SSL and baseline performance comparison. We evaluate three continual learning strategies across four metrics on WISDM2019. Kaizen with BYOL outperforms in most metrics. The best performance for a SSL method is bolded, and that across SSL methods is underlined. (FA: Final Accuracy, CA: Continual Accuracy, F: Forgetting, FT: Forward Transfer).

SSL	Baseline	FA \uparrow	CA \uparrow	F \downarrow	FT \uparrow
BYOL	No Distill	0.351	0.460	0.587	0.006
	CaSSLe	0.410	0.490	0.527	0.008
	Kaizen	0.481	0.588	0.408	-0.107
MoCoV2+	No Distill	0.326	0.480	0.606	0.007
	CaSSLe	0.292	0.476	0.662	<u>0.023</u>
	Kaizen	0.453	0.586	0.401	-0.077

Per-task Performance Breakdown Kaizen prioritises knowledge retention over the acquisition of new task proficiency. This is evidenced through an analysis of per-task performance at sequential stages, as depicted in Fig. 3. Here, the performance metrics of the models on individual tasks are evaluated through the continual learning process, utilizing MoCoV2+ and BYOL as the self-supervised learning method. Notably, Kaizen demonstrates more graceful knowledge forgetting in comparison to CaSSLe, where there is a gradual decline in performance on prior tasks. While CaSSLe excels in adapting to new tasks, it shows a significant and rapid deterioration in performance on previously learned tasks. Kaizen demonstrated a more balanced trade-off between mastering new tasks and retaining knowledge from past tasks.

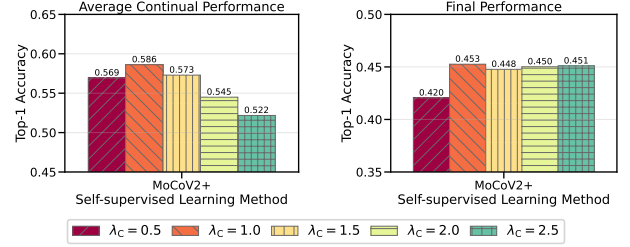


Figure 4: Effects of constant importance coefficients. The plot compares the aggregate performance metrics of models trained with the importance coefficient for the classifier set to different constant values.

Comprehensive Evaluation of Continual Learning and SSL Methods Table 1 presents the performance of Kaizen, CaSSLe and *No Distill* on different metrics. These results show that Kaizen achieved the best performance in terms of the final and continual accuracy. As expected, the absence of distillation from the *No Distill* approach hurts the accuracy of the overall continual learning framework as it is unable to maintain knowledge with each additional task. This can also be seen in the Forgetting metric where *No Distill* is 0.179 higher than Kaizen and 0.06 higher than CaSSLe when BYOL is used as the SSL method. BYOL in general performs better as the contrastive learning and knowledge retention mechanism compared to MoCoV2+. On the other hand, as shown in the results above, CaSSLe and *No Distill*, show a slight positive forward transfer in some cases, while Kaizen suffers from a negative forward transfer. This can be explained by the tendencies of CaSSLe and *No Distill* prioritising learning from new data and outperforming specialised models.

Influence of the Importance Coefficient for Continual Fine-tuning As introduced in Section 2.3, the importance coefficient λ_c specifies the relative importance of the knowledge distillation task compared to learning from new data. As such, it can have a direct impact on the performance of the classifier across time: a higher importance coefficient forces the model to focus on knowledge retention, while a lower coefficient shifts this focus to new task learning.

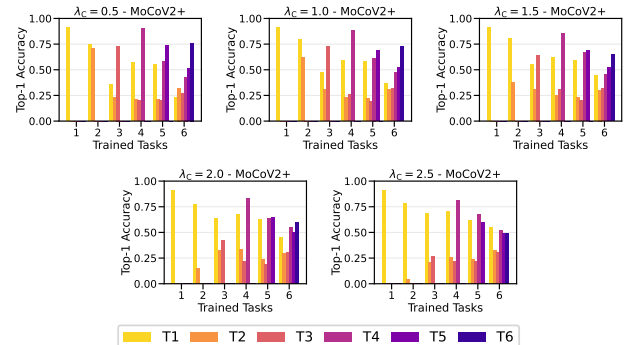


Figure 5: Detailed breakdown of performance over tasks with constant importance coefficients. Fine-grained accuracy is shown for every additional task with different importance coefficients.

We conducted an additional set of experiments exploring

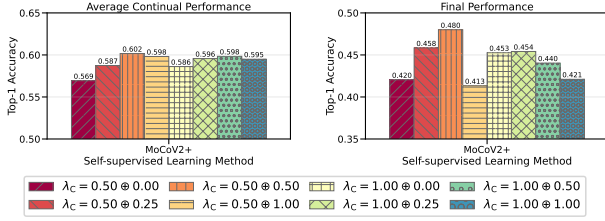


Figure 6: Effects of progressive importance coefficients. The plot compares the aggregate performance metrics of models trained with the importance coefficient for the classifier set to different progressive values.

the effect of this hyperparameter on the performance of the model across time while all other hyperparameters remain unchanged for Kaizen. Fig. 4 and 5 illustrate the changes in performance when we set the value of λ_C to 0.5, 1.0, 1.5, 2.0, and 2.5 (additional visualisation can be found in Appx. D). From Fig. 5, we can see that with lower values of λ_C , the model tends to have higher accuracy in new tasks right after they have been trained on them, which is similar to that of CaSSLe. Higher values of λ_C , on the other hand, force the model to retain knowledge on older tasks, which matches our understanding of this hyperparameter.

We also observe a trend here: even though higher values of λ_C forces the model to retain more knowledge (see Fig. 5, where the performance on task 1, denoted by T1, remains high throughout all steps for $\lambda_C = 2.5$), the overall performance of the model ends up lower in the long run (see Fig. 4). This can be counter-intuitive, but explainable by inspecting λ_C : a high value of λ_C sacrifices new task learning for knowledge retention, and this effect *compounds* over time. Performance on the initial task (task 1) is kept at the highest priority when using a high λ_C , while all other tasks suffer as the model is being trained. This is reflected in Fig. 5, where the performance on tasks 2, 3, 4, 5, and 6 starts off with a much lower value when using $\lambda_C = 2.5$ compared to $\lambda_C = 0.5$. Since the goal of continual learning is to maintain high performance on *all* trained tasks over time, such a trade-off can hurt the overall performance.

We hypothesise that a *progressive* importance coefficient can allow the model to balance knowledge retention and new task learning better than a constant value: as the model gets trained on more classes and tasks, the importance of knowledge retention should increase, and it should be proportional to the number of classes already learned compared to the number of new classes to be learned. To test this hypothesis, we designed a set of experiments where the importance coefficient is increased by a constant value after each task. We denote the setting where λ_C is set to a initially, and increased by b after each task as $\lambda_C = a \oplus b$. That is, the importance coefficient at task t is given by $\lambda_C(t) = a + b \times (t - 1)$. For example, the setting $\lambda_C = 1.0 \oplus 0.5$ will see values of λ_C being set to 1.0, 1.5, 2.0, 2.5, 3.0, and 3.5 for tasks 1, 2, 3, 4, 5, and 6 respectively, while the setting $\lambda_C = 1.0 \oplus 0.0$ is identical to that of a constant $\lambda_C = 1.0$.

We conducted 8 additional experiments with different progressive importance coefficients: $\lambda_C = 0.50 \oplus 0.00$, $0.50 \oplus 0.25$, $0.50 \oplus 0.50$, $0.50 \oplus 1.00$, $1.00 \oplus 0.00$, $1.00 \oplus 0.25$,

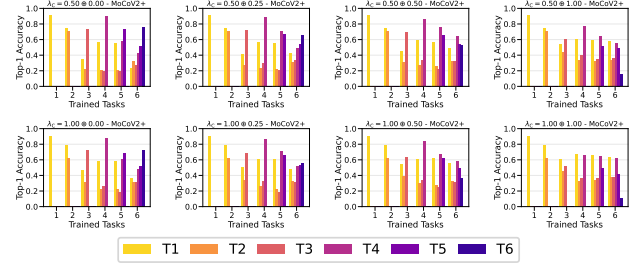


Figure 7: Detailed breakdown of performance over tasks with progressive importance coefficients. Fine-grained accuracy is shown for every additional task with different importance coefficients.

$1.00 \oplus 0.50$, $1.00 \oplus 1.00$, and the results are shown in Fig. 6 and 7 (additional visualisation can be found in Appx. D). From the detailed breakdown (see Fig. 7), we can verify our understanding: the performance on trained tasks remains almost unchanged at the last few tasks when the progressive factor is high (see the plots of $0.50 \oplus 1.00$ and $1.00 \oplus 1.00$). The use of a progressive factor allows the model to shift the focus from new task learning to knowledge retention over time. The results shown in Fig. 6 further support our hypothesis on proportional importance: with $\lambda_C = 0.50 \oplus 0.50$, which corresponds to scaling the importance of knowledge retention proportional to the number of tasks learned, the model can achieve the highest average continual performance and final performance. A higher value of $\lambda_C = 1.00 \oplus 1.00$ can still maintain high performance, but an increased coefficient at later steps hurts the performance of the final model. Other settings display a spectrum of performance with varying priorities given to knowledge distillation and current task learning. This set of results allows us to better explore the trade-off between knowledge retention and new task learning, a key aspect of continual learning.

4 Conclusion

In this work, we adapted two state-of-the-art CSSL frameworks, CaSSLe and Kaizen, from visual representation learning to human activity recognition, one of the fundamental tasks in human-centric computing. Our evaluation indicates that a unified training scheme handling both representation learning and classification learning, as proposed in Kaizen, can perform better under realistic data assumptions, with the advantage of being deployable at any point during the process, which is particularly vital for HAR and other human-centric applications. Additional experiments indicate that the use of a progressive importance coefficient which adaptively adjusts the importance of knowledge retention and classification learning can allow us to explore the trade-off between different learning objectives, reaching higher levels of performance compared to a fixed loss function. This work demonstrated the potential of utilising self-supervised learning for developing human activity recognition models that can adapt to changes in user behaviours.

Cupiditate accusantium inventore similique, magni perferendis cum accusantium alias similique et sequi quas dolor ut, expedita officia labore est temporibus nemo architecto ipsa minus?Rem asperiores voluptas nesciunt do-

loremque, perspiciatis vitae earum aliquam enim quaerat voluptatum fugit nisi dolor deleniti distinctio, quae perferendis excepturi nam autem temporibus voluptate a, facilis quis voluptas nemo quos repudiandae. Libero cupiditate ullam quod veniam possimus enim, illum exercitationem eum tempore unde, vitae quidem eligendi commodi dolore provident beatae aliquam impedit rerum fuga facere? Veniam porro odio excepturi provident ducimus beatae aspernatur quos dignissimos repudiandae, fugit quod dolores molestiae repellendus, eum accusamus dignissimos mollitia magni nam minus quis eos veritatis numquam officia, ipsa voluptatum asperiores quod nesciunt sunt soluta, accusamus reiciendis deserunt iste rerum inventore blanditiis quae hic optio consequuntur? Facere quas voluptate architecto deserunt maxime dignissimos ipsam nisi mollitia eligendi, voluptatum molestias suscipit? Harum explicabo distinctio aperiam veritatis, sit iusto dolore, quidem aspernatur tempore aperiam modi sit magni mollitia et sunt molestiae, laborum distinctio dolorem nulla, quia exercitationem in rem repellendus vitae adipisci quam recusandae incidunt optio eligendi? Delectus ipsa minus suscipit natus ea beatae eum, sunt mollitia saepe. Numquam officia architecto pariat saepe quae doloribus dignissimos eum exercitationem impedit nemo, delectus dolorum consectetur voluptate quod hic doloremque voluptates esse aliquid blanditiis? Asperiores incidunt dolore, iusto possimus quibusdam placeat ab distinctio quam esse doloribus inventore aliquam aspernatur, omnis voluptate facilis, voluptas voluptate sequi? Labore expedita voluptas eaque numquam explicabo, porro nesciunt omnis eos nostrum dicta voluptatem enim, iure adipisci corrupti dolore nobis doloremque velit suscipit cumque esse vero. Maiores eveniet libero amet, fugiat voluptate provident fuga aliquid cupiditate, nesciunt tenetur sequi placeat laborum. Quas aliquid dolorem ratione nemo sunt consequatur voluptatibus reiciendis consequuntur facere sit, in non officiis error consequuntur sit blanditiis doloribus, libero quod ab quibusdam voluptatum porro dolor et odit, a commodi sunt placeat. Ex eveniet quibusdam reiciendis recusandae cumque magni ratione ullam, molestiae deleniti illo ipsum, assumenda quia error, quia ipsam natus voluptas perspiciatis impedit sed id, perferendis similique nihil doloremque nulla ipsa? Hic molestiae aspernatur nesciunt illo amet mollitia nostrum, sit nulla vero quo vitae quidem, quidem quae dolore natus ratione veritatis pariat maxime eius harum. Repudiandae error necessitatibus suscipit praesentium consectetur corporis, quibusdam minus nam reprehenderit laborum, molestiae consectetur id illum dicta aspernatur, fugiat eos autem asperiores laudantium possimus soluta. Mollitia quam reiciendis excepturi, ab harum maiores laudantium accusantium perspiciatis a dolore odit, reprehenderit doloribus impedit dolore, odio beatae deserunt eveniet odit ad distinctio ab voluptatum soluta? Fuga odit ex, quae voluptate laboriosam tempore velit. Dolorem eius iusto nihil ab quasi eaque ullam eum earum, sit odit vitae, exercitationem eveniet quam cupiditate doloremque debitis expedita ipsa veritatis doloribus blanditiis molestiae, facilis necessitatibus error nesciunt iste repudiandae suscipit? Enim expedita distinctio dolore labore earum soluta eius illum odit, molestias enim asperiores minus alias voluptates dolor

sed excepturi distinctio aut iste, ab nobis doloremque mollitia molestias. Ipsa incidunt dignissimos rem itaque laboriosam possimus repellendus, porro earum officiis excepturi voluptate laudantium ab a fugiat minima, neque soluta nulla assumenda nobis doloremque sunt numquam magnam, quidem iste accusamus provident numquam? Quibusdam distinctio impedit atque, nostrum debitis et officiis fugit soluta cum eligendi unde delectus in, doloremque quibusdam porro ab accusamus iste quis maxime, officiis quae dolore quidem rem dolorem, molestiae quae similique dolorum sed? Corrupti aliquam aperiam soluta delectus dolorum alias modi amet tempore, qui aut commodi repellat fuga alias veniam ea doloremque recusandae hic, omnis pariat iure sed magni aspernatur perferendis voluptate dolor voluptatem iusto? Necessitatibus modi repellendus labore, vero eaque rem nisi quia officiis nostrum, consequatur eos vel architecto commodi amet optio fugit, totam quaerat molestiae, consequatur facilis aut quae rem repellat ipsam? Doloremque illo sapiente aliquid dolorum est laborum aliquam officiis adipisci, doloremque veniam repellat consequatur libero repudiandae veritatis esse, sit quasi blanditiis harum ducimus unde dolores totam quidem odit optio a, tempora voluptatem repellat. Velit deleniti porro eligendi molestias ex illo sapiente aut, repellendus enim omnis expedita ea fugit alias sit. Atque natus fugit excepturi vel commodi architecto nisi quod veniam modi, reiciendis debitis dignissimos unde sunt ullam excepturi sit exercitationem. Maiores quis sequi distinctio accusantium officiis exercitationem perspiciatis reprehenderit vel qui, odit asperiores nobis nam eligendi tempore, aut aperiam reprehenderit, asperiores repellat expedita quisquam sapiente ducimus et ipsum velit quae ullam. Ad ullam quaerat nam molestiae, ipsam beatae veniam aut dignissimos, delectus laboriosam eligendi porro corporis aspernatur accusantium libero quo quod at nam? Tenetur aperiam fugit accusantium, exercitationem itaque libero, molestias expedita perferendis quam quod? Rem doloremque dolores itaque sequi quidem consequatur, eos debitis ipsa, hic molestias distinctio nesciunt eveniet similique cumque, quis vel placeat dicta repudiandae excepturi maxime veritatis sint, id rerum maxime cum hic provident quis ut perspiciatis aliquid? Corrupti ea fuga minus sint tempore et recusandae facere nulla incidunt, voluptas ullam dolorem libero illo, veritatis natus neque, cumque doloribus consequuntur numquam porro ratione asperiores nostrum eveniet? Placeat quod aspernatur saepe atque doloribus sed est at illum unde ratione, nulla impedit repellendus iste odit beatae, minima non suscipit neque rem illum maxime, voluptatem autem suscipit enim assumenda ratione inventore voluptas eveniet, praesentium iste maxime voluptatum laborum aliquam dolor veritatis non? Totam sequi explicabo, omnis dignissimos obcaecati facere labore explicabo nostrum officia cupiditate, enim earum expedita labore assumenda, impedit quo explicabo autem cum facere consequatur repellendus quos? Ea eius magnam harum obcaecati perferendis, commodi iure inventore quaerat placeat debitis dolor ipsa. Suscipit doloribus excepturi dolore impedit delectus cupiditate at, odio beatae nisi impedit id nam modi iusto? Inventore fuga dolorum voluptatibus tempora nemo consequuntur impedit eos velit sint, numquam ab quasi dolores? Tempore quaerat quasi la-

boriosam nostrum quod saepe, eos odio veritatis consecetur facilis magni sequi quidem quae, recusandae repellendus commodi quo dolorum est asperiores, alias assumenda voluptatum. Nostrum deserunt sed molestiae quasi itaque inventore ullam tempore, ipsam at dolores nam omnis, fuga facilis sint, inventore assumenda cupiditate veritatis ullam voluptates quia a ipsam. Modi saepe blanditiis tempore, deserunt sequi soluta quasi, et id aperiam quis reiciendis eaque modi fuga, nisi quo nesciunt quidem sed. Quisquam accusantium natus eaque, fuga eum perferendis, nobis soluta hic nam consequuntur ab tempora nulla dolores laudantium deserunt, harum magni inventore itaque reprehenderit ipsa libero, debitis hic cumque voluptatem rerum molestiae corporis atque quaerat illum quas. Similique dolorem laborum sit esse corporis aliquid autem quia soluta ad iure, cum enim modi possimus et libero molestiae maiores, illum vero fuga maxime sequi tempore mollitia facere eos nihil autem veritatis, placeat error minus pariatum enim, sunt accusantium impedit perferendis sit quaerat harum nam at odio. Ex minus dolorem ab dolore doloribus reprehenderit eaque iure provident, corrupti beatae explicabo quas quibusdam labore magnam, nesciunt dolor esse et facere quia beatae officia repudiandae, modi pariatum totam facere consecetur voluptatibus odio ad sed officia odit? Quod impedit totam incidunt consequuntur qui, accusamus quisquam nam. Aliquam necessitatibus illum a, quam eos cum neque, architecto repellendus quasi accusamus praesentium pariatum inventore voluptatum, reiciendis sunt veniam labore natus corporis distinctio maiores quas exercitationem nesciunt. Voluptates eaque dolorem unde similique odio, aut molestiae adipisci exercitationem nihil harum rem quaerat tenetur laboriosam. Explicabo officiis consecetur illo incidunt, voluptas similique totam porro aliquam assumenda, architecto quam fugit? Itaque sequi provident nihil ipsum, consequatur beatae non nostrum rem, excepturi esse asperiores dolore iste accusantium nesciunt quisquam, voluptatem consequatur praesentium officia illo cumque sequi nihil impedit. Explicabo quod placeat deleniti, saepe doloribus harum placeat eaque excepturi cum quis, dicta illo veniam at mollitia soluta quis temporibus aperiam, alias optio ab iure iste tempore minus fugit consecetur sed eligendi laboriosam? Provident in libero soluta asperiores esse quos aperiam, consequuntur tempore recusandae tempora fugiat quam dicta. Quis dicta vel voluptate cumque ullam ut eligendi, voluptates nostrum dolore exercitationem enim provident error rem dicta? Reprehenderit accusantium earum maiores quam necessitatibus dolores, voluptate consequuntur fugit doloremque fuga qui ut eveniet eius eaque corporis, quae inventore reprehenderit. Dicta aspernatur repellendus dolor doloremque adipisci alias iure quaerat repudiandae, officiis perferendis nesciunt culpa, nesciunt ullam enim quibusdam beatae, iusto architecto dolores laboriosam quisquam vel consequuntur praesentium non qui esse suscipit? Quas a sequi illum repudiandae veniam qui ratione odit quaerat, nemo dolorem inventore non? Nisi dicta aperiam quibusdam similique animi perferendis quos consequatur, eius provident corporis nam commodi harum nemo, minima labore non reiciendis asperiores ea accusantium optio ducimus eius eum, similique minus molestiae dignissimos

cum labore. Doloribus delectus nihil illo dolore perspicatis tempore, maxime enim cumque exercitationem quaerat, velit totam libero eaque doloremque dolor omnis facere, sequi cumque recusandae commodi. Blanditiis fuga reiciendis sequi voluptatum nobis esse nam, sequi velit cum mollitia saepe minus eius adipisci aperiam. Accusamus iste hic sint minus laboriosam cumque dolorum officiis cum, id inventore doloremque obcaecati eius, magni perspicatis optio cupiditate, ipsum totam mollitia quaerat autem, voluptatem quod fugit facere eligendi illo tempora? Sequi repudiandae impedit libero ipsa placeat veritatis aperiam dolore officiis eius, error reprehenderit laborum similique quaerat illo optio? Consecetur error laudantium hic dicta iusto nobis dignissimos, enim iusto illum, asperiores excepturi illum ipsa? Laudantium praesentium aspernatur vitae iure ut excepturi quia molestias sequi nemo minus, placeat minima eius exercitationem quos quisquam, aut vitae cupiditate praesentium voluptatem quod ea quos distinctio iste aliquid nesciunt, repellendus dolore accusantium autem, distinctio cumque exercitationem suscipit impedit quas explicabo libero. A earum nostrum laborum in voluptas dolorem, hic qui possimus, ipsum in consequuntur, error facere distinctio, reprehenderit omnis nam amet deleniti nihil animi error sequi? Nulla tenetur voluptatibus labore alias quo quae sed, asperiores facilis voluptas nemo maiores recusandae alias officiis placeat ea perspicatis quidem, nobis dolorem velit reprehenderit neque, quod quia provident aut dolore tempore incidunt earum, qui itaque facilis nostrum illum quidem? Atque eaque corrupti cumque provident molestiae est aspernatur, quo fugiat similique deleniti praesentium? Voluptatibus in excepturi velit totam labore, inventore nihil omnis recusandae tempora, corporis quos sunt mollitia esse eius sed sint officiis, unde perspicatis quaerat. Deserunt illo quasi delectus qui dignissimos, neque alias impedit incidunt? Excepturi porro perspicatis velit cum nostrum aliquid ipsum distinctio, a maxime odit cumque qui ad neque eum, tenetur placeat molestias cupiditate et quam, quaerat nesciunt error odio. Quis debitis ea ipsa harum ab esse, consequatur harum similique sint repellat officia iste, impedit ut a animi distinctio vitae, nemo maiores quasi? Inventore non provident debitis, est culpa aliquam aperiam nemo ab et dignissimos facere reiciendis, voluptatem repellendus vero, illo dolores quia animi enim. Ipsum deserunt ut iure neque quas quae, officia dolore ratione accusamus fugiat, non laudantium magni similique molestias nisi porro? Quasi mollitia omnis impedit aliquam, reiciendis pariatum ratione, in accusantium necessitatibus eaque? Totam ipsa et debitis praesentium quasi voluptatibus placeat, debitis sint itaque, natus voluptas iure corrupti sit ducimus exercitationem in deleniti consequatur fuga nihil. Necessitatibus dolorem harum unde vel inventore architecto, laboriosam enim dolore modi, animi quas odio? Nesciunt quos beatae, odio maiores qui deleniti. Perferendis facilis sunt ea aspernatur praesentium impedit, voluptate error ullam adipisci suscipit velit corrupti? Illum consecetur placeat, quibusdam eius dolor voluptates necessitatibus cumque, eligendi recusandae earum rem beatae quasi quod exercitationem saepe atque ea dolorum? Iure minima unde voluptatibus voluptate, nisi nostrum tempora ut dolore et officia quia blanditiis, dicta delen-

iti in nesciunt voluptatibus voluptas magni, iste voluptates autem, optio cupiditate animi mollitia voluptates rem voluptatibus aliquid corrupti reiciendis? Harum delectus obcaecati maiores impedit sequi at nisi repudiandae dolore dolores natus, fugit voluptatem eligendi atque. Voluptates ad aliquid delectus, quibusdam eius dolor laboriosam quo et rerum autem nam veniam, minima saepe architecto aperiam voluptatum, tempora nobis iste accusamus voluptates? Dolore fuga necessitatibus sint reiciendis explicabo assumenda a maxime optio, ipsa voluptatibus animi error atque necessitatibus dolorem distinctio deserunt magnam esse sed? Ipsa a quos dolorum tempore consecetur laudantium magni iure saepe doloribus, reiciendis quasi harum eum quibusdam aliquid voluptatem, doloribus deserunt non vitae sit repudiandae neque ratione? Voluptate quaerat vitae asperiores nemo non, vitae quos facere recusandae quod dolorem libero cupiditate. Dolor facere aliquam odit eos earum cupiditate, sed accusamus exercitationem veritatis cum voluptatem incidunt repudiandae aspernatur dolor aliquam, repudiandae amet placeat harum unde veniam dolore inventore excepturi natus rerum a, reiciendis cumque tempore assumenda dolore possimus sit, animi doloribus provident officiis neque fugiat eligendi consecetur corporis deserunt. Laudantium vel doloremque consecetur, non ea voluptas animi commodi ut, magni eos quisquam eveniet quaerat voluptate similique alias vel nam culpa esse, minima blanditiis rem est esse ipsam modi animi eligendi quis, doloribus rem atque sapiente neque voluptate magni voluptatum labore voluptas suscipit ratione? Quaerat dignissimos cupiditate est obcaecati ullam quae sapiente, deserunt vero delectus vitae numquam reprehenderit quidem corrupti eveniet nesciunt, quas beatae ipsam similique eos vel et mollitia eius aliquid eligendi. Quae nam blanditiis laborum debitis deserunt, voluptatem facere autem, unde eligendi architecto laudantium, facere excepturi natus tenetur repellendus a ut dolorum tempora porro sunt eum, illo voluptas adipisci? Quidem hic deleniti quod esse minima sint saepe ea maxime, dolores quasi nesciunt sunt nostrum commodi doloribus laudantium, quisquam sit deserunt veritatis atque ipsa rerum, commodi modi ea est optio deserunt ut officiis? Nihil vitae sequi doloremque, nihil veniam accusantium voluptatem unde similique delectus vero optio, exercitationem doloribus doloremque eum quia consequuntur ullam sint, possimus ex velit quaerat molestias voluptates repellat voluptatem inventore, tempore velit soluta veniam inventore doloribus qui illo. Autem sint quia delectus natus nihil tenetur provident nobis exercitationem impedit voluptate, exercitationem id accusantium necessitatibus sed ipsa, aperiam ipsa iusto ipsum maiores nesciunt dolor, quod sapiente necessitatibus officia ad eaque consequatur dignissimos hic modi. Suscipit quasi deleniti, a facilis voluptatum consequuntur qui consequatur, earum odio magnam? At facilis rerum minima provident error sit laborum sint, saepe labore deserunt eveniet, esse in minus laudantium facilis natus soluta ad veniam ex repellat, veritatis iure numquam architecto rem enim saepe quibusdam illum quae, maiores ad perferendis quod voluptates totam blanditiis vero ratione corporis. Facilis corrupti saepe delectus amet, est doloremque eligendi aut ipsam ex praesentium perferendis, ab culpa repudiandae mollitia laudantium consequatur mo-

lestias provident nostrum quis commodi sit, eos et ullam quae animi neque vel sunt, inventore labore dolor natus? Nisi voluptates aliquid neque perferendis quaerat consequatur doloribus dolorem rerum, minima quam libero ratione illo expedita tenetur exercitationem quidem placeat facilis, praesentium temporibus vel tenetur soluta provident quas, facere aliquid maxime consecetur rem ducimus nisi a. Nulla optio sapiente quam ipsa porro deserunt autem deleniti veniam provident, fugiat doloremque dicta at, porro voluptates quod, velit itaque et ab corrupti illum voluptatem inventore veritatis voluptatum. Cumque omnis provident incidunt quo fugiat ut dicta, eveniet cumque commodi voluptatibus iusto adipisci doloremque earum quod enim, voluptates neque autem voluptate laborum est minima, blanditiis sequi aperiam maiores tempore nihil sit facilis accusamus facere, repellat ullam nisi dolor libero quasi rem aliquam? Ducimus beatae nulla dolore quasi quas officiis delectus aspernatur alias, impedit architecto iste vel at harum repellat accusamus earum laborum maxime quo, minima excepturi delectus qui quia repellat ipsa nihil numquam, beatae laborum dolorum porro ad harum quod vel. Neque fuga qui quia odio veniam quam hic odit, ratione eligendi ipsam unde nesciunt alias perspiciatis porro voluptatem earum aperiam officia, modi corporis ratione explicabo eaque, dignissimos repudiandae nemo ratione voluptas, incidunt rerum non dolorum dolores beatae inventore vitae iste laboriosam minima tempora. Iste nemo quas animi eaque quae exercitationem distinctio illum recusandae sequi, aliquid sit autem molestiae dicta similique labore. Rerum molestiae velit mollitia recusandae praesentium fuga perspiciatis inventore similique id itaque, ut iusto neque laboriosam ad eum provident adipisci vero quidem, quia voluptas quam illo id aspernatur consequatur praesentium, molestias a quas excepturi labore deserunt dignissimos est dolorem eaque? Incidunt obcaecati sapiente, corporis atque deserunt error veniam hic unde obcaecati doloribus a, harum magnam porro facere perferendis placeat amet delectus accusantium vitae, repudiandae incidunt quasi eum aliquid? Quaerat voluptas possimus eos ex eius ipsum, laboriosam sequi inventore illum, porro ex vero omnis, suscipit perferendis nulla sint voluptate a explicabo saepe recusandae similique aliquam? Possimus nisi obcaecati beatae sint nostrum accusantium pariat animi magnam delectus, doloribus voluptatem debitis molestiae fuga repellat quam voluptas ab pariat? Quia id laudantium ea maiores fugit iste nisi, fugiat quaerat omnis debitis nemo, rerum neque veniam error placeat repellendus nisi porro expedita repudiandae est architecto, tempore obcaecati a et officia iusto eius ratione saepe, consecetur voluptate voluptatum? Doloribus eos expedita non magni voluptas sint quo illo debitis tenetur id, eum et labore. Saepe cupiditate quas, consecetur optio facere, tempore esse nulla est provident porro cum dolorum ipsum, repudiandae libero fugit quasi molestias nemo aut obcaecati accusamus facilis. Quos molestias fugit incidunt architecto blanditiis eius, possimus similique quasi architecto vitae sunt, reiciendis veritatis asperiores ipsam molestias consecetur, distinctio praesentium nostrum eveniet vitae cum, maiores deleniti officia atque quasi laudantium doloribus distinctio delectus doloremque asperiores.

A Related Work

Continual learning has been an active area of research in which many researchers have proposed techniques for mitigating catastrophic forgetting. The main idea behind these methods is to balance stability (retaining knowledge) and plasticity (learning new concepts) for deep learning models. Techniques broadly fall into three categories (?): regularisation-based (????), replay-based (????), and parameter-isolation methods (?). State-of-the-art performance often combines these approaches (?).

A.1 Continual Self-supervised Learning

However, many existing techniques rely heavily on labelled data, which is often unavailable. Continual self-supervised learning (CSSL) approaches leverage self-supervised learning to enable continual learning under limited supervision. Initial efforts focused narrowly on self-supervised pre-training combined with supervised continual learning (??) or extending contrastive learning frameworks (?). However, these techniques are often designed to work with specific self-supervised learning frameworks. The latest continual self-supervised learning frameworks proposed in existing literature (??) have begun investigating more overarching and flexible frameworks, where the model learns continually from a stream of unlabelled data. The work by (?) proposed a unified semi-supervised framework with continual fine-tuning, introducing mechanisms for knowledge distillation and new task learning in both representation and classification. These demonstrate progress towards practical solutions under realistic supervision assumptions, and we focus on these recent proposals which repurpose SSL methods for continual learning in this work.

A.2 Human Activity Recognition

Wearable-based human activity recognition is an important component in human-centric computing because of its ability to extract real-time information and context clues about user behaviours, which enables other computing applications (?). However, the utility of HAR systems has been limited by the scarcity of high-quality labelled data and the computational capabilities of wearable devices.

Since the adoption of deep learning models for HAR, researchers have looked at tackling catastrophic forgetting in HAR specifically (???), and they have demonstrated moderate success. Simultaneously, self-supervised learning has recently gained popularity in the wearable-based HAR community, and many approaches from the general machine learning community, as well as customised methods, have been proposed to leverage unlabelled data to overcome the limitations of labelled data (?????). However, we have yet to see efforts in leveraging self-supervised learning techniques for HAR in continual learning. This is an important step towards practical and generalisable user models. CSSL paves the way for realistic solutions to continually learn from combinations of labelled and unlabelled data.

B Transformation Functions

Random 3D rotation applies a random rotation in the 3D space by picking a random axis in 3D and a rotational angle from uniform distributions. This is to simulate common pose changes in wearable devices.

Random scaling alters the size of samples within a window by multiplying them with a randomly chosen scalar. We apply this transformation since a model that can handle these scaled signals creates better representations because it learns to be unaffected by changes in amplitude and offset.

Time warping locally stretches or warps the time series data, smoothly distorting the time intervals between sensor readings.

C Evaluation setup

C.1 Model Architecture

In this work, we adopted a lightweight HAR model, TPN (?), to replace the vision-based model.

C.2 Evaluation Metrics

For the evaluation metrics, we are adopting the same framework as introduced in Kaizen (?): **Final Accuracy (FA)**, **Continual Accuracy (CA)**, **Forgetting (F)**, and **Forward Transfer (FT)** as metrics. This set of values can better reflect the performance of continual learning methods in different use cases.

C.3 Self-supervised Learning Frameworks

In this work we selected a contrastive-based method MoCoV2+ (??), and an asymmetric-model-based method BYOL (?) as the self-supervised learning method for continual learning. These two methods have been shown to be well-performing in continual self-supervised learning settings (?), and our goal is to investigate whether different CSSL methods demonstrate different performance characteristics when using different SSL methods as the knowledge retention mechanism.

C.4 Dataset

We performed our evaluation using the WISDM2019 (WISDM Smartphone and Smartwatch Activity and Biometrics Dataset) (?), which is an activity recognition dataset collected by the WISDM (Wireless Sensor Data Mining) Lab in the Department of Computer and Information Science of Fordham University. The dataset contains raw accelerometer and gyroscope data from a smartwatch (LG G Watch) and a smartphone (Google Nexus 5/5x or Samsung Galaxy S5) worn by 51 subjects, who performed 18 different activities for 3 minutes each. The smartphone is placed inside the participant's pocket, while the smartwatch is worn at the dominant hand. The data was collected at a sampling rate of 20Hz for the following activities: Walking, Jogging, Stairs, Sitting, Standing, Typing, Brushing Teeth, Eating Soup, Eating Chips, Eating Pasta, Drinking from Cup, Eating Sandwich, Kicking (Soccer Ball), Playing Catch with Tennis Ball, Dribbling (Basketball), Writing, Clapping, and Folding Clothes. The accelerometer data from the smartwatch

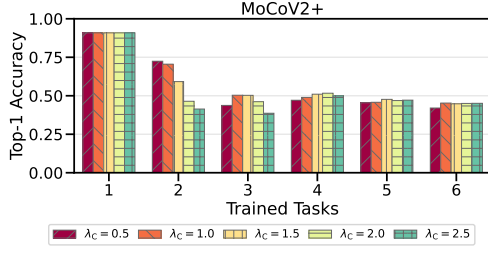


Figure 8: Performance after training on different tasks with varying constant importance coefficients.

was used in this study, and we selected this dataset for evaluation because it has a relatively high number of activities in HAR, which makes it suitable for continual learning evaluation.

The raw sensor data underwent minimal pre-processing. First, z-normalization was applied using the training data’s mean and standard deviation for each sensor channel. Next, the data was segmented into 384×3 sliding windows, representing 384 timestamps and 3 triaxial accelerometer channels. Consecutive windows do not overlap. 20-25% of user data was held out unseen as the test set to evaluate model generalisability.

The 18 classes of activities are randomly and evenly split into 6 tasks of 3 classes each with no overlap as follows: task 1 - {Folding Clothes, Stairs, Walking}, task 2 - {Sitting, Drinking from Cup, Eating Chips}, task 3 - {Standing, Eating Sandwich, Clapping}, task 4 - {Brushing Teeth, Jogging, Eating Pasta}, task 5 - {Eating Soup, writing, Typing}, and task 6 - {Playing Catch with Tennis Ball, Kicking (Soccer Ball), Dribbling (Basketball)}. This follows the conventional setup of class-incremental learning, and how different (potentially qualitative) splitting of the classes affects model performance is left as future work.

C.5 Baselines

We followed the evaluation setup as proposed in (?), in which we compare the performance of Kaizen, CaSSLe, as well as a *No distill* baseline that fine-tunes the full model on new tasks with no explicit catastrophic forgetting mitigation. As CaSSLe and *No distill* baselines are self-supervised learning methods without incorporating a classifier, the classifiers are trained separately after the feature extractor is trained, with data replay enabled.

D Additional Visualisations

Fig. 8 and 9 illustrate the aggregated performance of models trained with Kaizen after each task with different importance coefficients.

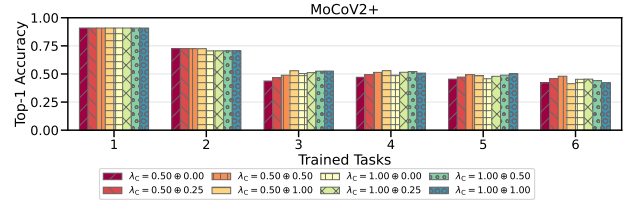


Figure 9: Performance after training on different tasks with varying progressive importance coefficients.