IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 30 May 2024, accepted 29 June 2024, date of publication 11 July 2024, date of current version 23 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3426542

RESEARCH ARTICLE

Memory-Efficient Continual Learning Object Segmentation for Long Videos

AMIR NAZEMI, MOHAMMAD JAVAD SHAFIEE, ZAHRA GHARAEE, AND PAUL FIEGUTH, (Senior Member, IEEE)
Vision and Image Processing Laboratory, Department of Systems Design Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada
Corresponding author: Amir Nazemi (amir.nazemi@uwaterloo.ca)

This work was supported in part by the Microsoft Office Media Group, in part by NSERC Alliance, and in part by the Digital Research Alliance of Canada.

ABSTRACT Recent state-of-the-art semi-supervised Video Object Segmentation (VOS) methods have shown significant improvements in target object segmentation accuracy when information from preceding frames is used in segmenting the current frame. In particular, such memory-based approaches can help a model to more effectively handle appearance changes (representation drift) or occlusions. Ideally, for maximum performance, Online VOS methods would need all or most of the preceding frames (or their extracted information) to be stored in memory and be used for online learning in later frames. Such a solution is not feasible for long videos, as the required memory size grows without bound, and such methods can fail when memory is limited and a target object experiences repeated representation drifts over time. We propose two novel techniques to reduce the memory requirement of Online VOS methods while improving modeling accuracy and generalization on long videos. Motivated by the success of continual learning techniques in preserving previously-learned knowledge, here we propose Gated-Regularizer Continual Learning (GRCL), which improves the performance of any Online VOS subject to limited memory, and a Reconstruction-based Memory Selection Continual Learning (RMSCL), which empowers Online VOS methods to efficiently benefit from stored information in memory. We also analyze the performance of a hybrid combination of the two proposed methods. Experimental results show that the proposed methods are able to improve the performance of Online VOS models by more than 8%, with improved robustness on long-video datasets while maintaining comparable performance on short-video datasets such as DAVIS16, DAVIS17, and YouTube-VOS18.

INDEX TERMS Video object segmentation, continual learning, regularization-based solutions, replay-based methods.

I. INTRODUCTION

Video object segmentation (VOS) aims to extract an accurate pixel-wise object mask in each frame of a given video. Broadly, existing VOS algorithms can be divided into two different streams: i) semi-supervised or one-shot VOS, when the ground truth masks of the target objects are provided in at least one frame at inference time, and ii) unsupervised VOS, when no information about the objects is provided. The focus of this paper is on the former, that of semi-supervised VOS.

The associate editor coordinating the review of this manuscript and approving it for publication was Sotirios Goudos.

The classic and initial solution for a semi-supervised VOS problem is to fine-tune the trained VOS model on the given information (i.e., the given object mask), separately for each test video. This ideal is not feasible, due to the limited training samples, the VOS model size, and the time-consuming fine-tuning process. In practice, online learning-based VOS approaches [1], [2], [3], [4] address these challenges by introducing efficient training (fine-tuning) mechanisms and keeping some amount of information in memory to augment the training set for model fine-tuning.

These approaches proceed on the assumption that sufficient memory is available at inference time, and that there

are no limitations in storing and exploiting information. It is also assumed that an object representation is not undergoing significant shifts between frames, such that the information stored in the memory is somehow representative of the target object in the query frame. In practice, these assumptions hold poorly, at best, and particularly in long videos it is common to experience a significant representation drift of the target object. Such a drift can lead to drastic drops in performance, particularly when there is a limitation on the memory available to store past object representations. A second bottleneck of Online VOS is its limitation to learn useful information from memory. As more training data (more frames of video) become available in the memory, Online VOS methods have difficulty to extract and learn discriminative information [5], due to their limited online model size and training process, since Online VOS prefers training small models on limited memory over few epochs. Clearly these issues become increasingly problematic on long video sequences, which are the focus of this paper.

We reformulate semi-supervised VOS as online continual learning [6], which benefits from two disjunctive solutions with a small fixed working memory to process long video sequences:

- In Section III-B, a Gated-Regularizer Continual Learning (GRCL) is proposed to improve the performance of Online VOS by preserving and consolidating the acquired knowledge from the target objects in preceding frames while limiting the required memory.
- A very different approach is developed in Section III-C, where we propose a Reconstruction-based Memory Selection Continual Learning (RMSCL) method which is able to augment any Online VOS framework and improve its performance, particularly on long videos.

The GRCL is inspired from prior-based continual learning [7], [8], whereas the latter RMSCL is motivated by rehearsal methods in continual learning [9], [10], [11], [12]. We apply the proposed methods to two state-of-the-art Online VOS algorithms, LWL [4] and Joint [3], both subject to a fixed memory. Our experimental results show an improvement of both LWL and Joint, particularly on long video sequences.

II. RELATED WORK

The primary objective of our work is to address online video object segmentation, specifically when dealing with long video sequences. Our objective particularly relates to the instances which are preserved in a memory for future selection and usage in the continuation of the learning process. We begin by overviewing baselines, state-of-the-art memory-based approaches, and methods proposed in continual learning.

We present feature selection methods with a wide range of applications in domains such as machine learning, data mining and computer vision, which can potentially be used as memory selection for VOS. Finally, we introduce several solutions available in the literature addressing the learning challenges of long video sequences.

A. MEMORY-BASED APPROACHES

Memory-based approaches [1], [2], [3], [4], [5], [13], [14], [15], [16], [17] try to address semi-supervised VOS by storing representations and predicted output masks of preceding frames in a memory, and then to use them to evaluate the current frame.

Within this strategy there are different approaches to retrieve information from the dynamic model's memory. One solution is to update (fine-tune) a small model proposed by online learning methods [2], [4], [18], [19], [20]. A second solution is to propagate the information of the most recent predicted object masks [21] or featurerepresentation of preceding frames [22], [23]. A third solution is to send a query to retrieve information of visited frames and their representation stored in the memory [1], [5], [13], [16], [24], [25], [26], [27], [28], [29], [30].

The approach proposed in this paper stems from the online learning methods, and will be compared to state-of-the-art query-based methods.

1) QUERY-BASED METHODS

Among query-based methods is STM [1], which uses a similarity matching algorithm to retrieve encoded information from the memory and pass it through a decoder to produce an output.

STM performs global matching between the query and memory frames; however, in VOS, a valid assumption is to consider the locality of the target object's appearance. Therefore, RMNet [25] developed a local-to-local matching algorithm that considers the local area where the target objects appeared in previous frames.

By limiting the potential correspondences between two consecutive frames to a local window and providing kernel guidance to the non-local memory matching, HMMN [13] offers kernel-based memory matching as a means of achieving temporal smoothness. HMMN uses tracking of the most likely relationship between a memory pixel and a query pixel to match distant frames. Unlike STM, which generates a specified memory bank for each object in the video, STCN [26] constructs a model that uses an affinity matrix based on RGB relations to learn all object relations beyond just the labeled ones. An object goes through the same affinity matrix for feature transfer when querying.

LCM [24] suggests using a memory strategy to recover pixels globally and to learn pixel position consistency for more accurate segmentation in order to deal with appearance changes and deformation.

In order to leverage the fine-grained features of instance segmentation (IS), ISVOS [16] suggest a two-branch network: a VOS branch performs spatial-temporal object level matching with the memory bank, while the proposed IS branch explores the instance details of the objects in the

current frame. They include instance-specific information into the query key using well-learned object queries from the IS branch, and then perform matching. A recent unified VOS framework, Joint-Former [17], represents the three characteristics of feature, correspondence, and dense memory. Cutie [15] benefits from a query-based object transformer that interacts with bottom-up pixel features, an object-level memory, and a small number of object queries that are continuously generated. While high-resolution feature maps are kept for exact segmentation, object queries offer an overview of the target item.

2) ONLINE LEARNING-BASED METHODS

Online learning-based methods learn a new object appearance within an online learning-based approach [3], [4], [31] simultaneously at inference time. In this scenario, instead of using a query-based (matching-based) algorithm on each frame, a small latent model network (the so-called target model) is updated every Δ_C frames, which is eventually used to produce the updated information about each video frame.

The target models proposed by FRTM [2], LWL [4] and the induction branch of JOINT [3] are formulated as a small convolutional neural network, which performs online learning on the available training data in the memory. As such, these methods can provide an efficient yet effective dynamic update process for VOS frameworks.

While target model-based approaches improve the performance of VOS, the effectiveness of online learning algorithms is highly dependent on their memory capacity and usage. In other words, to obtain the best performance, these models require storing all preceding output masks and the encoded features in memory, increasing the generalization of the updated model. The resulting memory limitation leads to facing similar challenges already known in the domain of continual learning (below).

In this paper, we hypothesize that these issues can be mitigated, specifically motivated by the success of continual learning algorithms in preserving learned knowledge while limiting required memory.

B. CONTINUAL LEARNING

Continual learning [32], [33], [34], [35] is a process of sequential learning, where the sequence of data may stem from different domains and tasks; that is, a model is learning from data in which an abrupt or gradual concept drift [36] may take place.

Similarly, in Online VOS methods a concept drift can easily happen with regards to the appearance of target objects. In such situations the distribution of the available data in the memory may significantly change with every update step. The primary challenge in this situation is known as *catastrophic forgetting*, a term which was first defined in the context of neural networks [37], [38], although it is a common problem in machine learning [39].

1) CATASTROPHIC FORGETTING

Catastrophic forgetting [40] commonly takes place in machine learning problems such as few shot learning [41], [42], graph neural networks [43], [44] knowledge distillation [45] and Bayesian inference frameworks [8].

Catastrophic forgetting occurs when a machine learning model is trained on a sequence of tasks, but at any point in time it has access to the training data of only the current task. Consequently, the learning has a tendency to update model parameters to be dominated by data from this task, resulting in a degree of forgetting previously-learned tasks.

In particular, a long video will typically have subsets in which a given object is seen from different view points, varying lighting, different object appearances, occlusion, and missing objects, all of which lead to a continual learning problem.

For an Online VOS approach, each section of a long video in the memory can be considered as a ''task'', thus forgetting the previously-learned tasks (earlier parts of the video), which can be problematic in practice since the number of tasks increases with the length of the video [46]. In this article we focus on developing continual learning-based solutions.

There are three broad approaches to catastrophic forgetting: prior-focused (regularization-based) [7], [8], likelihood-focused (rehearsal-based) [9], [10], [11], [12], and hybrid (ensemble) approaches [47], [48].

In GPM [49], a neural network model takes gradient steps in the opposite direction of the gradient subspace considered relevant for previous tasks in order to learn new tasks. GPM determines the basis of these subspaces by evaluating network representations via a Singular Value Decomposition (SVD) after learning each task. To specify a unique constraint imposed for each layer in a fine-grained fine-tuning regularization, TPGM [50] proposes an automatic constraint learning method known as trainable projected gradients.

In this paper, a regularized (GRCL) solution and a rehearsal-based based (RMSCL) solution are proposed to generalize the applicability of Online VOS on long video sequences.

C. FEATURE SELECTION

Memory reading is an important step in query-based VOS methods, as they typically employ similarity metrics to retrieve and merge partial information from memory for their decoder component. For instance, STCN [26] employs L2 similarity and STM [1] utilizes a dot product in their memory reading. This paper aims to enhance Online VOS methods that incorporate online training for a specific component of the model. Therefore, in order to maximize memory usefulness, it is beneficial to employ a simple efficient memory selection technique, because there is no requirement to partially select and merge samples from memory, as is typically done in memory reading methods. In other words, we are searching for memory selection strategies that are

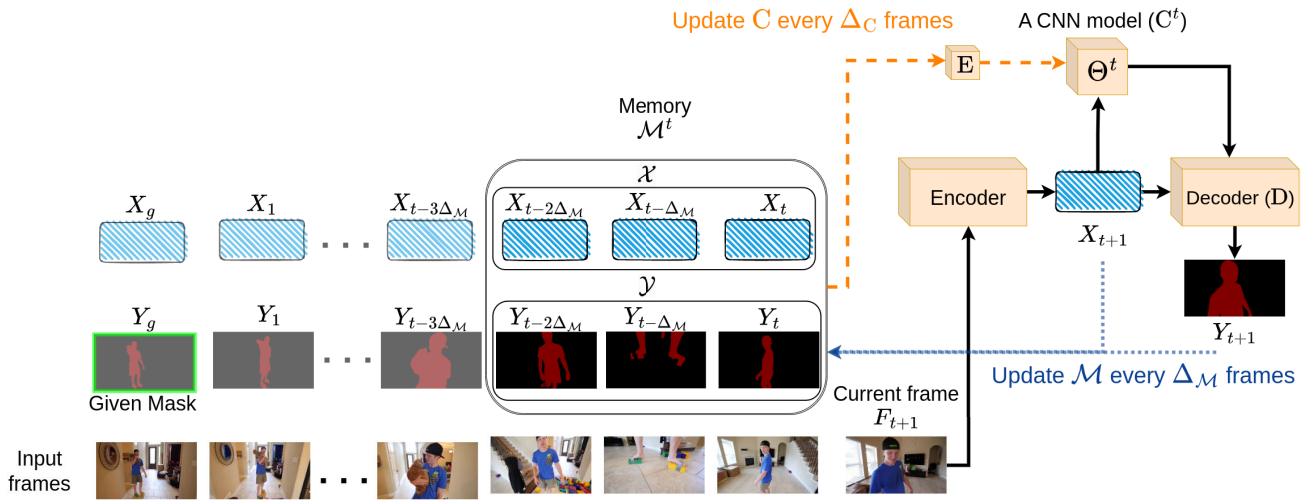

FIGURE 1. An Online VOS pipeline: The target model C^t is initialized based on the given ground truth mask Y_g and its associated feature X_g. The dashed orange line shows how the target model C^t is updated based on memory $\mathcal{M}^t$ every Δ_C frames. The blue dotted arrow illustrates how the memory $\mathcal{M}^t$ is updated every $\Delta_\mathcal{M}$ frames. The methods proposed in this paper are mainly engaged with the target model component of the pipeline.

covered in feature selection, rather than memory reading in query-based VOS methods.

For data or feature analytics, dealing with high-dimensional features greatly increases the need for memory and processing power. Additionally, the presence of duplicated, irrelevant, and noisy features raises the likelihood that learning algorithms may loose generalization ability, reducing efficiency and performance.

Feature selection for high-dimensional data are divided into supervised [51], [52], [53] and unsupervised [54], [55], [56] learning approaches. Supervised algorithms assume access to the discriminative information in the class labels, but real-world data are typically unlabeled since annotation is commonly prohibitively costly. Therefore, unsupervised feature selection techniques use several metrics, such as data similarity, density information, and data reconstruction error, to determine the quality of features.

Reconstruction based methods approximate the original data by performing a reconstruction on selected features [57], [58], [59], [60]. In this article we propose a Reconstruction-based Memory Selection Continual Learning (RMSCL) to improve Online VOS on long video sequences.

D. LONG VIDEO SEQUENCES

Long video sequences containing multiple concepts are challenging to learn, since such a model requires a memory having a large capacity in order to store information from previous frames.

In order to overcome the memory and training time constraints, AFB-URR [61] use an exponential moving average technique to store or merge new memory components. When the memory's capacity hits a set limit, the model eliminates any features that are not being used.

Global context modules [62] are another approach to limitations of long video; the model calculates a mean of the entire memory components and applies it as a single representation.

In any eventy, segmentation accuracy is compromised by both approaches since they use a compact representation of memory. In contrast, XMem [5] achieves significantly greater accuracy in both short- and long-term predictions by avoiding compression via the use of a multi-store feature memory.

III. PROPOSED APPROACH

In this section we develop two proposed methods (GRCL and RMSCL) in depth. It is important to understand that the proposed methods specifically apply to Online VOS. It needs emphasizing that these methods are not limited to one specific framework, rather they can be extended to *any* regular Online VOS architecture. The significance of this generality is that Online VOS frameworks are preferred against query-based methods in practical applications, since query-based architectures (such as XMem [5]) lead to memory requirements which grow with video length, whereas Online VOS methods assume a fixed memory size.

We begin with the general structure of Online VOS in Section III-A, followed by the formulation of the proposed gated-regularizer (GRCL) in Section III-B, and the reconstruction-based memory selection continual learning (RMSCL) in Section III-C. We conclude this section by proposing a GRCL / RMSCL hybrid.

A. ONLINE VOS

Online VOS [2], [3], [4], as overviewed in Figure 1, typically comprises the following pieces:

1) A pretrained encoder, extracting features from each frame;
2) A memory $\mathcal{M}^t$, storing features and their associated labels / mask;

3) A target model C^t, which is trained on the memory at updating time t, providing information to the decoder;
4) A label encoder network E [4], which generates sub-mask labels from each Y, guiding the target model in terms of what C^t should learn from Y.
5) A Pretrained decoder network, D, which obtains temporal information from the target model alongside the encoder's output, generating a fine-grained output mask Y_{t+1} from the current frame F_{t+1}.

The target model C^t is usually a small convolutional neural network, for reasons of efficiency. The target model is updated every Δ_{C} frames throughout the video, repeatedly trained on the complete set of features $X \in \mathcal{X}$ and the encoded labels $\mathrm{E}(Y)$ of stored decoder outputs $Y \in \mathcal{Y}$ from preceding frames. Both X and Y are stored in memory $\mathcal{M}^t$, constrained to some maximum size N, as shown in Figure 1 for $N = 3$. Y is provided to E and we seek C^t to learn what E specifies from Y. That is, the target model acts like a dynamic attention model to generate a set of score maps $\mathrm{E}(Y_i)$ in order for the target model C^t to learn to focus on important parts of mask Y_i. Thus E is only used in training the target model C^t and is not used during inference. During inference, the target model C^t is trained on the memory $\mathcal{M}^t$, with a goal of learning how to segment each stored image, basesd on loss function

$$L(\Theta^t, \mathcal{M}^t) = \sum_{n=1}^{N} \left\| d_n W_n \Big(\mathrm{E}(Y_n) - \mathrm{C}^t(X_n) \Big) \right\|_2^2 + \sum_{k=1}^{K} \lambda ||{\theta_k^t}^2||, \quad (1)$$

Here the first term of (1) pushes the target model to learn to produce the output of label encoder E, usually with a focus on recent frames, as controlled by weight d_n. The second term is a weight decay regularization. Depending on the overall architecture, E could be an offline / pre-trained label encoder network, as in [4], or just a pass-through identity function, as in [2]. Spatial pixel weight W_n balances the importance of the target and background pixels in each frame, whereas d_n defines the temporal importance of sample n in the memory, typically emphasizing more recent frames.

Online VOS methods suffer from three main limitations, particularly on long videos:

1) **Memory Size:** To maximize performance, Online VOS would need to store in memory all or most of the extracted information from all preceding frames. For videos of arbitrary length this requires an unlimited memory size, which is infeasible.
2) **Target Model Updating:** Even with an unlimited memory size, updating the target model C^t on an arbitrarily large memory is computationally problematic.
3) **Hyperparameter Sensitivity:** The sensitivity to the target model's configuration and memory updating step size affects both speed and accuracy.

The proposed GRCL and RMSCL aim to mitigate these limitations by incorporating simple yet effective methods applied to the target model C^t and memory $\mathcal{M}^t$.

Since video frame information is provided consecutively into the Online VOS framework, there is a high possibility of drift in the object's appearance, especially in long-video sequences. As such, the conventional approach of passing all of the information, as a whole, to the model to decide which to use, is not effective.

Instead, inspired by continual learning [32], we seek to regularize the parameters, Θ^t, of the target model C^t in each online learning step t, with a goal of preserving the model knowledge, acquired from those earlier frames which are no longer present in the memory. That is, we have three fundamental questions:

1) How do we constrain or regularize the model parameters? This question is explored in the gated-regularizer continual learning (GRCL) method of Section III-B. The proposed GRCL is inspired by Memory Aware Synapses (MAS) continual learning [63], allowing the memory size to be reduced while maintaining model performance, also increasing the robustness of the target model against the updating step size Δ_{C}.
2) How do we decide explicitly what to keep in the memory, or which subset of the memory to use in learning? This question is addressed in the context of reconstruction-based memory selection continual learning (RMSCL) of Section III-C, inspired by reconstruction-based feature selection methods, making it possible that updating C^t can efficiently benefit from information stored in memory.
3) What would be the performance of a solution based on both RMSCL and GRCL? Such a hybrid method is introduced in Section III-D.

B. REGULARIZATION-BASED CONTINUAL LEARNING SOLUTIONS

Parameter regularization seeks to preserve important parameters Θ, particularly those which were learned or significantly modified in preceding update steps. The MAS algorithm [63] is formulated such that at update step t the importance of each parameter θ_k^t is associated with its gradient magnitudes $\{u_k^l\}_{l=1}^{t-1}$ during preceding update steps. Therefore, during each online learning step we update the parameter weights ω_k^t based on the gradient magnitudes,

$$\omega_k^t = \omega_k^{t-1} + u_k^t \quad (2)$$

To apply a regularization-based continual learning solution, such as MAS, on Online-VOS with features $\mathcal{X}$, output masks $\mathcal{Y}$, memory $\mathcal{M}^t$, and target model C^t with K parameters Θ^t, the regularized loss function L_R is defined as

$$L_R(\Theta^t, \mathcal{M}^t) = L(\Theta^t, \mathcal{M}^t) + \gamma \sum_{k=1}^{K} \omega_k^{t-1} \left\| \theta_k^t - \theta_k^{t-1} \right\|_2^2, \quad (3)$$

where $L(\Theta^t, \mathcal{M}^t)$ is as described in (1). The latter term is a regularization, controlled by γ, where t counts the model update steps.

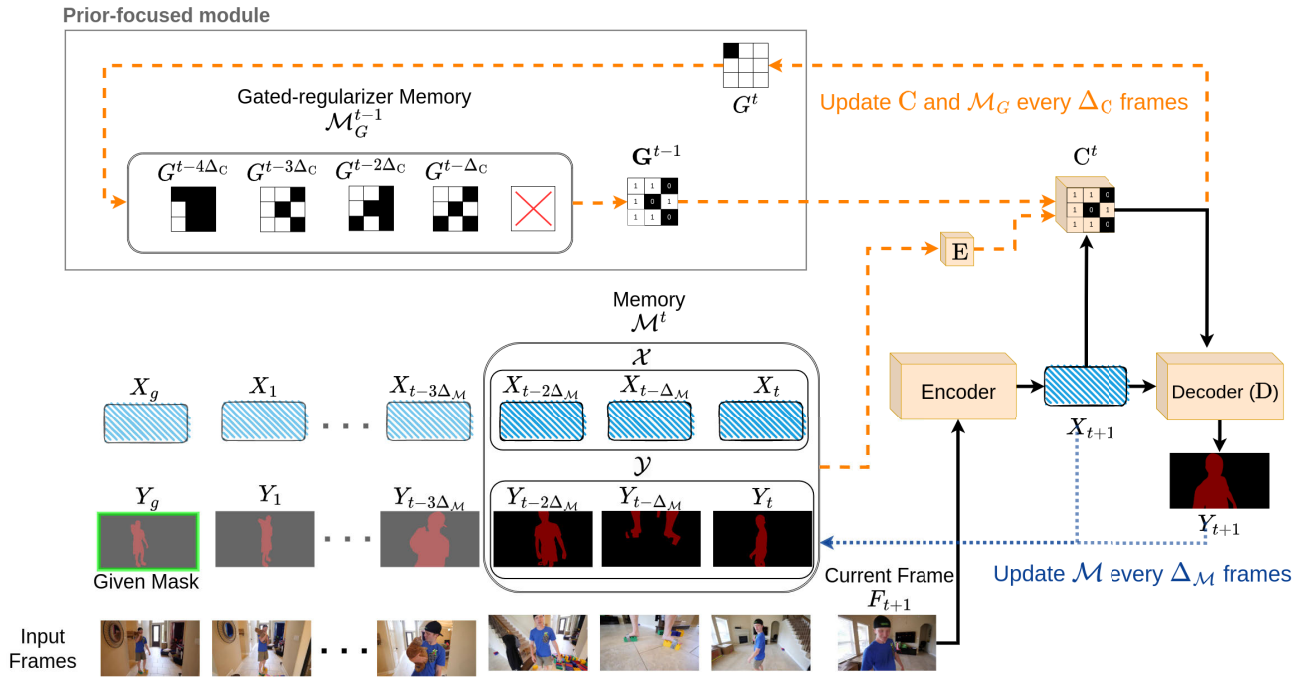

FIGURE 2. The proposed Online VOS framework, with adopted Gated-Regularized Continual Learning (GRCL): At time t, the overall gated-regularizer map $\mathbf{G}^{t-1}$ is calculated using the stored gated maps in the gated-regularizer memory $\mathcal{M}_G^{t-1}$ and regularizes the process of updating $\mathbf{C}^t$. Finally, $\mathcal{M}_G^{t-1}$ is updated and forms $\mathcal{M}_G^t$ using the calculated G^t.

The goal of all regularization-based continual learning solutions, such as MAS, is that the loss L_R allows the target model to be updated while preserving the important previously-learned knowledge. Clearly for a method such as MAS, the effectiveness of the loss function L_R deteriorates over time as $\Omega^t = \{\omega_k^t\}_{k=1}^K$ loses its effectiveness in regularization, since most parameters become important as the number of update steps t increases. This is because MAS only keeps a single ω_k^t for each parameter and accumulates the importance of newly-calculated parameters to the previous ones, as shown in (2). Other memory-based solutions, such as Elastic Weight Consolidation (EWC) [40], keep the set of gradient magnitudes $\{u_k^l\}_{l=1}^{t-1}$ for each parameter, which is not memory efficient. Our proposed GRCL tries to remove these constraints and broaden the concept to Online VOS.

GATED-REGULARIZER CONTINUAL LEARNING

We wish to formulate GRCL such that, instead of accumulating the importance parameters in Ω^t, it stores a maximum of P binarized importance maps $\{G^j\}_{j=1}^P$ in a dynamically-sized gated-regularizer memory $\mathcal{M}_G^t$, whose size is far smaller than that of the memory $\mathcal{M}^t$.

Thus, at each update step t, the overall gated-regularized map $\mathbf{G}^{t-1}$ is defined as

$$\mathbf{G}^{t-1} = \bigvee_{j=1}^{J} G^j \quad , \quad J = \left|\mathcal{M}_G^{t-1}\right| \tag{4}$$

Here $\bigvee$ is the "Logical Or" operator and $\left|\mathcal{M}_G^{t-1}\right|$ is the dynamic size of $\mathcal{M}_G^{t-1}$. The gated-regularized loss function L_G can be formulated as

$$L_G(\Theta^t, \mathcal{M}^t) = L(\Theta^t, \mathcal{M}^t) + \gamma \sum_{k=1}^{K} \mathbf{g}_k^{t-1} \left\|\theta_k^t - \theta_k^{t-1}\right\|_2^2 \tag{5}$$

where $\mathbf{g}_k^{t-1} \in \mathbf{G}^{t-1}$, such that with a sufficiently large coefficient $\gamma \cong \infty$, the latter term acts as a gating function that allows some parameters to be updated and others to be frozen. After updating the target model $\mathbf{C}^t$, a new gated-map (G^t) and memory $\mathcal{M}_G^{t-1}$ is are updated.

To this end, after accumulating the magnitude of the gradient in $U^t = \{u_k^t\}_{k=1}^K$, a binary gated-regularizer $g_k^t \in G^t$ will be defined as

$$g_k^t = \begin{cases} 1 & \text{if } \dfrac{u_k^t}{\max_k(U^t)} > h \\ 0 & \text{else} \end{cases} \tag{6}$$

where $0 < h < 1$ is a threshold, which is determined based on the distribution of the gradients in U^t. The larger the value of h, the sparser the resulting gated-regularized map G^t.

Figure 2 shows the flow-diagram of an Online VOS framework at time t when the target model $\mathbf{C}^t$ is regularized by the proposed GRCL. One of the main advantages of formulating the loss function of the Online VOS framework as L_G is to store an efficient set of binary maps $\{G^j\}_{j=1}^P$ in $\mathcal{M}_G^t$, much smaller in size compared to the sets of features $\mathcal{X}$ and masks $\mathcal{Y}$ stored in $\mathcal{M}^t$, since $P \ll N$.

1) DYNAMIC GATED-REGULARIZER MEMORY

The gated-regularizer memory $\mathcal{M}_G^t$ has a fixed size; as a result, as the number of stored gated-regularized maps increases, the remaining degrees of freedom for the target model will decrease, which could have negative effects on model performance. To address this, a mechanism is proposed to dynamically reduce the gated-regularizer memory size. When the overall gated-regularized map $\mathbf{G}^{t-1}$ is calculated, the number of ones in $\mathbf{G}^{t-1}$ determines the number of regularized parameters of the target model, and if it is smaller than a certain threshold $\eta_l = \xi_l \times K$, GRCL tends to expand $\mathcal{M}_G^t$. On the other hand, if the number of ones in $\mathbf{G}^{t-1}$ is greater than threshold $\eta_u = \xi_u \times K$, $\mathbf{G}^{t-1}$ will be shrunk, and the oldest stored gated-regularized maps in the memory $\mathcal{M}_G^{t-1}$ will be removed. The parameter thresholds are proportionate to the number of target model parameters K, so that the actual values to learn, ξ_u and ξ_l, are unit-less ratios, less problem-dependent, and will be found via cross-validation.

The GRCL strategy does not need to change an Online VOS method in order to be applied, and is therefore applicable to a wide range of proposed Online VOS techniques. GRCL only needs to regularize the loss function used to update the target model C . The only GRCL hyper-parameters that need tuning are h, that is used to binarize the gated maps (G), and ratios (ξ_l and ξ_u), which determine the bounds on parameter counts.

While several other techniques for continual learning have been described in the literature [64], [65], [66], none of them are as broadly applicable to Online VOS methods as GRCL, with its dynamic gated regularizer memory. It is worth noting that the encoder, decoder and network E in the proposed architecture are trained offline, and we use the same trained models in all experiments. The memory is initialized by the encoded features of the given frame F_g with the provided ground-truth mask Y_g, as defined in semi-supervised VOS frameworks.

C. RECONSTRUCTION-BASED MEMORY SELECTION CONTINUAL LEARNING

Given the forgetting behaviour of Online VOS due to the appearance drift of objects, a trivial solution for mitigating this problem is simply to have an unlimited memory size. However, it is difficult for a limited-size target model to extract generalized discriminating information from a considerably larger memory $\mathcal{M}^t$. As such, the effectiveness of updating the target model C^t becomes dramatically deteriorated on long videos.

To solve this limitation, we propose a dynamic working memory $\mathcal{M}_W^t$, a *subset* of $\mathcal{M}^t$, and update the target model using this new (smaller) memory instead of the (larger) memory $\mathcal{M}^t$. This new approach addresses three problems:

1) Allowing a limited size target model to benefit from a large memory.
2) The update step becomes significantly more efficient, since it is training on a smaller working memory $\mathcal{M}_W^t$.
3) The samples in the memory can have a weight in the training loss function independent of their temporal weight d_n.

The proposed RMSCL approach adapts a methodology similar to those of likelihood-based (rehearsal) approaches in continual learning, where selected observations from preceding tasks are preserved to mitigate the catastrophic forgetting of the target model on proceeding tasks.

As such, $\mathcal{M}_W^t$ needs to be a small, diverse memory which contains the required features X and masks Y of preceding evaluated frames. The goal of the proposed RMSCL is to select q samples from memory $\mathcal{M}^t$ and to place them in $\mathcal{M}_W^t$ for target model updating. This memory selection is performed on $\mathcal{M}^t$ every update step Δ_C since the goal of creating $\mathcal{M}_W^t$ is to update the target model C. The selection of samples from memory is formulated as a LASSO [67] optimization problem: To update the target model C^{t-1}, the optimal linear reconstruction of the stored features $\mathcal{X} \in \mathcal{M}^t$ for the next feature X_{t+1} is identified via a L_1 constraint on a randomly initialized vector of coefficients Ψ by minimizing

$$\begin{aligned}\Psi^t &= \arg\min_{\Psi} L_{RMSCL}(\Psi, \mathcal{M}^t, X_{t+1}) \\ &= \arg\min_{\Psi} \left(\frac{1}{2} \left\| X_{t+1} - \Psi\mathcal{X} \right\|_2^2 + \lambda \|\Psi\|_1 \right). \end{aligned} \tag{7}$$

It is worth noting that updating the target model C^{t-1} to create C^t will take place before segmenting the object in frame F_{t+1} and predicting Y_{t+1}. Moreover, $\mathcal{X}$ contains $\left|\mathcal{M}^t\right|$ features ($\mathcal{X} = \{X_l\}_{l=1}^{|\mathcal{M}^t|}$), similarly Ψ consists of $\left|\mathcal{M}^t\right|$ coefficients ($\Psi = \{\psi_l\}_{l=1}^{|\mathcal{M}^t|}$) weighting each feature X_l in reconstructing X_{t+1}. In other words, we want to have the best sparse linear reconstruction of newly received frame X_{t+1} using the stored features $\mathcal{X}$ in memory $\mathcal{M}^t$. The L_{RMSCL} loss leads to a sparse set of coefficients because of L_1-norm in (7) [68], meaning that only a small number of coefficients Ψ are non-zero after the optimization process, and the positive coefficients ψ and their associated features X are selected and are placed in $\mathcal{M}_W^t$ for updating the target model. It is important to mention that the deterministic temporal weight d_n is not involved in the loss function in (7) and instead RMSCL re-weights the selected samples in $\mathcal{M}_W^t$ by the coefficient Ψ calculated in (7). This re-weighting enables RMSCL to include the significance of selected samples in the current update phase. Thus, d_n is replaced with ψ_n in (1) as

$$\begin{aligned}&L(\Theta^t, \mathcal{M}_W^t) \\ &= \sum_{n=1}^{|\mathcal{M}_W^t|} \left\| \psi_n W_n \Big(\mathrm{E}(Y_n) - \mathrm{C}^t(X_n) \Big) \right\|_2^2 + \sum_{k=1}^{K} \lambda\, \theta_k^{t\,2}. \end{aligned} \tag{8}$$

The only problem with the LASSO minimization of (7) is that its computational complexity depends on the dimensionality of feature X, such that a gigantic feature size can lead the LASSO optimization to become the bottleneck of Online

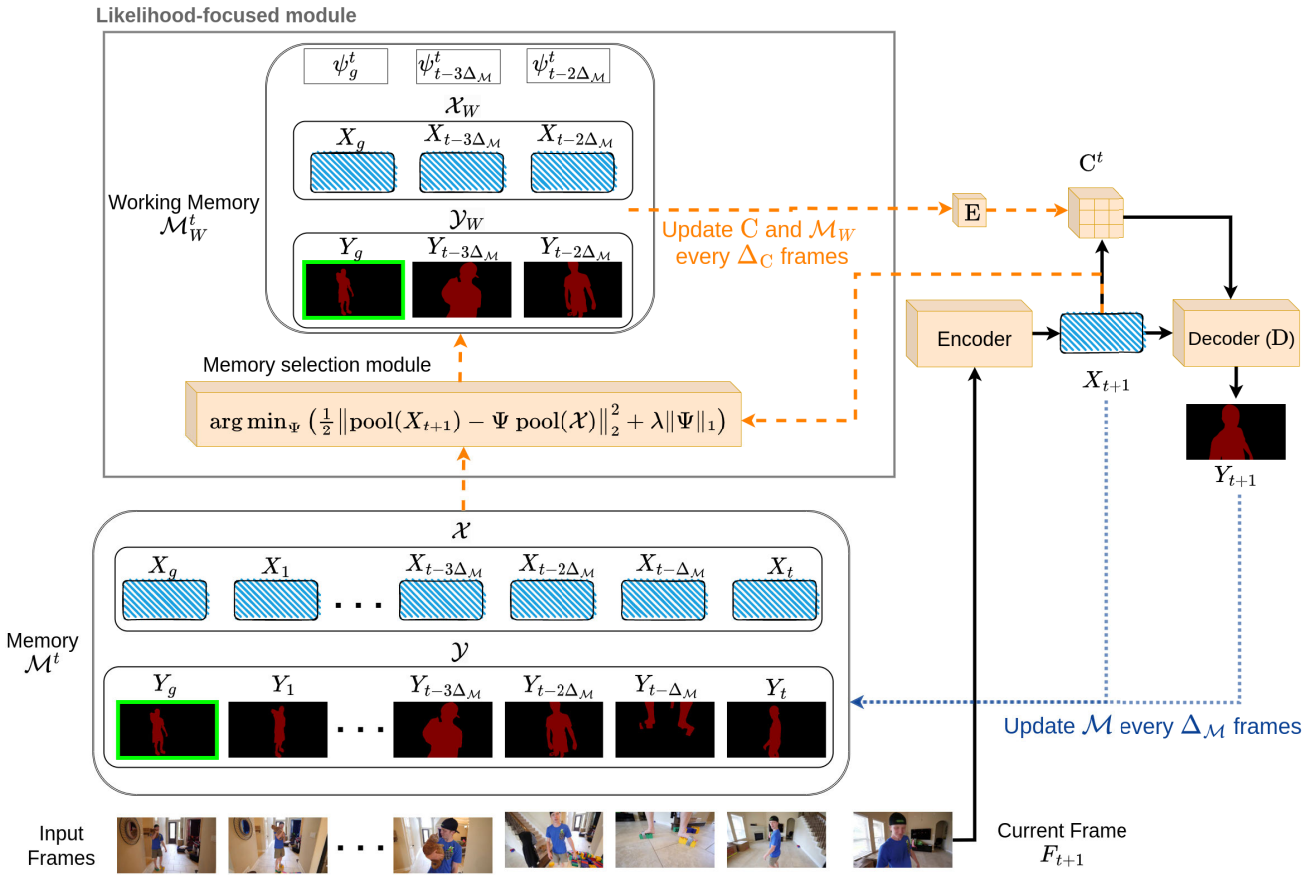

FIGURE 3. The proposed Online VOS framework with augmented Reconstruction-based Memory Selection Continual Learning (RMSCL). At the current time t, three samples associated to three positive ψ are selected using a reconstruction based (Lasso) optimization.

VOS. In order to handle this problem, a channel based max pooling function is applied on each feature X, such that (7) becomes

$$\begin{aligned}\Psi^t &= \arg\min_{\Psi} L_{RMSCL}(\Psi, \mathcal{M}^t, X_{t+1}) \\ &=\simeq \arg\min_{\Psi} \Big(\frac{1}{2}\|\text{pool}(X_{t+1}) - \Psi\text{pool}(\mathcal{X})\|_2^2 + \lambda\|\Psi\|_1 \\ &\text{s.t. } \Psi \geq 0\Big). \end{aligned} \tag{9}$$

The pooling function reduces feature X from dimensions $C \times W \times H$ to $1 \times W \times H$ by pooling over channels C, a dimensionality reduction by a factor of C. It is worth noting that the pooling function is only performed for estimating the coefficient set Ψ; it is still the actual features $\mathcal{X}$ which are used for creating the working memory $\mathcal{M}_W^t$ and updating the target model.

D. HYBRID METHOD

Hybrid methods usually benefit from three different continual learning solutions: regularization-based, replay-based, and structural-based [69]. Here, structural-based solutions of continual learning are not used since those models try to expand the model (increasing the parameters of the model) while keeping other important parameters fixed. For an Online VOS solution, expanding the model size over time is not a realistic option, since the computational bottleneck of Online VOS *is* the target model.

Here, we propose a hybrid approach that takes into account the contributions of both GRCL and RMSCL. In other words, we will evaluate a Hybrid method that has both working memory and gated-regularizer memory in its structure. The loss function L_H that we propose is

$$\begin{aligned}&L_H(\Theta^t, \mathcal{M}_W^t, \mathbf{G}^{t-1}) \\ &= L(\Theta^t, \mathcal{M}_W^t) + \gamma \sum_{k=1}^{K} \mathbf{g}_k^{t-1} \|\theta_k^t - \theta_k^{t-1}\|_2^2. \end{aligned} \tag{10}$$

The next section evaluates the proposed methods.

IV. RESULTS

The effectiveness of the proposed methods to improve the performance of Online VOS frameworks is evaluated by augmenting state-of-the-art Online VOS algorithms. The proposed GRCL, RMSCL, and Hybrid approaches can augment *any* given Online VOS framework. It is worth noting that the proposed methods can not be added to

matching-based methods, such as XMem [5], since they do not have online learning as part of their structure.

We test two well-known and state-of-the-art Online VOS frameworks: LWL [4] and JOINT [3]. LWL is an extension of the well-known FRTM [2] framework, benefiting from a label encoder network E which tells the target model what to learn [4]. JOINT approaches the VOS problem by using an online learning induction branch, jointly with a transduction branch which benefits from a lightweight transformer for providing temporal and spatial attention to its decoder. JOINT has reported the state-of-the-art performance for the problem of Online VOS in terms of accuracy.

A. DATASETS

We compared the proposed methods in the context of both short and long video sequences. The Long Video Dataset [61] contains objects with a long trajectory subject to multiple distribution drifts; the short videos are from the standard DAVIS16 [70], DAVIS17 [70], and YouTube-VOS18 [71] datasets, where the target objects are being tracked over a short period of time and usually without significant changes in appearance. Evaluating the competing methods on both short and long-video datasets demonstrates algorithm robustness to different environments.

For **long video evaluation**, in which target objects exhibit appearance changes which lead to significant representation drifts, three datasets are evaluated: LVOS [72], CLVOS23 [46], and our primary evaluation, the Long Video Dataset [61], which contains three videos with a single object which is recorded for more than 7000 frames. Each video in the dataset has 21 labelled frames for evaluation. Building on the Long Videos dataset, CLVOS23 has 9 videos with 13362 frames. The validation set of LVOS has 50 videos and a total of 31208 frames, but with less severe distribution drift than in the other two datasets.

With regards to **short-video datasets**, the DAVIS16 [70] validation set has 20 videos, each of which has a single object for segmentation; the validation set of DAVIS17 [70] contains 30 video sequences with multiple objects to be segmented in each frame. The validation set of YouTube-VOS18 has 474 video sequences of 65 seen (which are present in the training set) and 26 unseen object classes. The target objects in these datasets are mostly with a short trajectory, with modest changes in object appearance.

B. EXPERIMENTAL SETUP

We use a fixed parameter setup for the baselines, with maximum memory sizes of $N = 32$ for LWL and $N = 20$ for JOINT, as is suggested by their authors. For all experiments, the target model C^t is updated for three epochs in each updating step to have a fair comparison with baselines. The target model is updated every time the memory is updated, following the proposed setup in [5]. The memory $\mathcal{M}^t$ is initialized based on the given (ground truth) frame F_g.

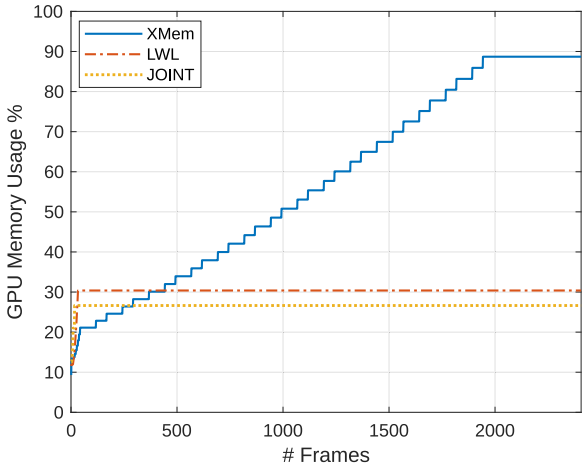

FIGURE 4. GPU memory usage of XMem, LWL and JOINT when processing the 2416 frames of the *blueboy* video in the Long Video Dataset [61]. As shown, the GPU memory usage of XMem increases significantly over time, whereas LWL and JOINT have a far more constrained GPU memory usage.

TABLE 1. Results on the Long Videos dataset [61], comparing Online VOS baseline methods, their augmented versions with GRCL, RMSCL, Hybrid, and four matching-based VOS methods. The evaluation metric $\mathcal{J}$ is related to the Intersection over Union (IoU) of an estimated object mask and the ground truth, and $\mathcal{F}$ assesses boundary accuracy.

Method	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
LWL	79.8 ± 4.2	78.0 ± 4.3	81.6 ± 4.2
LWL-GRCL (ours)	84.5 ± 1.6	82.8 ± 1.3	86.1 ± 2.0
LWL-RMSCL (ours)	83.4 ± 2.7	81.5 ± 2.6	85.2 ± 2.8
LWL-Hybrid (ours)	$\mathbf{85.4 \pm 1.0}$	84.0 ± 1.0	86.9 ± 1.1
JOINT	67.5 ± 4.4	65.7 ± 4.2	69.3 ± 4.7
JOINT-GRCL (ours)	70.5 ± 6.8	68.7 ± 6.6	72.3 ± 7.0
JOINT-RMSCL (ours)	$\mathbf{75.6 \pm 5.1}$	73.6 ± 5.0	77.5 ± 5.2
JOINT-Hybrid (ours)	72.3 ± 5.3	70.4 ± 5.2	74.0 ± 5.3
RMNet	59.8 ± 3.9	59.7 ± 8.3	60.0 ± 7.5
STM	80.6 ± 1.3	79.9 ± 0.9	81.3 ± 1.0
STCN	87.3 ± 0.7	85.4 ± 1.1	89.2 ± 1.1
XMem	$\mathbf{89.8 \pm 0.2}$	88.0 ± 0.2	91.6 ± 0.2

In all experiments, as suggested in the semi-supervised Online VOS baselines (LWL and JOINT), the information in F_g is preserved and is used throughout the whole video. For GRCL, we keep the gated-regularizer map G^0 related to the training of C^0 in $\mathcal{M}_G^t$. For RMSCL, the feature X_g and mask Y_g are always placed in working memory with a minimum weight ψ_g as shown in Figure 3. We use the same available pre-trained decoder and encoder models for all experiments of LWL and JOINT. To measure the effectiveness of the competing methods, consistent with the standard DAVIS protocol [70], the mean Jaccard $\mathcal{J}$ index, mean boundary $\mathcal{F}$ scores, and the average of $\mathcal{J}\&\mathcal{F}$ are reported. For YouTube-VOS18, the reported results are found using the YouTube-VOS18 official evaluation server [71]. The overall score $\mathcal{G}$ of seen and unseen object classes is also reported. The speed of each method is reported on DAVIS16 [2] in units of Frames Per Second (FPS) when $\Delta_C = \Delta_{\mathcal{M}} = 1$. All experiments were performed using a single NVIDIA V100 GPU.

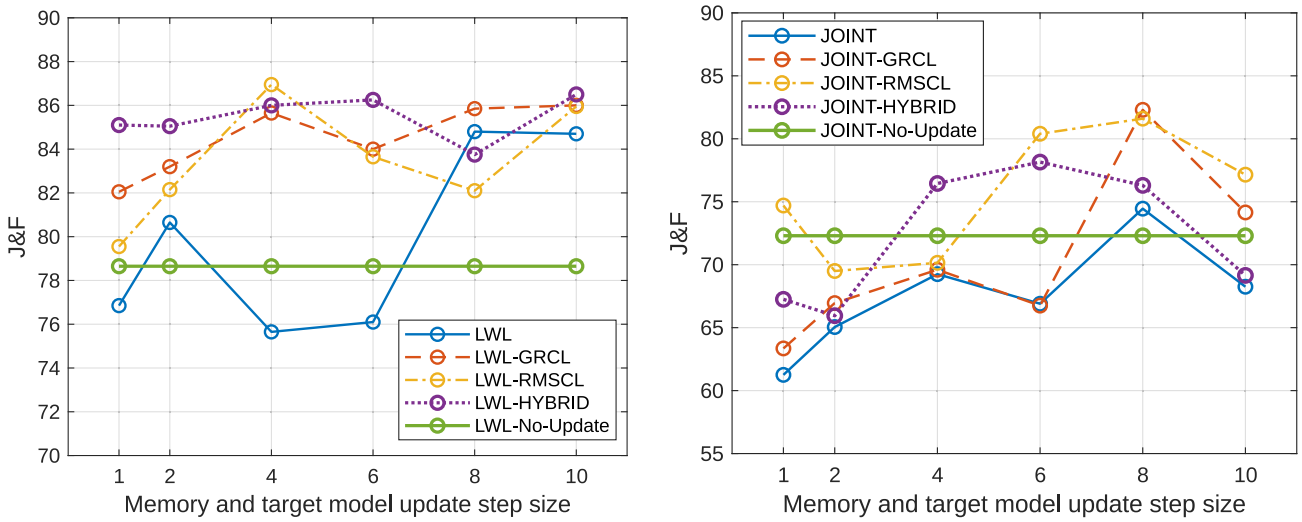

FIGURE 5. Performance comparison of competing methods as a function of memory and target model update step sizes, ($\Delta_C = \Delta_{\mathcal{M}}$), on the Long Videos dataset [61]. The left panel shows the average $\mathcal{J}\&\mathcal{F}$ on LWL and the right panel on JOINT. The green line shows the performance of LWL and JOINT without updating their target model on the memory.

TABLE 2. Performance analysis of LWL [4], proposed methods on LWL and XMem [5] against the validation sets of LVOS [72] and CLVOS23 [46]. The mean and standard deviation of six runs with different memory and target model update step sizes ($\Delta_C = \Delta_{\mathcal{M}} = \{1, 2, 4, 6, 8, 10\}$) are reported.

Method	CLVOS23 [47]			LVOS [73]		
	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
LWL [4]	67.4 ± 4.3	71.3 ± 4.7	69.3 ± 4.5	48.8 ± 0.9	57.7 ± 0.9	$\mathbf{53.3 \pm 0.9}$
LWL-GRCL (ours)	69.3 ± 4.3	73.2 ± 4.6	71.3 ± 4.5	48.5 ± 1.4	57.2 ± 1.3	$\mathbf{52.8 \pm 1.4}$
LWL-RMSCL (ours)	70.0 ± 2.9	74.3 ± 3.2	$\mathbf{72.2 \pm 3.0}$	46.8 ± 1.0	55.1 ± 1.2	50.9 ± 1.1
LWL-Hybrid (ours)	67.1 ± 1.6	73.1 ± 1.7	70.1 ± 1.7	45.4 ± 1.4	53.4 ± 1.7	49.4 ± 1.5
XMem [5]	72.9 ± 3.8	75.5 ± 4.3	$\mathbf{74.2 \pm 4.0}$	46.8 ± 7.2	56.2 ± 7.9	51.5 ± 7.6

C. EXPERIMENTAL RESULTS

1) LONG VIDEO EVALUATION

Figure 4 shows the GPU memory usage of LWL, JOINT and XMem on the "blueboy" video sequence from the Long Video Dataset. Online VOS methods (LWL and JOINT) require only a fixed GPU memory size, which enables them to be used on smaller devices with more modest GPUs. This section will show that the proposed methods do not further increase the GPU memory requirement while they *do* improve the performance of Online VOS methods.

The effectiveness of the proposed GRCL and RMSCL is evaluated by augmenting two state-of-the-art Online VOS frameworks, LWL and JOINT, however our proposed methods can be extended to any Online VOS method having a periodically-updated target model network, as in Figure 1.

Table 1 shows the results of the selected baselines (LWL and JOINT), each augmented by the proposed GRCL, RMSCL and Hybrid methods, evaluated on the Long Video Dataset. For LWL-GRCL and JOINT-GRCL, the threshold h is dynamically set to the 99.5^{th} percentile of the distribution of normalized U^t in (6). Additionally, h is limited ($0.1 < h < 0.55$) for LWL-GRCL and ($0.1 < h < 0.6$) for JOINT-GRCL. Bounding the threshold h prevents the model from extremes in over- or under-regularization. The hyper-parameters related to h were selected by cross-validation. The chosen ratios of GRCL (ξ_l and ξ_u) are 0.07 and 0.15, respectively. These ratios are defined for the target model C, and are therefore identical for LWL and JOINT.

For the frameworks in RMSCL, the parameter λ defines the sparsity of Ψ in (9). To select the best λ, the Akaike Information Criterion (AIC) [73], [74] is used for model selection, automatically selecting λ and the number of positive non-zero coefficients Ψ^t, which defines the size of the working memory $\mathcal{M}_W^t$. Thus, for each update step t, in principle $\mathcal{M}_W^t$ could have a different size in comparison to $\mathcal{M}^t$, depending upon the selected λ, the current feature X_{t+1}, and the set of features $\mathcal{X}$ in $\mathcal{M}^t$.

It is worth noting that the selected hyper-parameters for the Hybrid solution are the same as those for GRCL and RMSCL, learned for each dataset.

We conducted six experiments with six different memory and target model update steps $s \in \{1, 2, 4, 6, 8, 10\}$, where the target model C^t was updated after each memory update. The performance of RMSCL fluctuates with update step size Δ_C, because of the differing distributions which are formed in the memory as a function of sampling frequency. For

FIGURE 6. Qualitative comparison of the competing frameworks in the context of the long-video dataset. The associated frame number for each image is shown along the bottom. The leftmost column shows the given mask Y_g, which is the same for all methods. The results show that the proposed GRCL, when augmenting the baseline frameworks (LWL and JOINT), can lead to better performance against representation drift. Additionally, the frameworks based on RMSCL (LWL-RMSCL, JOINT-RMCSL) are less vulnerable to the distribution changes which take place in long video sequences. LWL-Hybrid clearly has the best performance among the compared methods.

reference, the means and standard deviations of all competing methods are reported in Table 1.

In [5], authors also compare the performance of different methods by taking the average of five runs, however, they did not report the five update steps which they used. Comparing the standard deviations of JOINT in Table 1 with those reported in [5], we see that our six selected memory update steps are close to those in [5].

The long videos tested in Table 1 are subject to trajectories with sudden representation drifts. It is therefore gratifying to see the proposed methods improving the performance of both Online VOS models, with $\mathcal{J}\&\mathcal{F}$ in LWL-Hybrid improving by more than 4% over LWL, and JOINT-RMSCL improving by a stunning 8% over JOINT.

The proposed methods in Table 1 improve the robustness of LWL with regards to the choice of memory and target model update step sizes, since the standard deviations in Table 1 are taken over the six experiments with different step sizes. It is worth mentioning that JOINT has a parallel transduction branch in its structure, which benefits from a transformer

TABLE 3. Performance analysis of the evaluated methods against the validation sets of DAVIS16, DAVIS17, and YT-VOS18. The first and second best results are highlighted in each section of the table, demonstrating that the proposed method performs similarly, if not better, on short-video datasets in comparison to the baseline methods (LWL and Joint).

Method	YT-VOS 2018					DAVIS17			DAVIS16			FPS
	$\mathcal{J}_s$	$\mathcal{F}_s$	$\mathcal{J}_u$	$\mathcal{F}_u$	$\mathcal{G}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	
LWL [4]	79.5	83.9	75.7	83.4	**80.6**	77.1	82.9	**80.0**	87.3	88.5	**87.9**	18.87
LWL-GRCL (ours)	79.7	84.0	75.9	83.9	**80.9**	77.0	83.0	**80.0**	87.3	88.6	**88.0**	15.89
LWL-RMSCL (ours)	79.4	84.0	75.2	83.2	80.5	75.6	81.8	78.7	86.5	88.3	87.4	19.01
LWL-Hybrid (ours)	77.2	81.4	71.8	79.9	77.6	72.7	79.2	76.0	83.8	85.5	84.7	16.19
JOINT [3]	81.6	85.6	78.6	86.0	**82.9**	80.6	86.0	**83.3**	87.5	89.4	88.5	6.21
JOINT-GRCL (ours)	81.6	85.5	77.6	84.9	82.4	80.4	85.8	**83.2**	87.5	89.4	88.5	6.09
JOINT-RMSCL (ours)	81.0	85.0	77.6	84.9	82.2	79.8	85.4	82.6	87.9	90.0	**89.0**	11.15
JOINT-Hybrid (ours)	81.6	85.6	78.6	85.9	**82.9**	79.9	85.4	82.6	87.8	89.9	**88.9**	10.8
RMNet [25]	82.1	85.7	75.7	82.4	81.5	81.0	86.0	83.5	88.9	88.7	88.8	11.9
STM [1]	79.7	84.2	72.8	80.9	79.4	79.2	84.3	81.8	88.7	89.9	89.3	14.0
STCN [26]	81.8	86.5	77.9	85.7	83.0	82.2	88.6	85.4	90.8	92.5	91.6	26.9
XMem [5]	84.6	89.3	80.2	88.7	85.7	82.9	89.5	86.2	90.4	92.7	91.5	29.6

model that acts like a matching-based method. Although the transduction branch of JOINT can boost the positive or even negative effects of the proposed solutions, the average performance $\mathcal{J}\&\mathcal{F}$ of JOINT is improved significantly.

For a more comprehensive comparison, the proposed methods and the baseline Online VOS frameworks are compared with four matching-based methods including RMNet [25], STM [1], STCN [26], and the current long VOS state-of-the-art XMem [5]. The reported results of the matching-based methods on short-video datasets are taken from [5]. STM is a query-based VOS baseline upon which RMNet, STCN and XMem are built. RMNet and STCN try to improve the memory functionality of STM by having a better memory encoding and memory reading methods; XMem is specifically designed to work on long video sequences.

Figure 5 lists the average performance $\mathcal{J}\&\mathcal{F}$ of six runs over different memory and target model update step sizes for the first eight methods of Table 1. On LWL, GRCL outperforms RMSCL in most cases, whereas on JOINT, RMSCL is better than GRCL, because of the effect of working memory on both branches of JOINT. It is worth noting that GRCL can impact only the induction branch (online learning part) of JOINT.

Figure 6 shows the qualitative results of the proposed methods (GRCL, RMSCL, and Hybrid) and baselines (LWL and JOINT) on seven selected frames of the "dressage" video sequence from the Long Videos dataset. The results in are produced by applying the evaluated methods to the Long Videos dataset when $\Delta_C = \Delta_{\mathcal{M}} = 1$. LWL-Hybrid has the best performance on the Long Videos dataset. The results show how RMSCL improves the performance of LWL and JOINT, and how GRCL improves LWL. The challenge of GRCL with respect to JOINT is with regards to the correctness of the prior information (which is the case with LWL). In general, baseline methods are vulnerable to the distribution drift of target objects, which are explained, discussed and formulated in [46].

In addition to the Long Video Dataset, we evaluate LWL and its augmentations by GRCL, RMSCL, and Hybrid against the strongest baseline, XMem, on two other long-video datasets, CLVOS23 [46] and LVOS [72] for a range of update step sizes ($\Delta_C = \Delta_{\mathcal{M}} = \{1, 2, 4, 6, 8, 10\}$). We used the same setup and hyper-parameters as in the Long Video Dataset. As shown in Table 2, LWL produces comparable results to XMem on long videos. The proposed methods improve LWL performance on CLVOS23, but not on LVOS, which stems from the fact that continual learning challenges are carefully designed into CLVOS23 and are present in the Long Video Dataset, such that each video contains numerous abrupt appearance changes or distribution drifts, but not so much in LVOS, where videos have some rapid but not sudden appearance changes, and only in a small subset of validation videos.

2) SHORT VIDEO EVALUATION

Table 3 demonstrates performance on short-video datasets (DAVIS16, DAVIS17, and YouTube-VOS18). The same hyper-parameters are used for short and long videos, meaning that the models have no prior knowledge of video length. Objects in short-video datasets have a short trajectory and their representations are mostly kept intact or only gradually changing. From Table 3, augmenting by GRCL performs the same as the baseline, and the proposed regularizer not only does not affect the performance of the baseline method when there is no representation drift on objects in videos, but also LWL-GRCL performs slightly better compared to LWL on YouTube-VOS18.

Table 3 uses parameters as suggested by the baseline models for reporting $\mathcal{J}$, $\mathcal{F}$ and FPS. For JOINT, $\mathcal{M}^t$ is updated every three frames; for LWL $\mathcal{M}^t$ is updated every frame; XMem updates its so called working memory every five frames. The proposed RMSCL improves the performance of JOINT on DAVIS16 but it slightly degrades the performance of JOINT on DAVIS17 and YouTube-VOS18. In general, the degredations are modest, and in any event the short-video results are shown for completeness, however the methods are designed to particularly address the distributional changes encountered, in practice, in nearly all videos, but not in the short-video datasets. The baselines perform slightly better than GRCL in terms of FPS, since

FIGURE 7. The qualitative comparison of the evaluated methods on a short video from DAVIS16 [70]reflecting the quantitative results of Table 3. The proposed GRCL and RMSCL can improve the performance of JOINT while having no or only modest negative impacts on LWL. The red-highlighted frames emphasize how the proposed approach outperforms the JOINT baseline.

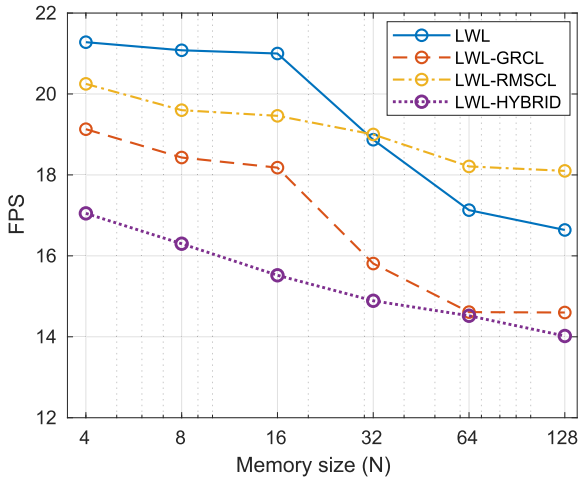

FIGURE 8. Run-time evaluation: The proposed methods' run-times are compared against the LWL baseline. LWL-RMSCL reports higher Frame Per Second (FPS) when the memory size *N* is increased.

GRCL needs to calculate a new G^t after every updating step t, however for a small target model this FPS degradation is not significant.

Figure 7 offers a qualitative comparison of the proposed methods and baselines on the "soapbox" video sequence of DAVIS16, one of the longest video sequences in DAVIS16 at 99 frames. The proposed methods offer positive improvements on JOINT, with slight changes on LWL, in agreement with the reported results in Table 3.

On long video sequences it is not feasible to store all of the previously evaluated frames in memory $\mathcal{M}$, as such it is important consider the effect of memory size N, tested at $N \in \{8, 16, 32, 64, 128\}$ on the Long Videos dataset, with the target model and memory update step $\Delta_{\mathcal{M}} = \Delta_{\mathrm{C}} = 4$. In general, and not surprising, increasing the memory size improves performance (Figure 9) but increases computational complexity (Figure 8). Increasing the memory size does not have a significant effect on LWL-RMSCL, since it does not have any hyper-parameters that are affected by the memory size, however the hyper-parameter tuning (P, h, ξ_l, ξ_u) of LWL-GRCL and consequently LWL-Hybrid is implicitly affected by the size of the memory. RMSCL, on the other hand, provides a small set of diverse data with new weights Ψ in its dynamic working memory $\mathcal{M}_W^t$, which improves both accuracy and speed of the baseline methods on long videos.

Figure 8 illustrates the impact of memory size N on speed, measured in FPS on DAVIS16, for memory and target model update steps set to $\Delta_{\mathcal{M}} = \Delta_{\mathrm{C}} = 1$. Since LWL-RMSCL uses a smaller working memory $\mathcal{M}_W^t$ for training the target model, it is faster than LWL and LWL-GRCL when the memory size

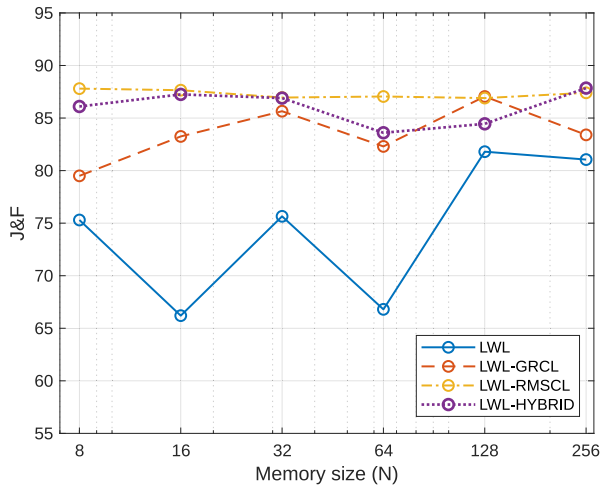

FIGURE 9. The effect of different memory size N against the proposed methods compared to the baselines on the Long Video Dataset [61]. Observe how the LWL baseline fluctuates significantly, in contrast to improved performance and reduced fluctuations associated with the proposed methods. The target model and memory update step are $\Delta_{\mathcal{M}} = \Delta_{\mathrm{C}} = 4$.

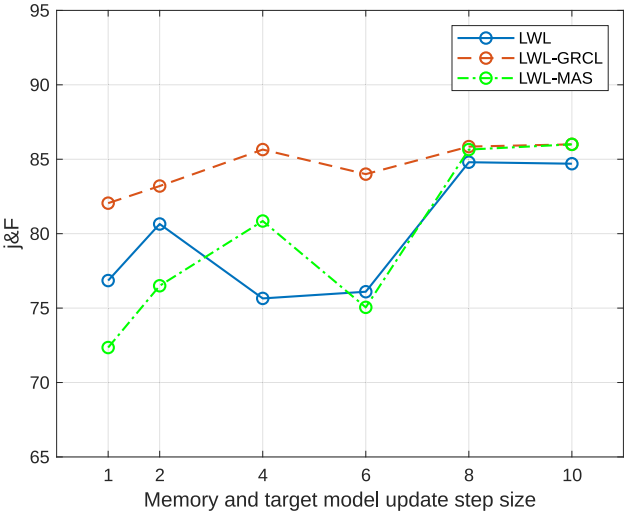

FIGURE 10. Quantitative evaluation of the proposed GRCL with conventional continual learning (MAS) [63] on the Long Video Dataset [61]. MAS [63] is less effective than the proposed GRCL when adding onto LWL.

is increased. It is worth mentioning that minimizing (9) in RMSCL is affected by the memory size N, and consequently it affects the FPS of LWL-RMSCL as well. LWL-Hybrid has the lowest FPS among the proposed approaches since it has the computational complexity of both GRCL and RMSCL.

3) CONVENTIONAL CONTINUAL LEARNING

One important aspect of the proposed continual learning approaches to augment Online VOS frameworks is that they are designed especially for this purpose. To illustrate, Figure 10 compares the performance of proposed LWL-GRCL against LWL augmented by standard MAS continual learning [63] as a regularizer for updating the target

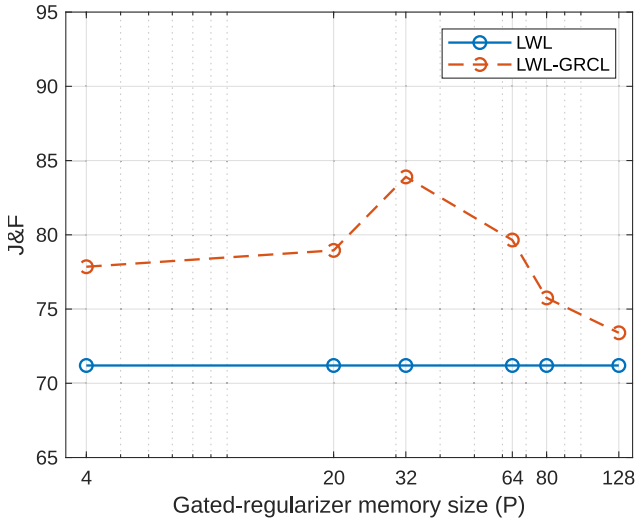

FIGURE 11. The effect of regularized-gated memory size P on the LWL-GRCL framework with a fixed size gated memory $\mathcal{M}_G^t$. For this experiment, memory size $\mathcal{M}^t$ is fixed ($N = 8$) in order to properly analyze the impact of GRCL. By setting P to a large number, the target model C^t will not have enough free parameters to be updated on memory $\mathcal{M}^t$.

model, tested on the Long Video Dataset. As shown in the figure, LWL-GRCL reported higher average performance of $\mathcal{J}\&\mathcal{F}$ compared to LWL-MAS, for two reasons: i) The overall gated-regularized map $\mathbf{G}^{t-1}$ described in Figure 3 preserves the efficiency of GRCL, whereas the MAS regularizer loses its efficiency as update steps are increased. MAS benefits highly from Ω^t, however the efficiency of Ω^t is degraded as more and more target model gradients are processed, accumulated, and stored over time, causing all of the parameters to become important as the number of updates increases. In contrast LWL-GRCL, with its dynamic memory size, guarantees that the target model C^t has enough free parameters to learn new tasks. ii) For a small number of training epochs, in each updating step of C^t the binarized (hard) regularizer $\mathbf{G}^{t-1}$ is more effective than MAS with a soft regularizer Ω^t.

4) MEMORY EFFICIENCY

To compare the memory efficiency of GRCL against the baseline, we can compare each unit of memory $\mathcal{M}$ of LWL and of LWL-GRCL. In LWL, each sample in the memory $\mathcal{M}$ consists of the preceding estimated object masks $\mathcal{Y}$ and its related extracted features $\mathcal{X}$. Each feature $X \in \mathcal{X}$ has a dimension of $512 \times 30 \times 52$ floats (64 bits). In contrast, each binary regularized-gated map (G) has a dimension of $512 \times 16 \times 3 \times 3$ bits. Moreover, each unit of $\mathcal{M}$ also has a binary mask of the target model C. As a result, each unit of $\mathcal{M}_G^t$ is almost 700 times smaller than each unit of $\mathcal{M}$. Thus, having a large gated-regularizer memory $\mathcal{M}_G^t$ is much less expensive than having a large memory $\mathcal{M}$.

D. ABLATION STUDY

In this section, we evaluate the effect of key parameters of the proposed methods on the performance of both LWL and

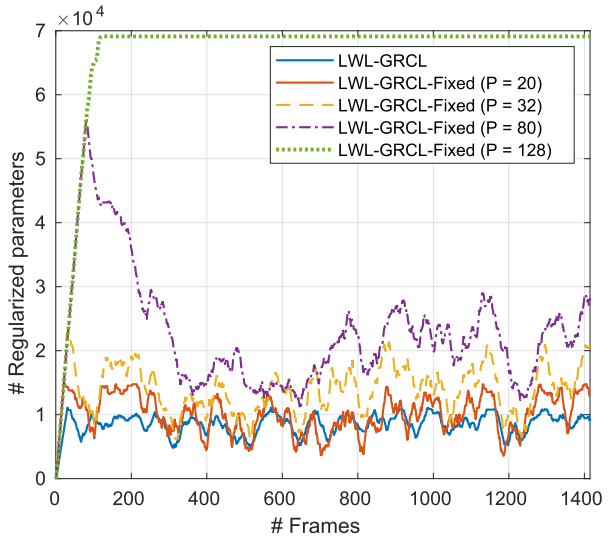

FIGURE 12. The number of regularized target model parameters when incorporated into LWL-GRCL, as a function of regularized-gated memory sizes ($P = \{20, 32, 80, 128\}$). The result are based on 1416 frames of the *rat* video sequence of the Long Video Dataset [61]. C^t is updated every frame and the memory size is set to eight ($N = 8$).

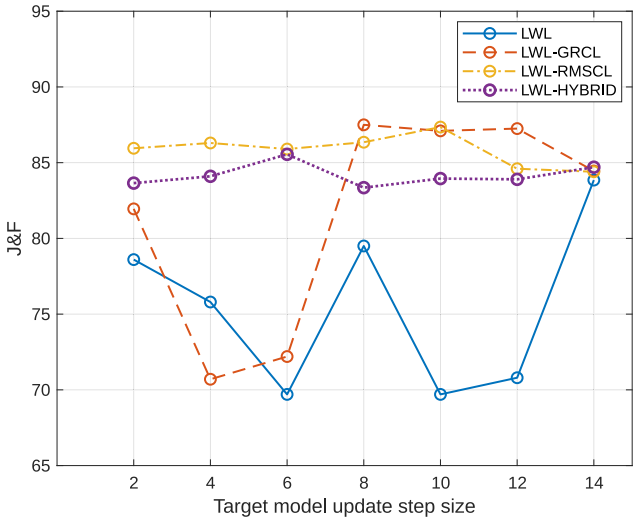

FIGURE 13. The effect of target model update step size Δ_C: The competing methods are evaluated via the Long Video Dataset [61]. The results show that the proposed LWL-RMSCL is more robust to target model update step size Δ_C.

JOINT, based on the Long Video Dataset [61], to clearly illustrate the impacts of the proposed approaches. In general the gated-regularizer memory $\mathcal{M}_G^t$ has a dynamic size, controlled by (ξ_l, ξ_u), however a fixed gated-regularizer memory size P allows a more systematic study of memory size dependence. Figure 11 shows the performance of LWL-GRCL with memory sizes $P \in \{4, 20, 32, 64, 80, 128\}$. Increasing P improves the performance of LWL-GRCL until the number of regularized parameters do not degrade target model learning; the best value for P will depend on N, however for $N = 8$ LWL-GRCL has its best performance for $P = 32$.

The number of regularized parameters in C^t is an important factor related to the ability of the target model to learn new information. As seen in Figure 12, the number of regularized parameters increases while the gated-regularizer memory $\mathcal{M}_G$ is growing, evaluating new frames. This growth is clearly dependent on P, so for $P = 128$, almost all of the parameters of C are regularized, and in this case C^t does not have any free parameters to be trained, and even removing or replacing one gated-regularization map G^j from $\mathcal{M}_G$ would not free enough parameters to allow C^t to be updated. In contrast, when the gated-regularizer memory $\mathcal{M}_G$ reaches its maximum capacity, the oldest G in memory is replaced by the next gated-regularizer map, typically freeing up parameters, which can clearly be seen in Figure 12 for $P = 80$. To address the issue discussed issue regarding GRCL-Fixed, a mechanism is proposed that makes $\mathcal{M}_G$ dynamic in size.

To demonstrate the effect of target model update step size, an ablation study compares the performance of LWL-GRCL, LWL-RMSCL, LWL-Hybrid, and LWL. The memory update step size is set to $\Delta_{\mathcal{M}} = 1$, whereas the target model update step size varies $\Delta_C \in \{2, 4, 6, 8, 10, 12, 14\}$. The memory $\mathcal{M}$ and C^t were updated sequentially at the same time index ($\Delta_C = \Delta_{\mathcal{M}}$). The memory capacity is set to $N = 4$, making the situation difficult for all of the evaluated approaches. Figure 13 shows that LWL has the lowest performance when compared to LWL-GRCL and LWL-RMSCL. The proposed methods reduce the degree of LWL performance degradation, except for LWL-GRCL at $\Delta_C = 4$, a case which demonstrates a limitation when the model concentrates on an incorrect prior, which is maintained and amplified during the evaluation of future frames.

V. CONCLUSION AND FUTURE WORKS

In this paper, we proposed two novel modules, Gated-Regularizer Continual Learning (GRCL) and Reconstruction-based Memory Selection Continual Learning (RMSCL), which can be integrated with any Online VOS algorithm to improve their memory limitations, improving their performance on long videos with the distribution drift, and preserving their performance accuracy on short videos. The offline trained parts — the encoder, decoder D, and label encoder E — do not need to be re-trained for the proposed methods, making it possible to apply the proposed methods on any Online VOS consistent with Figure 1 without any further fine-tuning.

We showed that the proposed Hybrid method (a combination of two proposed methods) in many cases increases the performance of the augmented baselines, although further refined combinations of GRCL and RMSCL would be desirable. The proposed methods improve the performance of Online VOS in a variety of different scenarios and aspects such as speed, accuracy and robustness. The proposed regularization-based GRCL, although designed for long videos having distributional shifts, maintains baseline per-

formance even on short-video datasets (DAVIS16, DAVIS17, and YouTube-VOS18).

The main limitation of GRCL is that if the prior knowledge learned during the previous update time is incorrect, GRCL insists on remembering the incorrect prior (learned information) for the next update steps, causing the model to lose the correct target object and being unable to recover from the incorrect information in memory. This paper proposes future work that uses heuristic data to establish a quality metric for the available data in memory, thereby bypassing certain target model update steps. It is worth mentioning that there is no message passing between RMSCL and GRCL in the proposed Hybrid method, so their effects may be similar or contradictory. For future work, we can consider an interactive relationship between RMSCL and GRCL to improve the Hybrid method. In semi-supervised VOS, the given information may become less important as we approach the end of a long video; thus, incorporating the self-supervision method [75] and fusing it with the current method could be one future work. Finally, we could apply the proposed methods to other areas of video processing, specifically object tracking, which may encounter issues such as memory constraints and distribution drift during the online learning process.

REFERENCES

[1] S. W. Oh, J.-Y. Lee, N. Xu, and S. J. Kim, "Video object segmentation using space-time memory networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9225–9234.

[2] A. Robinson, F. J. Lawin, M. Danelljan, F. S. Khan, and M. Felsberg, "Learning fast and robust target models for video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7404–7413.

[3] Y. Mao, N. Wang, W. Zhou, and H. Li, "Joint inductive and transductive learning for video object segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Montreal, QC, Canada, Oct. 2021, pp. 9650–9659.

[4] G. Bhat, F. J. Lawin, M. Danelljan, A. Robinson, M. Felsberg, L. Van Gool, and R. Timofte, "Learning what to learn for video object segmentation," in *Proc. Eur. Conf. Comput. Vis.* Glasgow, U.K.: Springer, Nov. 2020, pp. 777–794.

[5] H. K. Cheng and A. G. Schwing, "XMem: Long-term video object segmentation with an atkinson-shiffrin memory model," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 640–658.

[6] G. I. Parisi and V. Lomonaco, "Online continual learning on sequences," in *Recent Trends in Learning From Data*. Springer, 2020, pp. 197–221.

[7] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3366–3385, Jul. 2022.

[8] P.-H. Chen, W. Wei, C.-J. Hsieh, and B. Dai, "Overcoming catastrophic forgetting by Bayesian generative regularization," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 1760–1770.

[9] C. Atkinson, B. McCane, L. Szymanski, and A. Robins, "Pseudo-rehearsal: Achieving deep reinforcement learning without catastrophic forgetting," *Neurocomputing*, vol. 428, pp. 291–307, Mar. 2021.

[10] F. M. Castro, M. J. Marín-Jiménez, N. Guil, C. Schmid, and K. Alahari, "End-to-end incremental learning," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 233–248.

[11] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, and Y. Fu, "Large scale incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 374–382.

[12] B. Zhao, X. Xiao, G. Gan, B. Zhang, and S.-T. Xia, "Maintaining discrimination and fairness in class incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13205–13214.

[13] H. Seong, S. W. Oh, J.-Y. Lee, S. Lee, S. Lee, and E. Kim, "Hierarchical memory matching network for video object segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 12869–12878.

[14] Z. Zhou, L. Ren, P. Xiong, Y. Ji, P. Wang, H. Fan, and S. Liu, "Enhanced memory network for video segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 689–692.

[15] H. K. Cheng, S. W. Oh, B. Price, J.-Y. Lee, and A. Schwing, "Putting the object back into video object segmentation," 2023, *arXiv:2310.12982*.

[16] J. Wang, D. Chen, Z. Wu, C. Luo, C. Tang, X. Dai, Y. Zhao, Y. Xie, L. Yuan, and Y.-G. Jiang, "Look before you match: Instance understanding matters in video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 2268–2278.

[17] J. Zhang, Y. Cui, G. Wu, and L. Wang, "Joint modeling of feature, correspondence, and a compressed memory for video object segmentation," 2023, *arXiv:2308.13505*.

[18] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool, "One-shot video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5320–5329.

[19] P. Voigtlaender and B. Leibe, "Online adaptation of convolutional neural networks for video object segmentation," in *Proc. Brit. Mach. Vis. Conf.*, London, U.K., 2017, pp. 1–16.

[20] Y. Liu, L. Liu, H. Zhang, H. Rezatofighi, Q. Yan, and I. Reid, "Meta learning with differentiable closed-form solver for fast video object segmentation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 8439–8446.

[21] F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung, "Learning video object segmentation from static images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3491–3500.

[22] Y.-T. Hu, J.-B. Huang, and A. Schwing, "MaskRNN: Instance level video object segmentation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–10.

[23] C. Ventura, M. Bellver, A. Girbau, A. Salvador, F. Marques, and X. Giro-I-Nieto, "RVOS: End-to-end recurrent network for video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5272–5281.

[24] L. Hu, P. Zhang, B. Zhang, P. Pan, Y. Xu, and R. Jin, "Learning position and target consistency for memory-based video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4142–4152.

[25] H. Xie, H. Yao, S. Zhou, S. Zhang, and W. Sun, "Efficient regional memory network for video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1286–1295.

[26] H. K. Cheng, Y.-W. Tai, and C.-K. Tang, "Rethinking space-time networks with improved memory coverage for efficient video object segmentation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 11781–11794.

[27] Z. Yang, Y. Wei, and Y. Yang, "Associating objects with transformers for video object segmentation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 2491–2502.

[28] F. Lin, H. Xie, Y. Li, and Y. Zhang, "Query-memory re-aggregation for weakly-supervised video object segmentation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, 2021, pp. 2038–2046.

[29] Y. Liu, R. Yu, J. Wang, X. Zhao, Y. Wang, Y. Tang, and Y. Yang, "Global spectral filter memory network for video object segmentation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 648–665.

[30] H. Seong, J. Hyun, and E. Kim, "Kernelized memory network for video object segmentation," in *Proc. Eur. Conf. Comput. Vis.* Glasgow, U.K.: Springer, 2020, pp. 629–645.

[31] J. Kivinen, A. J. Smola, and R. C. Williamson, "Online learning with kernels," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2165–2176, Aug. 2004.

[32] R. Aljundi, "Continual learning in neural networks," Ph.D. thesis, KU Leuven, Leuven, Belgium, 2019.

[33] Y.-C. Hsu, Y.-C. Liu, and Z. Kira, "Re-evaluating continual learning scenarios: A categorization and case for strong baselines," 2018, *arXiv:1810.12488*.

[34] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2935–2947, Dec. 2018.

[35] Y. Yang, B. Lai, and S. Soatto, "DyStaB: Unsupervised object segmentation via dynamic-static bootstrapping," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2825–2835.

[36] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Comput. Surv.*, vol. 46, no. 4, pp. 1–37, Apr. 2014.

[37] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of Learning and Motivation*, vol. 24. Amsterdam, The Netherlands: Elsevier, 1989, pp. 109–165.

[38] R. Ratcliff, "Connectionist models of recognition memory: Constraints imposed by learning and forgetting functions," *Psychol. Rev.*, vol. 97, no. 2, pp. 285–308, 1990.

[39] Z. Erdem, R. Polikar, F. Gurgen, and N. Yumusak, "Ensemble of svms for incremental learning," in *Proc. Int. Workshop Multiple Classifier Syst.* Berlin, Germany: Springer, 2005, pp. 246–256.

[40] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, and A. Grabska-Barwinska, "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 13, pp. 3521–3526, Mar. 2017.

[41] G. Shi, J. Chen, W. Zhang, L. M. Zhan, and X. M. Wu, "Overcoming catastrophic forgetting in incremental few-shot learning by finding flat minima," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 6747–6761.

[42] P. Yap, H. Ritter, and D. Barber, "Addressing catastrophic forgetting in few-shot problems," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 11909–11919.

[43] H. Liu, Y. Yang, and X. Wang, "Overcoming catastrophic forgetting in graph neural networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, 2021, pp. 8653–8661.

[44] F. Zhou and C. Cao, "Overcoming catastrophic forgetting in graph neural networks with experience replay," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 5, pp. 4714–4722.

[45] K. Binici, N. T. Pham, T. Mitra, and K. Leman, "Preventing catastrophic forgetting and distribution mismatch in knowledge distillation via synthetic data," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 3625–3633.

[46] A. Nazemi, Z. Moustafa, and P. Fieguth, "CLVOS23: A long video object segmentation dataset for continual learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2023, pp. 2495–2504.

[47] S.-W. Lee, C.-Y. Lee, D.-H. Kwak, J. Kim, J. Kim, and B.-T. Zhang, "Dual-memory deep learning architectures for lifelong learning of everyday human behaviors," in *Proc. IJCAI*, 2016, pp. 1669–1675.

[48] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," 2016, *arXiv:1606.04671*.

[49] G. Saha, I. Garg, and K. Roy, "Gradient projection memory for continual learning," 2021, *arXiv:2103.09762*.

[50] J. Tian, X. Dai, C.-Y. Ma, Z. He, Y.-C. Liu, and Z. Kira, "Trainable projected gradient method for robust fine-tuning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 7836–7845.

[51] M. Thomas and A. T. Joy, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 2006.

[52] L. Jian, J. Li, K. Shu, and H. Liu, "Multi-label informed feature selection," in *Proc. IJCAI*, vol. 16, 2016, pp. 1627–1633.

[53] J. Li and H. Liu, "Challenges of feature selection for big data analytics," *IEEE Intell. Syst.*, vol. 32, no. 2, pp. 9–15, Mar. 2017.

[54] J. Li, X. Hu, J. Tang, and H. Liu, "Unsupervised streaming feature selection in social media," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manag.*, Oct. 2015, pp. 1041–1050.

[55] J. Li, X. Hu, L. Jian, and H. Liu, "Toward time-evolving feature selection on dynamic networks," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 1003–1008.

[56] X. Wei and S. Y. Philip, "Unsupervised feature selection by preserving stochastic neighbors," in *Proc. Artif. Intell. Statist.*, 2016, pp. 995–1003.

[57] Y. Wu, Q. Ren, S. Sun, and T. Tang, "Memory reconstruction based dual encoders for anomaly detection," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2022, pp. 2244–2250.

[58] R. Zhang and X. Li, "Unsupervised feature selection via data reconstruction and side information," *IEEE Trans. Image Process.*, vol. 29, pp. 8097–8106, 2020.

[59] Y. Liu, D. Ye, W. Li, H. Wang, and Y. Gao, "Robust neighborhood embedding for unsupervised feature selection," *Knowl.-Based Syst.*, vol. 193, Apr. 2020, Art. no. 105462.

[60] J. Li, J. Tang, and H. Liu, "Reconstruction-based unsupervised feature selection: An embedded approach," in *Proc. IJCAI*, 2017, pp. 2159–2165.

[61] Y. Liang, X. Li, N. Jafari, and Q. Chen, "Video object segmentation with adaptive feature bank and uncertain-region refinement," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, vol. 33, Dec. 2020, pp. 3430–3441.

[62] Y. Li, Z. Shen, and Y. Shan, "Fast video object segmentation using the global context module," in *Proc. Eur. Conf. Comput. Vis.* Springer, Nov. 2020, pp. 735–750.

[63] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars, "Memory aware synapses: Learning what (not) to forget," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, Sep. 2018, pp. 139–154.

[64] D. Abati, J. Tomczak, T. Blankevoort, S. Calderara, R. Cucchiara, and B. E. Bejnordi, "Conditional channel gated networks for task-aware continual learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3930–3939.

[65] J. Serra, D. Suris, M. Miron, and A. Karatzoglou, "Overcoming catastrophic forgetting with hard attention to the task," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4548–4557.

[66] S. Jung, H. Ahn, S. Cha, and T. Moon, "Continual learning with node-importance based adaptive group sparse regularization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 3647–3658.

[67] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Stat. Soc. Ser. B, Stat. Methodol.*, vol. 58, no. 1, pp. 267–288, Jan. 1996.

[68] H. Lee, A. Battle, R. Raina, and A. Ng, "Efficient sparse coding algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 19, 2006.

[69] J. Yoon, J. Lee, E. Yang, and S. J. Hwang, "Lifelong learning with dynamically expandable network," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–11.

[70] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 724–732.

[71] N. Xu, L. Yang, Y. Fan, D. Yue, Y. Liang, J. Yang, and T. Huang, "Youtube-vos: A large-scale video object segmentation benchmark," 2018, *arXiv:1809.03327*.

[72] L. Hong, W. Chen, Z. Liu, W. Zhang, P. Guo, Z. Chen, and W. Zhang, "LVOS: A benchmark for long-term video object segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 13480–13492.

[73] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Autom. Control*, vol. AC-19, no. 6, pp. 716–723, Dec. 1974.

[74] K. Aho, D. Derryberry, and T. Peterson, "Model selection for ecologists: The worldviews of AIC and BIC," *Ecology*, vol. 95, no. 3, pp. 631–636, Mar. 2014.

[75] Z. Qin, X. Lu, D. Liu, X. Nie, Y. Yin, J. Shen, and A. C. Loui, "Reformulating graph kernels for self-supervised space-time correspondence learning," *IEEE Trans. Image Process.*, vol. 32, pp. 6543–6557, 2023.

AMIR NAZEMI received the B.Sc. and M.Sc. degrees in computer software engineering and artificial intelligence, in 2010 and 2014, respectively, and the Ph.D. degree in systems design engineering from the University of Waterloo, Canada, in 2023, with a focus on continual learning-based video object segmentation. He is currently a Postdoctoral Researcher with the Department of Systems Design Engineering, University of Waterloo. He has involved in many research projects with ETRI, Microsoft, and the Faculty of Health, University of Waterloo. His research interests include continual learning and video object segmentation on long videos, computer vision, machine learning, specifically generative AI, and medical imaging.

MOHAMMAD JAVAD SHAFIEE received the B.Sc. and M.Sc. degrees in computer science and artificial intelligence, in 2008 and 2011, respectively, and the Ph.D. degree in systems design engineering from the University of Waterloo, Canada, in 2017, with a focus on machine learning and deep learning. He is currently an Adjunct Professor with the Department of Systems Design Engineering, University of Waterloo. His research interests include statistical learning and graphical models with random fields and deep learning approaches, computer vision, machine learning, and biomedical image processing.

ZAHRA GHARAEE received the B.Sc. degree in electrical control engineering and the M.Sc. degree in mechatronics from the K. N. Toosi University of Technology, Iran, in 2009 and 2012, respectively, and the Ph.D. degree in cognitive science from the Department of Philosophy and Cognitive Science, Lund University, Sweden, in 2018. She was a Postdoctoral Researcher with the Computer Vision Laboratory (CVL), Department of Electrical Engineering, Linköping University, Sweden, from 2018 to 2022. She is currently a Postdoctoral Research Fellow with the Vision and Image Processing Laboratory (VIP), Department of Systems Design Engineering, University of Waterloo, Canada. She has been involved in different research projects with WASP, EU-FET WYSIWYD, Microsoft, and BIOSCAN. Her research interests include artificial intelligence, machine learning, and computational cognitive science.

PAUL FIEGUTH (Senior Member, IEEE) received a BASc in Electrical Engineering at the University of Waterloo and a PhD from the Massachusetts Institute of Technology (MIT). He has been a member of the faculty at the University of Waterloo in Systems Design Engineering since 1996, where he has been Associate Chair Undergraduate, Department Chair, Associate Dean and, since 2023, Associate Vice President. His research interests include statistical signal and image processing, hierarchical algorithms, data fusion, machine learning, and the interdisciplinary applications of such methods. He has significant pedagogical interests in the area of complex systems, specifically developing a much deeper understanding among engineering students on the impact of complex systems in many areas of engineering decision making. He is the author three textbooks: a 2010 text on Statistical Image Processing & Multidimensional Modeling, a 2021 text on Complex Systems, and a 2022 text on Pattern Recognition.