6 Appendix

6.1 Proofs

In Sec. 3.1, we propose an updated method for the Memorandum and provide the simplification formula for the conclusion. For a clearer explanation, we derived the detailed process from Eq. 3 to Eq. 4 in this section. As shown in Eq. 3, the difference between $H(t+1)$ and $H(t)$ can be represented as:

$$H(t+1) - H(t) = \sum_{i=0}^{t+1} S_i e^{-\alpha(t+1-i)} - \sum_{i=0}^{t} S_i e^{-\alpha(t-i)} \tag{14}$$

which can be further converted into:

$$H(t+1) - H(t) = S_{t+1} e^{-\alpha(t+1-(t+1))} + \sum_{i=0}^{t} S_i e^{-\alpha(t+1-i)} - \sum_{i=0}^{t} S_i e^{-\alpha(t-i)} \tag{15}$$

$$= S_{t+1} + \sum_{i=0}^{t} S_i e^{-\alpha(t+1-i)} - \sum_{i=0}^{t} S_i e^{-\alpha(t-i)} \tag{16}$$

$$= S_{t+1} + \sum_{i=0}^{t} S_i e^{-\alpha(t+1-i)} - S_i e^{-\alpha(t-i)} \tag{17}$$

$$= S_{t+1} + \sum_{i=0}^{t} S_i e^{-\alpha(t-i)} (e^{-\alpha} - 1) \tag{18}$$

$$= S_{t+1} + (e^{-\alpha} - 1) \sum_{i=0}^{t} S_i e^{-\alpha(t-i)} \tag{19}$$

combined with Eq. (2), Eq. (19) can be simplified to:

$$H(t+1) - H(t) = S_{t+1} + (e^{-\alpha} - 1) H(t) \tag{20}$$

which can be further simplified as:

$$H(t+1) = e^{-\alpha} H(t) + S_{t+1} \tag{21}$$

From Eq. 14 to Eq. 21, we have proved the derivation process from Eq. 3 to Eq. 4 in Sec. 3.1. With Eq. 21, the Memorandum can be updated through the statement of the last moment and the statistical matrix of current moment iteratively.

Table 5. The recall of each predicate in SGCls. The baseline model without SS and SA is MOTIFS-TDE, and the MOTIFS-TDE with both SS and SA is our proposed CSS. The columns of *Fre* are the frequency of the predicates' occurrence.

Predicate	R@100 - SGCls				Fre/k	Predicate	R@100 - SGCls				Fre/k
SS		✓		✓		SS		✓		✓	
SA			✓	✓		SA			✓	✓	
on	20.18	17.74	26.60	23.79	712	covering	19.46	9.13	17.82	19.26	4
has	48.08	47.36	50.27	46.01	277	laying on	11.97	20.41	10.77	11.90	4
in	16.62	16.16	19.36	18.04	251	playing	0.0	0.0	0.0	0.0	4
of	46.95	36.87	45.05	45.30	146	against	0.0	0.0	0.0	0.0	3
wearing	48.45	48.24	50.94	51.49	136	along	21.64	27.45	20.41	19.95	3
near	4.73	4.90	15.27	17.66	96	and	0.0	0.0	0.0	7.1	3
with	8.57	8.48	6.86	12.50	66	belonging	0.0	28.42	4.80	3.47	3
above	18.00	17.92	16.79	15.85	47	between	0.0	36.04	0.0	0.0	3
holding	24.80	22.79	15.85	29.68	42	from	0.0	0.0	0.0	0.0	3
behind	38.12	6.32	41.2	40.14	41	looking at	6.86	12.38	9.17	9.30	3
under	18.46	17.60	20.71	20.47	23	painted on	0.0	0.0	0.0	0.0	3
flying in	0.0	0.0	0.0	0.0	20	watching	14.55	18.16	18.06	15.25	3
sitting on	24.50	26.51	25.08	27.07	18	across	0.0	0.0	0.0	0.0	2
wears	0.0	2.2	0.78	0.85	15	covered in	29.88	26.80	29.82	35.06	2
standing on	12.29	16.61	12.49	15.25	14	lying on	0.0	0.0	0.0	0.0	2
in front of	19.18	27.03	18.38	18.64	13	made of	0.0	0.0	0.0	0.0	2
at	31.13	31.76	31.34	33.25	10	mounted on	0.0	0.0	0.0	0.0	2
attached to	5.58	7.76	2.52	8.89	10	on back of	0.0	0.0	0.0	0.0	2
hanging from	10.16	22.42	7.9	17.60	10	parked on	62.04	62.60	62.13	60.56	2
for	4.92	9.13	6.53	11.24	9	part of	0.0	0.0	0.0	0.0	2
over	3.14	8.40	5.98	9.62	9	says	0.0	0.0	0.0	0.0	2
riding	45.44	45.70	46.54	45.99	9	to	0.0	0.0	0.0	0.0	2
carrying	31.40	36.49	28.22	29.73	5	using	15.0	26.73	26.63	26.56	2
eating	28.79	26.80	23.49	22.72	5	walking in	0.0	0.0	0.0	0.0	1
walking on	39.62	44.45	42.60	35.59	5	growing on	0.0	0.0	0.0	0.0	1

6.2 Relationship Retrieval

We supplement the experimental results of the recall of each predicate in SGCls and rank the predicates by their frequency of occurrence. As illustrated in Tab. 5, the SS module mainly improves the recall of the tail predicates, e.g., 84.9% relative gain of *using*, 24.8% relative gain of *watching*, 32.7% relative gain of *looking at* and so on, which is better than SA. Meanwhile, the SA module promotes both the head predicates, e.g., 32.1% relative gain of *on*, 4.6% relative gain of *has*, 16.5% relative gain of *in*, and the tail predicates, e.g., 77.5% relative gain of *using*, 24.1% relative gain of *watching*, 33.7% relative gain of *looking at*, and so on. Compared with the submodules, the recall in head predicates of the proposed CSS is better than SS module, and the recall in tail predicates of CSS is better than SA module, which means that CSS achieves a more balanced result. This is in accordance with the law of mR@K and R@K in Tabs. 1 and 2 which summarized in Sec. 4.4.

Table 6. The results of mean Recall@K between different encoding granularity.

	Predicate Classification			Scene Graph Classification			Scene Graph Detection		
Granularity	R@20	R@50	R@100	R@20	R@50	R@100	R@20	R@50	R@100
5×5	19.95	26.08	28.50	11.08	14.15	15.48	6.38	8.90	10.77
8×8	19.97	26.11	28.53	11.09	14.19	15.53	6.38	8.91	10.80
13×13	19.94	26.10	28.52	11.06	14.16	15.50	6.37	8.87	10.74

Table 7. The results of Recall@K between different encoding granularity.

	Predicate Classification			Scene Graph Classification			Scene Graph Detection		
Granularity	R@20	R@50	R@100	R@20	R@50	R@100	R@20	R@50	R@100
5×5	42.02	53.09	56.53	26.21	31.74	33.53	13.30	18.56	22.36
8×8	42.04	53.13	56.56	26.24	31.78	33.57	13.32	18.59	22.40
13×13	41.99	53.08	56.50	26.20	31.71	33.51	13.31	18.58	22.40

6.3 Encoding Granularity

We evaluated the influence of the granularity of the relative position encoding in the SA module. The granularity is set to be 5×5, 8×8 and 13×13. As illustrated in Tabs. 6 and 7, the promotion of both mR@K and R@K with the increase of the encoding granularity is limited. However, the increased cost of time and resources is unacceptable. Therefore, we chose to sacrifice the performance moderately in exchange for the improvement of training speed, and we used the 5×5 encoding in all other experiments.