用于视频目标分割的稳健高效记忆网络

1st YadangChen计算机科学学院 南京信息工程大学中国，南京

2nd DingweiZhang南京信息工程大学软件学院中国南京

3rd Zhi-XinYang澳门大学智慧城市物联网国家重点实验室中国澳门

4th EnhuaWu中国科学院大学 计算机科学国家重点实验室中国，北京

arXiv:2304.11840v1 [cs.CV] 24 Apr 2023

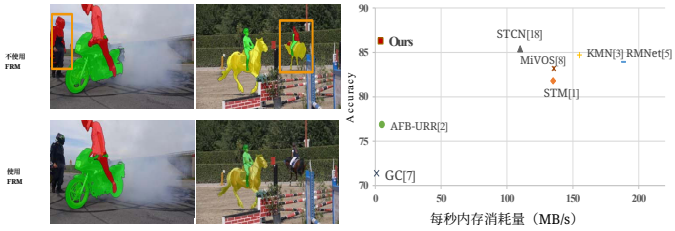
摘要—本文提出了一种用于研究半监督视频目标分割（VOS）的稳健高效记忆网络，简称 REMN。近年来，基于记忆的方法通过在查询与记忆之间执行非局部逐像素匹配，在 VOS 任务中取得了出色的性能。然而，这些方法存在两个局限性。1）非局部匹配可能导致背景中的干扰物体被错误分割。2）具有高度时间冗余的记忆特征会消耗大量计算资源。针对局限性 1，我们引入了一种局部注意力机制，通过利用前一帧掩码增强前景物体的特征，从而应对背景干扰。针对局限性 2，首先，我们根据前景物体的变化自适应地决定是否更新记忆特征，以减少时间冗余。其次，我们采用动态记忆库，该记忆库使用轻量且可微的软调制门来决定在时间维度上需要移除多少记忆特征。实验表明，我们的 REMN 在 DAVIS 2017 上取得了最先进的结果，$\mathcal{J}\&\mathcal{F}$ 得分为 86.3%；在 YouTube-VOS 2018 上，$\mathcal{G}$ 平均得分为 85.5%。此外，我们的网络具有较高的 25+ FPS 推理速度，并且仅使用相对较少的计算资源。

索引词—视频目标分割、时空记忆网络、背景干扰、时间冗余

I. 引言

视频目标分割（VOS）是计算机视觉中的一项具有挑战性的任务，在自动驾驶和视频编辑领域受到了广泛关注。与图像分割相比，VOS 的关键问题在于充分利用帧之间的时空线索。本文聚焦于半监督视频目标分割：给定第一帧中前景物体的标注，对剩余视频序列中的前景物体进行分割。

最近，基于记忆的方法——时空记忆网络（STM） [1]提出的方法已广泛应用于半监督视频目标分割（semi-VOS）。STM 及其变体 [2]–[17] 通过点积运算，在查询帧（即当前帧）与记忆库中的模板之间进行时空匹配；这些模板是记忆帧（即首帧和已处理帧）的特征。然而，这些方法采用的点积运算记忆覆盖率较低，并且需要为每个对象构建记忆库来存储特征，导致分割效率低下。为此，GC[7] 开发了一种紧凑的全局表示，以降低计算复杂度。此外，时空对应网络（STCN） [18] 采用效率更高的 L2 相似度取代点积运算，计算与目标无关的帧间对应关系，并让所有对象共享同一个记忆库。



(a) (b)

图 1：基于记忆的方法存在的两项局限性。（a）如第一行所示，与前景相似的背景对象被错误地分割。第二行的结果表明，我们提出的前景增强模块（FRM）能够有效抑制背景干扰。（b）基于记忆的方法在 DAVIS2017 验证集上的准确率和记忆消耗。大多数方法以牺牲计算资源为代价来获得更好的性能。

尽管上述方法在 VOS 基准测试中取得了最先进的性能，[19]，[20]，仍有两个局限性尚未解决。首先，如图 1(a) 所示，背景中的干扰物体被错误地分割为前景。原因在于 STM 使用的点积和 STCN 提出的 L2 相似度，均在未考虑时间一致性的情况下进行非局部逐像素匹配。

本工作部分得到了中国国家自然科学基金（项目编号：61802197、62072449 和 61632003）、澳门特别行政区科学技术发展基金（项目编号：0018/2019/AKP 和 SKL-IOTSC(UM)-2021-2023）、广东省科学技术厅（项目编号：2020B1515130001）以及澳门大学（项目编号：MYRG2020-00253-FST、MYRG2022-00059-FST）的资助。

一般而言，前景及其周围的像素在匹配中更有价值，而其他像素的贡献较小。尽管一些方法 [3], [5], [6], [21],[22] 通过双向匹配（例如 KMN [3], 、CFBI[6]）或光流模型（例如 RMNet [5]）限制匹配空间，以减轻背景干扰，但这会引入大量参数，增加计算负担。其次，许多时间冗余度高的特征被存储在记忆库中，导致大量消耗计算资源。如图 1(b) 所示，大多数方法[1], [3], [5], [8], [18] 以牺牲计算资源为代价来提升性能。即使采用固定的采样间隔，记忆特征之间仍存在较高的时间冗余。虽然 AFB-URR [2] 根据最不常用索引丢弃旧特征，GC [7] 则通过全局上下文保留固定数量的特征，但当前景物体被遮挡或发生形变时，这些方法会丢失关键特征。

我们通过在稳健高效记忆网络（REMN）中引入以下创新来解决上述局限性。针对第一个局限性，我们提出了前景增强模块（FRM），利用前一帧掩码提供的空间位置先验。不同于以往方法， [3], [5], [6], [22], FRM采用更轻量的局部注意力操作来增强查询帧中的前景特征，而不是额外引入模型来约束匹配空间。注意力图中的注意力权重表示前一帧中的每个物体像素与查询帧中对应位置局部区域内像素之间的空间相关性。图1(a)展示了FRM的有效性。针对第二个局限性，我们首先引入自适应采样模块（ASM），通过计算查询帧与最新记忆帧中物体掩码在形状和位置上的差异来确定采样间隔。尽管ASM能够通过控制采样间隔在一定程度上减少时间冗余，但随着视频时长增加，时间冗余仍然存在。因此，我们还采用了一个由冗余消除模块（RRM）组成的动态记忆库。RRM并非简单地丢弃过时特征，而是根据软调制门输出的时间策略自适应地减少特征冗余。RRM的特征分布与原始分布相似，从而避免丢失关键特征。因此，即使处理超长视频，我们的网络也能在推理时保持固定大小的记忆库，并实现快速推理。

我们的贡献总结如下：

- 我们提出了一种轻量级前景增强模块，以抑制非局部逐像素匹配造成的背景干扰。
- 我们提出了一种高效的自适应采样模块和冗余消除模块，以减少记忆特征的时间冗余。
- 我们的 REMN 在以下数据集上达到了业界领先的性能：DAVIS 2017 和 YouTube-VOS 2018，仅使用相对较少的计算资源，同时保持较高的 25+ FPS 推理速度。

II. 所提方法

图 2 展示了所提 REMN 的总体结构。与 STCN 类似，[18], 我们将当前帧作为查询帧，$F_t \in \mathbb{R}^{H^0 \times W^0 \times 3}$ 并将带有掩码的过去帧作为记忆帧，其中 H^0 和 W^0 表示当前帧的原始尺寸。Key Encoder 以一帧为输入，提取查询帧的查询键 $k'^Q \in \mathbb{R}^{C^k \times H \times W}$，其中 $H = \frac{H^0}{16}$ 和 $W = \frac{W^0}{16}$ 表示空间维度，C^k 表示键空间的维度。对于前一帧掩码 $M_{t-1} \in \mathbb{R}^{H^0 \times W^0 \times 1}$，我们将其下采样至 $m_{t-1} \in \mathbb{R}^{H \times W \times 1}$。为使查询帧更能抵御背景干扰，前景增强模块（FRM）为 k'^Q $\mathbb{R}^{H \times W \times C^k}$ M_{t-1} $k^Q \in$提供目标位置先验，以获得增强后的查询键。

记忆读取组件对 k^Q 与记忆帧的记忆键 $k^M \in \mathbb{R}^{T \times H \times W \times C^k}$ 进行非局部像素级匹配。随后，它将匹配结果与记忆值 $v^M \in \mathbb{R}^{T \times H \times W \times C^v}$ 相乘，得到查询帧的聚合读取特征 $v^Q \in \mathbb{R}^{H \times W \times C^v}$，其中 T 为记忆帧的数量。解码器对 v^Q 进行上采样，以生成预测掩码 $M_t \in \mathbb{R}^{H^0 \times W^0 \times 1}$。

随后，自适应采样模块（ASM）计算 M_t 与最新记忆帧掩码 $M_n \in \mathbb{R}^{H^0 \times W^0 \times 1}$ 之间的差异，以决定是否将 F_t 保留为记忆帧。由于所有记忆帧都曾是查询帧，我们重用 k^Q 作为该记忆帧的记忆键；接着，值编码器以图像帧和掩码为输入，提取记忆值。我们还设计了一个动态记忆库，最多可存储 N 对记忆键和值。此外，当记忆帧数量超过 N时，冗余消除模块（RRM）会自适应地从记忆库中移除冗余特征。

A. 前景增强模块

在本小节中，我们介绍图 3 所示的 FRM。该模块利用局部注意力机制，抑制非局部逐像素匹配所导致的背景干扰。

首先，我们将 m_{t-1} 与 k'^Q 拼接，得到特征 $F \in \mathbb{R}^{H \times W \times (C^k+1)}$。由于相邻帧之间的前景物体变化不大，对于前一帧中的每个像素，我们只需考虑该像素与查询帧中同一位置邻域之间的空间相关性。具体而言，我们通过一个卷积层 $Conv_{w \times h}$ 将每个像素的感受野限制在卷积核大小为 $w \times h$的范围内，从而生成局部注意力特征 $F_{att} \in \mathbb{R}^{H \times W \times C^k}$。接下来，通过一个 1×1卷积层 $Conv_{1 \times 1}$ 将 F_{att}的维度转换为 $H \times W \times Z$，其中$Z = w \times h$。

i，j中每个像素的向量表示为$F^{i,\ j} \in \mathbb{R}^Z$，经过softmax函数归一化后生成局部注意力权重 $\alpha^{i,\ j} \in \mathbb{R}^Z$。此外，我们利用前一帧掩码降低背景像素的权重

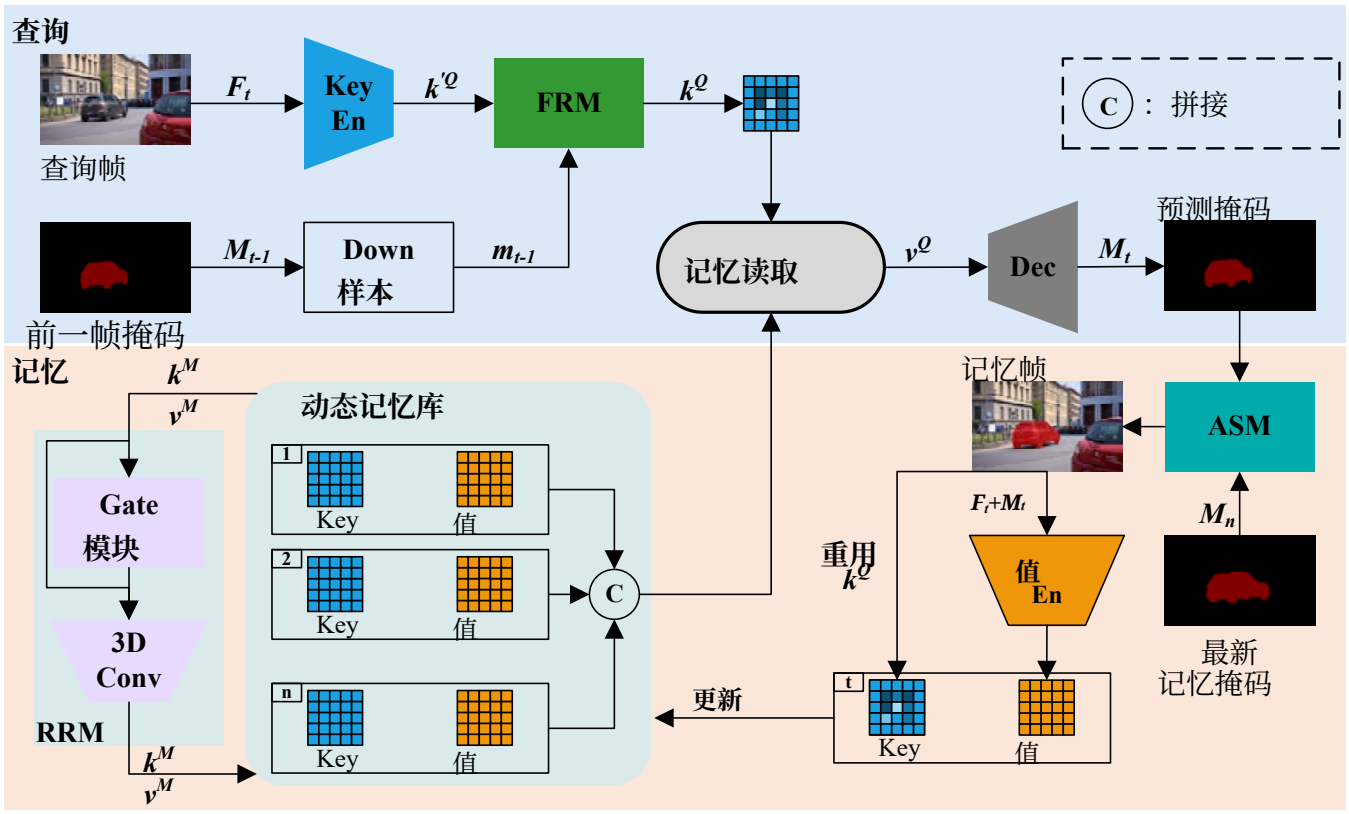


图 2：REMN 概览。我们沿用 STCN [18] 的结构，并提出一种用于应对背景干扰的前景增强模块（FRM）。自适应采样模块（ASM）用于决定是否将查询帧保存为记忆帧，而冗余特征去除模块（RRM）用于从记忆库中移除冗余特征。

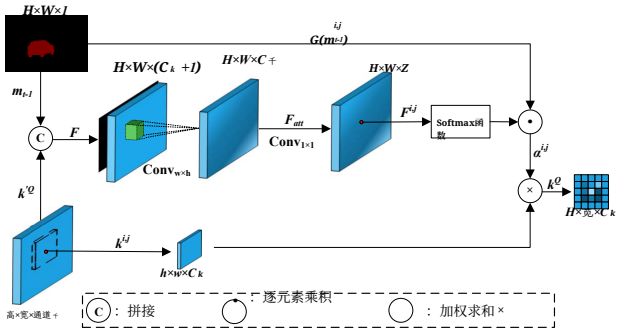


图 3：前景增强模块的实现。

相应位置的值为 $m^{i,\ j}_{t-1}$。计算定义为：

$$\alpha_p^{i,j} = \frac{\exp(F_p^{i,j})}{\sum_{m=1}^{Z}\exp(F_m^{i,j})} \odot G(m_{t-1}^{i,j}) \tag{1}$$

其中，$p \in \{1, \cdots, Z\}$ 和权重 $\alpha^{i,\ j}_{\ p}$ 表示查询帧中对应局部区域内的 p^{th} 像素相对于像素（i，j）的空间相关性。$\odot$ 表示逐元素乘积。$G(\cdot) = \exp(\cdot)\, e$ 使背景像素的权重趋近于 0，因为前一帧掩码不一定准确。最后，通过对每个权重及其对应的局部查询特征进行加权求和，得到增强查询键 k^Q 求和 $k^{i,\ j} \in \mathbb{R}^{h\times w\times C}$。我们将该过程描述如下：

$$k^Q = \sum_{i=1}^{H}\sum_{j=1}^{W}\alpha^{i,j}k^{i,j} \tag{2}$$

B. 自适应采样模块

在本小节中，我们介绍 ASM。与以往采用固定间隔将查询帧保存为记忆帧的工作不同，ASM 可根据前景物体的变化自适应地确定采样间隔。对于每个前景物体，我们通过计算查询帧与最新记忆帧之间的差异，决定是否更新记忆库。具体而言，对于物体 i^{th} 的预测掩码 M_t^i 及其最新记忆掩码 M_n^i，ASM 直接利用二者的交并比来衡量前景物体在位置和形状上的差异。该过程如下所示：

$$D^i = 1 - \frac{M_t^i \cap M_n^i}{M_t^i \cup M_n^i} \tag{3}$$

其中，D^i 表示 i^{th} 对象的变化率。一旦任何对象的变化率超过阈值σ，ASM 就会将查询帧视为新的记忆帧，并用它 M_t 替换 M_n，作为后续比较时最新的记忆掩码。

C. 冗余消除模块

在实践中，仅控制采样间隔不足以减少记忆库中的时间冗余。因此，我们引入了一种由 RRM 组成的动态记忆库，它可以动态移除冗余特征，只保留非冗余部分。在以往基于记忆的方法中，记忆读取的复杂度会随着记忆帧数量的增加而呈指数级增长。与这些方法不同，我们提出的 RRM 不仅能移除记忆库中的冗余特征，还能将复杂度控制在合适的范围内。

遵循 STCN [18], ，对于每个记忆帧，我们存储两项内容：记忆键和记忆值。当记忆帧的数量达到 N （即 $T = N$）时，软调制门以 k^M 为输入，并学习一个概率向量 $Prob \in \mathbb{R}^{S^t}$ ，用于确定沿时间维度保留的特征数量。S^t 是搜索空间大小（即时间策略的数量）。该过程总结如下：

$$Prob = \omega(\mathcal{F}(\beta, \delta(\mathcal{N}(\mathcal{F}(\lambda, \mathcal{A}(k^M))))) + \gamma) \tag{4}$$

其中， $\mathcal{A}$ 表示全局平均池化（GAP），其输出大小为 $T \times 1 \times 1 \times C^k$； $\mathcal{F}$ （$\cdot$, $\cdot$）是一个卷积函数； $\mathcal{N}$ 表示批归一化； λ、 β 和 θ 是卷积参数。 δ 和 ω 均表示激活函数，其中 δ 是 $Relu$ 激活函数，而 ω 定义为：

$$\omega(\cdot) = \max(0, Tanh(\cdot)) \tag{5}$$

使用所提出的激活函数 ω 后， $Prob$ 的取值范围为 $[0, 1)$，它表示选择每种时间策略的概率。随后，通过 $Argmax$ 函数找到 $Prob$ 中最大值的索引 S ，该索引对应软调制门输出的时间策略。

此外，对于 k^M ，我们使用时间步长为 2^{S+1} 的三维卷积，将 k^M 的大小从 $T \times H \times W \times C^k$ 转换为 $_{2^{S+1}}^{T} \times H \times W \times C^k$。同样，我们对 v^M 执行此操作，并将其大小更改为 $_{2^{S+1}}^{T} \times H \times W \times C^v$。

为确保 RRM 输出的特征能够高质量地进行分割，我们还引入了 RRM 损失函数：

$$\mathcal{L}_{RRM} = KL(k^M \| RRM(k^M)) + KL(v^M \| RRM(v^M)) \tag{6}$$

其中， RRM（$\cdot$）表示移除冗余特征的操作， KL（$\cdot\|\cdot$）表示 Kullback-Leibler 散度。通过 $\mathcal{L}_{RRM}$ ，我们使 RRM 输出的分布与原始特征的分布相似，以避免关键特征丢失和分割效果不佳。

D. 记忆读取与解码器

与 STCN [18], 一样，我们用亲和矩阵 W表示非局部逐像素匹配。该矩阵是对 k^M 和 k^Q之间的 L2 相似度进行 Softmax函数归一化后的结果。计算公式如下：

$$W(i,j) = \frac{\exp(-\|k_i^M - k_j^Q\|_2^2)}{\sum_i \exp(-\|k_i^M - k_j^Q\|_2^2)} \tag{7}$$

其中， i 和 j 分别是记忆键和查询键的位置。 $-\|\cdot\|_2^2$ 是负的平方欧氏距离，我们用它表示 L2 相似度。 W 可以从记忆库中检索出与查询帧最相似的像素特征位置。最后，将其与记忆库中的 v^M 相乘，得到聚合读取特征 v^Q:

$$v^Q = \sum_i W(i,j) \odot v_i^M \tag{8}$$

我们的解码器结构与 STM [1]解码器相似，由一个残差块和两个上采样块组成。解码器以记忆读取的输出 v^Q 作为输入，并在上采样块中将 Key Encoder 的低层特征进行跳跃连接，以生成预测掩码。

III. 实验

在本节中，我们在DAVIS2017 [19], 、YouTube-VOS 2018 [20],以及 [8]提出的一个长时数据集上评估我们提出的 REMN。常用的评估指标包括区域相似度 $\mathcal{J}$ 和轮廓准确率 $\mathcal{F}$。为使比较更加直观，我们使用它们的平均值 $\mathcal{J}\&\mathcal{F}$ 进行比较。此外，推理速度也是一个不可忽视的指标，尤其是在现实世界中，因此我们使用每秒帧数（FPS）来表示网络的推理速度。

A. 定量比较

如表I所示，与以往方法相比，REMN在DAVIS 2017上取得了具有竞争力的结果，同时拥有最快的推理速度（即 FPS为27.5）。在验证集上，REMN的 $\mathcal{J}\&\mathcal{F}$ 比我们的基线STCN [18],高1.0%，且推理速度快36%。此外，与我们的速度最为接近的GC [7] （FPS27.5对25.0）的平均准确率比我们低14.9%。特别是，如图1(b)所示，REMN在保持高准确率的同时消耗更少的计算资源。AFB-URR[2] 和GC [7] 与我们的网络一样，在推理过程中消耗恒定的计算成本，但当前景上下文失去跟踪时，它们的效果会降低。此外，REMN在test-dev集上的 $\mathcal{J}$ 也取得了最高得分76.5%。

表 I：DAVIS2017 验证集和 test-dev 集上的定量比较。粗体和下划线分别表示每列中的最佳和次佳结果。Gen-error 报告验证集与 test-dev 集之间的泛化差距。

方法	验证集			Test-dev 集			FPS	泛化误差
	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$		
STM [1]	81.8	79.2	84.3	72.2	69.3	75.2	6.3	9.5
AFB-URR [2]	76.9	74.4	79.3	-	-	-	6.8	-
KMN [3]	82.8	80.0	85.6	77.2	74.1	80.3	8.3	<u>5.6</u>
RMNet [5]	83.5	81.0	86.0	75.0	71.9	78.1	4.4	8.5
CFBI [6]	81.9	79.1	84.6	74.8	71.1	78.5	5.9	7.1
GC [7]	71.4	69.3	73.5	-	-	-	<u>25.0</u>	-
MiVOS [8]	84.5	81.7	87.4	78.6	74.9	<u>82.2</u>	11.2	6.0
STCN [18]	<u>85.3</u>	<u>82.0</u>	<u>88.6</u>	**79.9**	<u>76.3</u>	**83.5**	20.2	**5.4**
RENM（我们的方法）	**86.3**	**83.4**	**89.2**	<u>79.2</u>	**76.5**	81.9	**27.5**	7.1

我们将 YouTube-VOS 中的前景物体分为两类：训练阶段已见和未见的前景物体，并衡量网络在这两类物体之间的泛化能力

。表 II 展示了 REMN 与以往方法在 YouTube 2018 验证集上的定量比较结果。总的来说，REMN 不仅在平均准确率和推理速度方面达到了最先进的水平，还将泛化误差降至最低。

表 II：YouTube 2018 验证集上的定量比较。$\mathcal{G}$ 表示总体平均值。Gen-error 报告已见类别与未见类别之间的泛化差距。

方法	Seen		未见		$\mathcal{G}$	FPS	泛化误差
	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$			
STM [1]	79.7	84.2	72.8	80.9	79.4	-	5.1
AFB-URR [2]	78.8	83.1	74.1	82.6	79.6	-	2.6
KMN [3]	80.4	84.5	73.8	81.4	80.0	-	4.9
RMNet [5]	82.1	85.7	75.7	82.4	81.5	-	4.9
CFBI [6]	80.6	85.1	75.2	83.0	81.0	3.4	3.8
GC [7]	72.6	75.6	68.9	75.7	73.2	-	**1.9**
MiVOS [8]	81.1	85.6	77.7	86.2	82.7	-	2
STCN [18]	83.2	87.9	79.0	87.3	84.3	13.2	2.4
REMN（我们的方法）	**84.5**	**88.1**	**81.0**	**88.3**	**85.5**	**25.6**	**1.9**

为了验证我们的网络在长视频场景中的性能，我们使用了一个包含 3 个视频序列的长视频数据集，每个视频序列的平均长度超过 2000。然而，现有的大多数方法使用不受限制的记忆库，因此随着视频长度增加，存在 GPU 内存溢出的风险。我们通过限制这些方法的记忆更新频率，使其能够处理长时间视频，但这也导致这些方法的性能下降。如表 III 所示，与我们结果最接近的是 STCN；我们的网络准确率提升了 0.6%，速度快了 3.5 倍（FPS 25.3 对比 7.2）。我们还通过重复播放将该长视频的长度扩展至原来的 3 倍，此时 REMN 的 FPS 约为 STCN 的 8 倍（FPS 23.9 对比 3.0）。总之，REMN 在长视频数据集上取得了有竞争力的结果，并展现出出色的推理速度。

表 III：长视频数据集上的定量比较。$\times n$ 表示视频的帧数为 n 倍。$\times 1 \rightarrow \times 3$ 表示值的差异。

方法	视频长度（×1）		视频长度（×3）		×1 → ×3	
	$\mathcal{J}\&\mathcal{F}$	FPS	$\mathcal{J}\&\mathcal{F}$	FPS	$\mathcal{J}\&\mathcal{F}$	FPS
STM [1]	80.6	-	75.3	-	-5.3	-
AFB-URR [2]	83.7	-	83.8	-	0.1	-
CFBI [6]	53.5	-	58.9	-	**5.4**	-
STCN [18]	87.3	7.2	84.6	3.0	-2.7	-4.2
REMN（我们的方法）	**87.9**	**25.3**	**87.4**	**23.9**	-0.5	**-1.4**

B. 定性比较

在图 4 中，我们选取 DAVIS 2017 和 YouTube 2018 中的两个视频作为示例，并展示 REMN 与基线 STCN [18] 的定性比较结果。可以看出，我们提出的 REMN 具有很强的鲁棒性，并取得了出色的性能。

C. 消融研究

我们在 DAVIS 2017 和 YouTube 2018 的验证集上进行了消融实验。具体而言，我们报告了 DAVIS 2017 上的 $\mathcal{J}\&\mathcal{F}$，以及 YouTube 2018 上的 $\mathcal{G}$ 和 FPS，这些指标表示我们网络的结果。

表 IV：模块消融研究。

FRM	ASM	RRM	$\mathcal{J}\&\mathcal{F}$	$\mathcal{G}$	FPS
✓	✓	✓	86.3	85.5	25.6
	✓	✓	85.1	84.1	25.8
✓		✓	86.3	85.4	23.9
✓	✓		86.2	85.7	14.1
✓			85.6	84.7	12.9

如表 IV 所示，我们对 REMN 中的三个模块进行了消融实验，以分析它们的有效性。启用所有模块时（即采用默认配置），在 DAVIS 2017 和 YouTube 2018 上分别取得 86.3% 和 85.5% 的成绩，推理速度达到每秒 25.6 帧。禁用 FRM 后，性能分别下降 1.2% 和 1.4%。在图 5 中，我们选取了一个存在背景干扰的视频序列，以展示 FRM 的有效性。可视化结果表明，Key Encoder 提取的原始特征包含一个背景物体，这可能导致背景被错误地分割为前景。相比之下，FRM 通过抑制背景特征，避免了这些错误分割情况。

在内存管理方面，我们首先采用每隔 5 帧采样一次的策略来代替 ASM，使FPS 降低了 1.7。接着，我们停用 RRM，以保持记忆库不受限制。尽管采用此设置后，YouTube 2018 上的结果提升了 0.2%，但 FPS 仅为 14.1。最后，我们冻结 ASM 和 RRM（即使用 STCN 中的记忆库），使推理速度降至每秒 12.9 帧。出现这一结果是因为 STCN 使用了一个容量不受限制且包含大量冗余特征的记忆库。为了直观展示由 ASM 和 RRM 组成的动态记忆库的有效性，我们比较了 STCN 与我们的方法中记忆库内特征的分布。如图 6(a) 所示，许多点相互重叠，因此 STCN 的记忆库中特征冗余度很高。相反，图 6(b) 中的点数少于图 6(a)，且分布更均匀，这表明我们提出的 ASM 和 RRM 能有效去除冗余特征。

D. 局限性

我们提出，RRM 应及时移除记忆库中的冗余特征，但一些细节特征不可避免地会丢失。这将导致部分前景物体无法被正确分割，尤其是在前景物体的尺度较小时。我们希望在未来工作中，通过细化细尺度前景物体的特征来获得更好的性能。

参考文献

[1] S. W. Oh、J.-Y. Lee、N. Xu 和 S. J. Kim，《使用时空记忆网络进行视频目标分割》，载于*IEEE/CVF* 国际计算机视觉会议论文集，2019 年，第 9226–9235 页。[2] Y. Liang、X. Li、N. Jafari 和 J. Chen，《结合自适应特征库和不确定区域细化的视频目标分割》，载于神经信息处理系统进展，第 33 卷，第 3430–3441 页，2020 年。

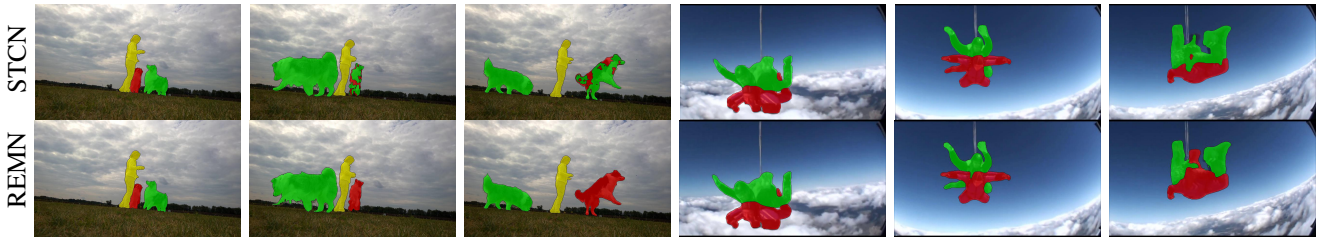


图 4：将 REMN 与基线 STCN [18] 进行比较所得的定性结果。

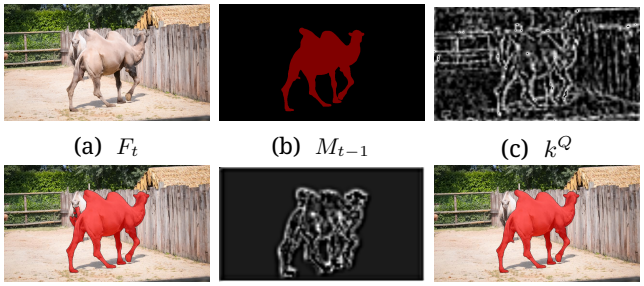
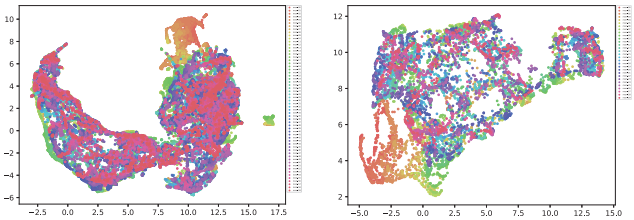


图 5：FRM 的有效性。(a) 查询帧。(b) 前一帧的前景物体掩码。(c) Key Encoder 生成的原始特征图。(d) 使用原始特征的结果。(e) FRM 生成的增强特征图。(f) 使用增强特征的结果。



(a) STCN [18]　(b) 我们的方法

图 6：与基线 STCN [18] 相比，我们提出的动态记忆库的可视化结果。不同颜色表示来自不同记忆帧的特征。不同颜色的点重叠越多，表示时间冗余越高。

[3] H. Seong、J. Hyun 和 E. Kim，《用于视频目标分割的核化记忆网络》，载于欧洲计算机视觉会议。Springer，2020 年，第 629–645 页。[4] Y. Chen、C. Hao、A. X. Liu 和 E. Wu，《基于多项事件模型的外观一致视频目标分割》，载于*ACM* 多媒体计算、通信与应用汇刊（*TOMM*），第 15 卷，第 2 期，第 1–15 页，2019 年。[5] H. Xie、H. Yao、S. Zhou、S. Zhang 和 W. Sun，《用于视频目标分割的高效区域记忆网络》，载于*IEEE/CVF* 计算机视觉与模式识别会议论文集，2021 年，第 1286–1295 页。[6] Z. Yang、Y. Wei 和 Y. Yang，《通过前景—背景整合实现协同视频目标分割》，载于欧洲计算机视觉会议。Springer，2020 年，第 332–348 页。[7] Y. Li、Z. Shen 和 Y. Shan，《使用全局上下文模块进行快速视频目标分割》，载于欧洲计算机视觉会议。Springer，2020 年，第 735–750 页。

[8] H. K. Cheng、Y.-W. Tai 和 C.-K. Tang，“模块化交互式视频目标分割：交互到掩码、传播与差异感知融合”，载于*IEEE/CVF* 计算机视觉与模式识别会议论文集，2021 年，第 5559–5568 页。[9] Y. Chen、C. Hao、Z.-X. Yang 和 E. Wu，“面向少样本视频目标分割的快速目标感知学习”，中国科学：信息科学，第 65 卷，第 8 期，第 1–182 104 页，2022 年。[10] Y. Chen、C. Hao、A. X. Liu 和 E. Wu，“基于监督优化的视频目标分割多层模型”，*IEEE* 多媒体汇刊，第 21 卷，第 8 期，第 1934–1945 页，2019 年。[11] L. Hu、P. Zhang、B. Zhang、P. Pan、Y. Xu 和 R. Jin，“面向基于记忆的视频目标分割的位置与目标一致性学习”，载于*IEEE/CVF* 计算机视觉与模式识别会议论文集，2021 年，第 4144–4154 页。[12] H. Seong、S. W. Oh、J.-Y. Lee、S. Lee、S. Lee 和 E. Kim，“ 用于视频目标分割的分层记忆匹配网络 ”，载于 *IEEE/CVF* 国际计算机视觉会议论文集，2021 年，第 12 889–12 898 页。[13] Z. Zhou、L. Ren、P. Xiong、Y. Ji、P. Wang、H. Fan 和 S. Liu，“用于视频分割的增强记忆网络”，载于*IEEE/CVF* 国际计算机视觉会议论文集（研讨会），2019 年，第 0–0 页。[14] S. Liang、X. Shen、J. Huang 和 X.-S. Hua，“结合动态记忆网络与自适应目标对齐的视频目标分割”，载于*IEEE/CVF* 国际计算机视觉会议论文集，2021 年，第 8065–8074 页。[15] K. Park、S. Woo、S. W. Oh、I. S. Kweon 和 J.-Y. Lee，“逐片段视频目标分割”，载于*IEEE/CVF* 计算机视觉与模式识别会议论文集，2022 年，第 1352–1361 页。[16] H. Wang、X. Jiang、H. Ren、Y. Hu 和 S. Bai，“SwiftNet：实时视频目标分割”，载于*IEEE/CVF* 计算机视觉与模式识别会议论文集，2021 年，第 1296–1305 页。[17] Z. Lin、T. Yang、M. Li、Z. Wang、C. Yuan、W. Jiang 和 W. Liu，“Swem：利用序列加权期望最大化实现实时视频目标分割”，载于*IEEE/CVF* 计算机视觉与模式识别会议论文集，2022 年，第 1362–1372 页。[18] H. K. Cheng、Y.-W. Tai 和 C.-K. Tang，“重新思考时空网络：通过改进记忆覆盖实现高效视频目标分割”，神经信息处理系统进展，第 34 卷，第 11 781–11 794 页，2021 年。[19] J. Pont-Tuset、F. Perazzi、S. Caelles、P. Arbeláez、A. Sorkine-Hornung 和 L. Van Gool，“2017 年 DAVIS 视频目标分割挑战赛”，*arXiv* 预印本 *arXiv:1704.00675*，2017 年。[20] N. Xu、L. Yang、Y. Fan、D. Yue、Y. Liang、J. Yang 和 T. Huang，“YouTube-VOS：大规模视频目标分割基准 ”，*arXiv* 预印本 *arXiv:1809.03327*，2018 年。[21] S. Cho 、 H. Lee 、 M. Lee 、 C. Park、S. Jang、M. Kim 和 S. Lee，“解决视频目标分割中的背景干扰”，*arXiv* 预印本 *arXiv:2207.06953*，2022 年。[22] P. Zhang、L. Hu、B. Zhang、P. Pan 和 D. Alibaba，“用于半监督视频目标分割的空间一致记忆网络”，载于*CVPR* 研讨会，第 6 卷，2020 年，第 2 页。