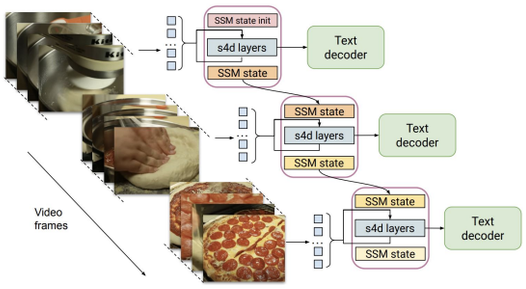
Google DeepMind

Time-Scaling State-Space Models for Dense Video Captioning

AJ Piergiovanni, Ganesh Mallya, Dahun Kim, Anelia Angelova

Introduction

- State-Space models with Transfer State (STS) where the state is maintained, reused and propagated across longer input sequences.
- We present efficient update rules for the state itself, and prove the equivalence of the Transfer State, verifying that the state will be preserved over long sequences.
- We show the effectiveness of this approach on dense video captioning datasets



Equivalence

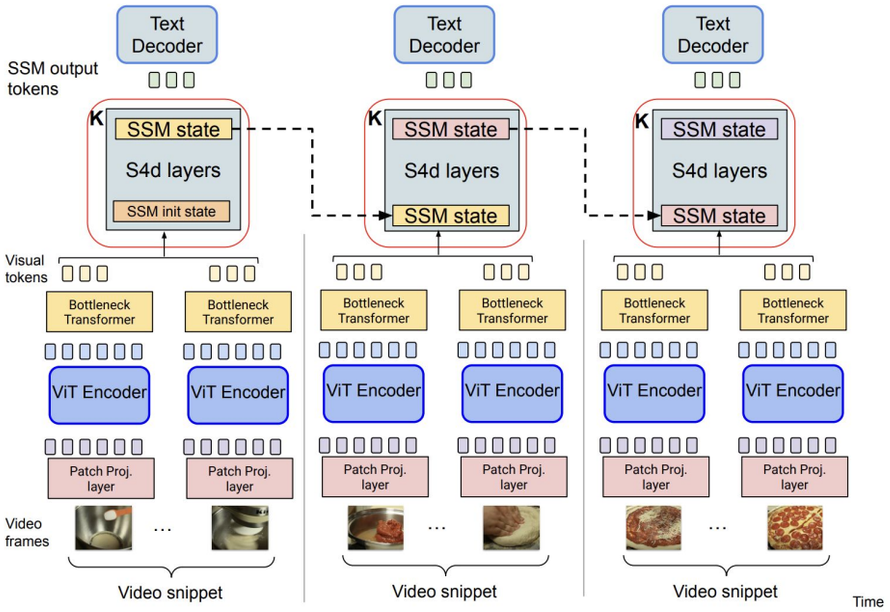
We show this splitting and updating is equivalent to a standard SSM.

$$h_k = \overline{A} * h_{k-1} + \overline{B} * x_k$$
$$y_k = \overline{C} * h_k + D * x_k$$

$$h_M = \overline{A}^{M-1}\overline{B} * x_1 + \overline{A}^{M-2}\overline{B} * x_2 + \ldots + \overline{B} * x_M.$$

Model

The model consists of a ViT-based encoder that encodes each frame into a set of tokens. Each of the per-frame tokens are compressed with a bottleneck transformer module. After this, the SSM runs on these tokens, and propagates the state across sequences, enabling long video training and evaluation.



$$h_L = \overline{B} * x_L + \overline{AB} * x_{L-1} + \overline{A}^{L-M-1}\overline{B} * x_{L-M+1} + \overline{A}^{L-M} * h_M$$
$$h_L = \overline{B} * x_L + \overline{AB} * x_{L-1} + \ldots + \overline{A}^{L-M-1}\overline{B} * x_{L-M+1} + \overline{A}^{L-M} * (\overline{A}^{M-1}\overline{B} * x1 + \ldots + \overline{B} * x_M)$$
$$h_L = \overline{B} * x_L + \overline{AB} * x_{L-1} + \ldots + \overline{A}^{L-M-1}\overline{B} * x_{L-M+1} + \overline{A}^{L-M}\overline{B} * x_{L-M+1} + \ldots + \overline{A}^{L}\overline{B} * x_1.$$

Results

STS performs at SOTA levels for dense video captioning on ViTT, YouCook2, and ActivityNet

Method	ViTT S	ViTT C	ViTT M	ViTT F1	YouCook2 S	YouCook2 C	YouCook2 M	YouCook2 F1	ActivityNet S	ActivityNet C	ActivityNet M	ActivityNet F1
UEDVC[24]	-	-	-	-	3.3	8.4	2.2	-	5.5	-	-	-
Masked[45]	-	-	-	-	-	6.1	3.2	-	-	9.3	5.0	-
PDVC[41]	-	-	-	-	4.9	28.9	5.7	-	6.0	29.3	7.6	-
E2ESG[54]	-	-	-	-	-	25.0	3.5	-	-	-	-	-
TimeCh[54]	-	-	-	-	3.4	11.0	-	19.5	-	-	-	-
OmniV[40]	-	-	-	-	-	-	-	-	-	26.0	7.5	-
Vid2Seq[74]	9.8	23.0	5.0	**37.7**	5.7	25.3	6.4	23.5	5.9	30.2	8.5	51.8
DIBS†[53]	-	-	-	-	6.4	**44.4**	7.5	31.4	5.9	31.9	8.9	**55.6**
SDC†[50]	**10.0**	25.2	5.8	35.4	6.0	32.9	7.1	24.1	6.2	**37.8**	10.0	52.9
GIT†[62]	7.1	15.1	3.4	32.5	3.1	12.1	3.4	17.7	5.7	29.8	7.8	50.6
VMS [0]	-	-	-	-	4.3	22.1	4.4	28.4	5.3	26.8	7.2	54.3
STS Ours	8.5	**42.4**	**19.6**	34.9	**9.7**	37.8	**18.7**	**37.7**	**8.9**	33.4	**16.0**	41.6

Compared to transformers and SSMs, it does better as well and uses fewer FLOPs than prior works.

		C	S	M	mIOU	F1
Global	Trans	15.0	5.1	6.3	38.1	27.4
Global	SSM	14.7	5.1	6.2	40.3	29.1
Online	STS	**16.1**	**6.1**	**14.2**	**50.4**	**32.3**

Model	GFLOPs
SDC [50]	12800
STS (Ours)	1854

STS allows using more frames

Compared to prior works, using STS allows ussing far more frames (256) than prior works, which can allow us to capture up to 315 seconds of video and more frames is better

# Frames	C	S	M	mIOU	F1
16	28.7	6.8	16.0	57.5	31.3
32	27.5	7.3	16.5	59.9	33.2
64	34.1	**7.8**	17.5	60.1	35.4
128	34.9	**7.8**	17.9	61.5	35.6
256	**35.0**	**7.8**	**18.0**	**62.0**	**35.9**

We also find that using last token per segment when captioning improves performance.

Method	Tokens	C	S	M	mIOU	F1
SSM Global	All Tokens (of sequence)	14.1	5.2	6.1	38.3	27.4
SSM Global	Last token per segment	14.7	5.1	6.2	40.3	29.4
SSM Global	Final Token (of sequence)	13.5	5.0	6.3	39.0	30.3
STS Online	All Tokens (of sequence)	14.7	5.8	11.0	**52.2**	30.1
STS Online	Last token per segment	**16.1**	**6.1**	**14.2**	50.1	**32.4**