

Supplementary materials to “Robust Image hashing based on Contrastive Masked Autoencoder with Weak-Strong Augmentation Alignment”

Cundian Yang¹, Guibo Luo¹, Yuesheng Zhu^{1 *}, Jiaqi Li², Xiyao Liu^{2 *}

¹ Guangdong Provincial Key Laboratory of Ultra High Definition Immersive Media Technology, Shenzhen Graduate School, Peking University

² School of Computer Science and Engineering, Central South University
yangcundian@stu.pku.edu.cn, {luogb, zhuys}@pku.edu.cn, {jiaqi-li, lxyzoewx}@csu.edu.cn

This supplementary material serves as an extension to our main paper, offering a deeper dive into the implementation details for reproducibility, as well as additional experimental results that were omitted due to page limitations. We begin by delineating the network architecture and meticulous training settings of our proposed CMAA. Subsequently, we present the results on UCID. And then we conduct a comprehensive ablation study that dissects the contributions of key components and the impact of hyperparameter tuning. Finally, we compare the time complexity of different methods.

Implementation details

Network architecture

We use ViT-Small (Dosovitskiy et al. 2020) as our encoder backbone E and our momentum encoder backbone E_m . After ViT-Small, we use linear projection layers, which include three Fully-Connected (FC) layers. The outputs of the first two FC layers are 4096 dimensions, and the first two lay-

*Corresponding Authors.

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

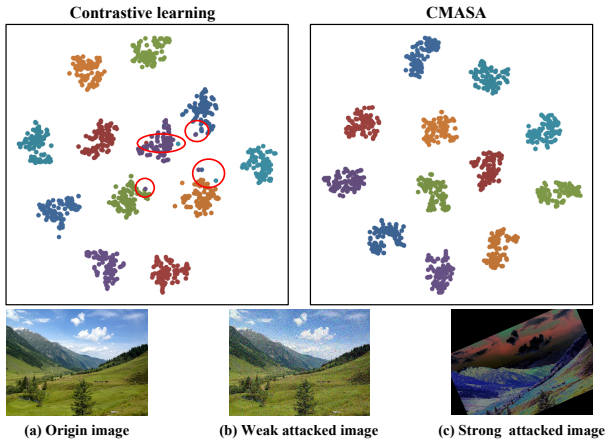


Figure 1: The t-SNE visualization of contrastive-learning features and CMAA features. The features of contrast learning have more outliers compared to the features of CMAA.

Table 1: Discrimination metrics in terms of NHD_d , where τ represents the threshold of NHD_d given false positive rate to 1%, μ represents the average values of NHD_d , and σ represents the standard deviation values of NHD_d .

	ProtoHash	TMICH	Swin-T	GLIF	ISM	Ours
$\tau \uparrow$	0.1055	<u>0.3520</u>	0.3570	0.2125	0.2188	0.3203
$\mu \uparrow$	0.4690	0.4655	0.4698	0.3906	0.4976	<u>0.4941</u>
$\sigma \downarrow$	0.1854	<u>0.0391</u>	0.0342	0.0745	0.1164	0.0560

ers include BatchNorm and ReLU. The last FC layer projects the 4096-dimension features to 128 dimensions. And our decoder architecture is designed according to (He et al. 2022), it has 8 blocks and a width of 512.

Training detail

To train the CMAA, we use the SGD optimizer (Bottou 2010) with an initial learning rate of 0.01, a weight decay of $1e^{-4}$, a momentum of 0.9. We also use the cosine scheduler (Loshchilov and Hutter 2016) to gradually decay the learning rate to 0. And the batch size is set to 256. To avoid collapsing, the shuffle BN (He et al. 2020) is also adopted. Typically, the experiment with the strong augmentations for each training image (CMAA) takes roughly 10 hours to finish on a machine with four NVIDIA GeForce RTX 3090 GPUs.

Results on UCID dataset

In Table 1, similar to the main paper’s finding, CMAA demonstrates competitive discrimination performance. Notably, the average NHD_r values for CMAA, as depicted in Table 2, are consistently lower than those of other compared methods. This indicates a favorable balance between discrimination and robustness. Consequently, as illustrated in Table 3, our method leads in F1 score, underscoring its superior content identification capabilities.

To further demonstrate the superiority of CMAA on UCID dataset, we plot detection error tradeoff (DET) curves, which chart the false negative rate (P_{fn}) against various false positive rates (P_{fp}). As evident in Figure 2, the curves of CMAA are continuously and significantly below those of the other methods for UCID dataset, confirming the superior overall performance. Additionally, Figure 3 presents select

Table 2: Robustness metric in terms of NHD_r (\downarrow) for UCID dataset.

Attack	ProtoHash	TMICH	Swin-T	GLIF	ISM	Ours
BA	0.1831	0.0589	0.0764	0.2250	0.0401	0.1313
CA	0.0944	0.0595	0.0806	0.2278	0.1366	0.1404
ST	0.0689	0.0294	0.0372	0.2219	0.0767	0.0625
SH	0.0056	0.0176	0.0275	0.2171	0.0179	0.0050
AF	0.0381	0.1862	0.2006	0.2239	0.0951	0.0563
MF	0.0327	0.1836	0.2135	0.2296	0.0683	0.0350
GF	0.0575	0.2284	0.2381	0.2266	0.1237	0.0867
GN	0.0811	0.2884	0.3004	0.2586	0.3561	0.1517
SN	0.0432	0.1894	0.2045	0.2373	0.1502	0.0924
SPN	0.0459	0.2386	0.2713	0.2388	0.3004	0.0769
JC	0.0122	0.1583	0.2057	0.2232	0.1493	0.0208
PT	0.0269	0.0443	0.0673	0.2196	0.0889	0.0386
RS	0.0067	0.0315	0.0486	0.2188	0.0185	0.0035
RT	0.1998	0.2889	0.2674	0.3661	0.3844	0.1209
FL	0.2242	0.1834	0.1716	0.2823	0.3666	0.0288
SHR	0.1432	0.1629	0.1518	0.3309	0.2867	0.0983
TR	0.2053	0.1775	0.1669	0.3251	0.3796	0.1350
RT + JC	0.1818	0.0654	0.3743	0.3743	0.4318	0.1201
RS + BA	0.1405	0.2429	0.2193	0.2193	0.0317	0.1307
TR + GN	0.1596	0.2933	0.3259	0.3259	0.4469	0.1898
FL + SH	0.2127	0.1840	0.2826	0.2826	0.3668	0.0282
GF + PT + CA	0.0624	0.3272	0.2363	0.2363	0.2329	0.1298
ST + RT + SHR	0.2228	0.2395	0.3828	0.3828	0.4371	0.1583
SH + TR + SN	0.1554	0.3163	0.3242	0.3242	0.3669	0.1455
RS + SPN + AF	0.0543	0.3338	0.2318	0.2318	0.1996	0.1455
Average	0.1136	0.1879	0.2044	0.2776	0.2426	0.0984

instances of successful and unsuccessful content identification, visually highlighting the method’s resilience against a spectrum of attack types, including hybrid attacks.

Ablation study

To further demonstrate the effectiveness of CMAA, we randomly select 12 images from the CASIA dataset, and then use CMAA and a network trained based on contrastive learning represented as (a) in Table 8 of the main paper, to extract the features of these 12 original images and their attacked images. The attacks are the same as those used in the main paper to test the robustness and content identification performance of CMAA. Finally, we use T-distribution random neighbor embedding (T-SNE) to visualize these features. As

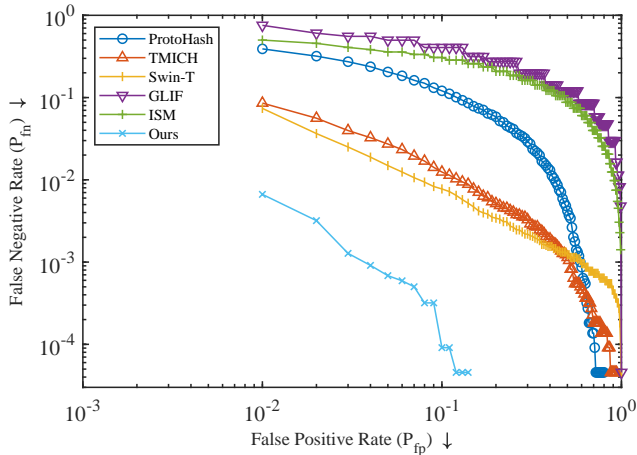


Figure 2: DET graphs on UCID dataset for the different schemes.

Table 3: Content identification performance in terms of F_1 score (\uparrow) for UCID dataset.

Attack	ProtoHash	TMICH	Swin-T	GLIF	ISM	Ours
BA	0.5235	1.0000	1.0000	0.6263	0.9944	0.9976
CA	0.8579	1.0000	1.0000	0.6011	0.9031	0.9976
ST	0.9334	1.0000	1.0000	0.6458	0.9870	1.0000
SH	1.0000	1.0000	1.0000	0.6699	1.0000	1.0000
AF	0.9839	0.9936	0.9963	0.6335	0.9626	1.0000
MF	0.9937	0.9887	0.9806	0.6075	0.9824	1.0000
GF	0.9559	0.9563	0.9502	0.6251	0.9242	0.9982
GN	0.8566	0.8303	0.8638	0.4476	0.3352	0.9724
SN	0.9761	0.9982	0.9926	0.5530	0.8512	0.9973
SPN	0.9644	0.9629	0.9500	0.5476	0.5150	0.9991
JC	0.9993	0.9988	0.9967	0.6168	0.8752	1.0000
PT	0.9984	1.0000	1.0000	0.6418	0.9279	1.0000
RS	1.0000	1.0000	1.0000	0.6602	1.0000	1.0000
RT	0.4802	0.8607	0.9168	0.0766	0.1821	1.0000
FL	0.4282	0.9567	0.9812	0.3717	0.3588	1.0000
SHR	0.7142	1.0000	1.0000	0.1727	0.4793	0.9997
TR	0.4514	1.0000	1.0000	0.2113	0.2093	0.9982
RT + JC	0.3374	1.0000	0.8512	0.0653	0.0877	1.0000
RS + BA	0.5308	0.9125	1.0000	0.6602	1.0000	1.0000
TR + GN	0.4411	0.7962	0.8550	0.1839	0.0512	0.9858
FL + SH	0.3375	0.9578	0.9802	0.3644	0.3435	1.0000
GF + PT + CA	0.9475	0.8089	0.8214	0.5748	0.7070	0.9988
ST + RT + SHR	0.1910	0.9976	0.8860	0.0399	0.1008	0.9970
SH + TR + SN	0.4512	0.8581	0.9951	0.2018	0.2051	0.9982
RS + SPN + AF	0.9165	0.7594	0.6916	0.5953	0.7881	0.9677
Average	0.7168	0.9455	0.9542	0.4080	0.5742	0.9965

Table 4: Effect of key components on CMAA for CASIA dataset. (a) in the current table and (a) in Table 8 of the main paper have the same experimental setup. Note that NHD_r is the mean value of NHD under different parameters within one type of attack, and the same applies to F_1 score.

attacks	$\mathcal{L}_C + \mathcal{L}_Q$ (a)		Ours	
	NHD_r	F_1 score	NHD_r	F_1 score
RT 50	0.2038	0.9937	0.1230	1.0000
RT 60	0.2285	0.9937	0.1276	1.0000
RT 70	0.2526	0.9610	0.1377	1.0000
RT 80	0.2691	0.9262	0.1512	1.0000
RT 90	0.2843	0.8966	0.1600	1.0000
FL horizontal	0.1655	0.9873	0.0212	1.0000
FL vertical	0.2461	0.9543	0.0328	1.0000
RT 50 + JC 50	0.2049	0.9937	0.1245	1.0000
RT 50 + JC 70	0.2046	0.9937	0.1223	1.0000
TR 0.2 + GN 0.05	0.2947	0.9041	0.2219	0.9474
RS 2.0 + BA 0.8	0.3113	0.8489	0.1493	0.9937
RS 2.0 + BA 0.9	0.1686	1.0000	0.0848	1.0000

shown in Figure 1, there are many outliers in the visualization of the features of (a), which affects the robustness and discrimination of the hash. The results in Table 4 also reveal this problem. When facing large-scale attacks (attacks with large rotation angles) and hybrid attacks, the NHD_r of (a) is much greater than that of CMAA, the F_1 score of (a) is lower than that of CMAA. This further illustrates the superiority of our CMAA. Additionally, NHD_d is the average of NHD among different images, NHD_r is the average of mean NHD over all attacks, F_1 score is the average of F_1 score over all attacks.

We select the optimal hyperparameters for CMAA that yield the best content identification performance according to the results shown in Table 5. The hyperparameters are finally selected in our experiments are marked in gray. It is worth noting that CMAA can still achieve competitive results on shorter hash sequences (64-bits). However, in or-


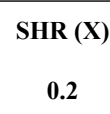







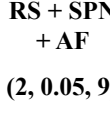
Original image	Attacked image	Attack param	ProtoHash	TMICH	Swin-T	GLIF	ISM	Ours
		SHR (X)	0.1016	0.1890	0.1650	0.1625	0.2656	0.1094
		0.2	✓	✓	✓	✓	✗	✓
		RT	0.2148	0.2780	0.2240	0.3875	0.3438	0.0781
		40°	✗	✓	✓	✗	✗	✓
		TR + GN	0.1602	0.4030	0.3370	0.3375	0.4063	0.2266
		(0.2, 0.05)	✗	✗	✓	✗	✗	✓
		GF + PT + CA	0.0820	0.2750	0.4090	0.2625	0.0469	0.1484
		(9, 8, 1.2)	✓	✓	✗	✗	✓	✓
		RS + SPN + AF	0.1328	0.2560	0.2850	0.2750	0.1719	0.0859
		(2, 0.05, 9)	✗	✓	✓	✗	✓	✓
Threshold of NHD_d			0.1055	0.3520	0.3570	0.2125	0.2188	0.3203

Figure 3: Examples of successful and failed content identification on UCID dataset. Check-mark (✓) indicate successful identification and cross-mark (✗) denote failure. The numerical values below the check-mark (✓) or cross-mark (✗) represents the NHD between the attacked image and the original image.

Table 5: Effect of hyper parameters on CMAA for CASIA dataset. The best results in each column are highlighted in **bold**. Default hyperparameters are marked in gray.

hyperparameter		NHD _d ↑	NHD _r ↓	F ₁ score ↑
mask ratio (m)	10%	0.4910	0.1054	0.9915
	30%	0.4947	0.0987	0.9946
	50%	0.4911	0.1018	0.9915
temperature (t)	0.08	0.4953	0.0997	0.9913
	0.1	0.4947	0.0987	0.9946
	0.12	0.4907	0.0912	0.9943
memory bank (K)	1024	0.4927	0.0911	0.9933
	2048	0.4947	0.0987	0.9946
	4096	0.4959	0.0101	0.9937
hash length (l)	64	0.4972	0.1045	0.9905
	128	0.4947	0.0987	0.9946
	256	0.4945	0.1005	0.9933

der to pursue optimal content identification performance, we choose a hash sequence scheme with a length of 128.

In addition, we also compared the performance of CMAA on the CASIA dataset using ViT-small, ResNet-18, and VGG as the backbone architectures, while maintaining our experimental setting. The results also shown in the table below which further demonstrates the superiority of ViT architecture.

Table 6: Attacks for generalization evaluation.

attack	details/parameters
BA	0.6, 0.95, 1.05, 1.4
CA	0.6, 0.95, 1.05, 1.4
ST	0.6, 0.95, 1.05, 1.4
SH	0.6, 0.95, 1.05, 1.4
AF	11×11, 13×13
MF	11×11, 13×13
GF	11×11, 13×13
GN	0.06, 0.25
SN	0.06, 0.25
SPN	0.06, 0.25
JC	5, 45, 95
PT	2, 3
RS	0.4, 0.8, 1.75, 2.25
RT	-150, -120, -100, 100, 120, 150
SHR	-0.4, 0.4
TR	-0.4, 0.4
RT + JC	(70, 50), (90, 50), (120, 50)
RS + BA	(0.5, 0.6), (2.0, 0.6)
TR + GN	(0.2(X/Y), 0.25), (0.3(X/Y), 0.2)
FL + SH	(Horizontal, 0.6), (Vertical, 0.6)
GF + PT + CA	(11×11, 4, 1.3), (9×9, 8, 1.4)
ST + RT + SHR	(0.7, 120, 0.2(X/Y)), (0.8, 120, 0.3(X/Y))
SH + TR + SN	(1.4, 0.1(X/Y), 0.1), (1.2, 0.2(X/Y), 0.05)
RS + SPN + AF	(0.4, 0.05, 9×9), (2.5, 0.05, 11×11)
Crop from image edge (CIE)	5%, 10%
Crop from image center (CIC)	5%, 10%
Random Affine (RA)	degrees=[10, 20], translate=[0.1, 0.2], scale=[0.75, 0.95]

Generalization evaluation

We evaluate the generalization ability of CMAA in terms of robustness and content identification performance by per-

Table 7: Robustness metrics in terms of NHD_r and content identification performance in terms of F_1 score when tested with stronger attacks.

attack	CASIA		Copydays	
	NHD_r	F_1 score	NHD_r	F_1 score
BA	0.1193	0.9853	0.1316	0.9840
CA	0.1115	0.9904	0.1198	0.9944
ST	0.0748	1.0000	0.0693	1.0000
SH	0.0092	1.0000	0.0045	1.0000
AF	0.1713	0.9670	0.0197	1.0000
MF	0.1405	0.9911	0.0231	0.9987
GF	0.2223	0.8868	0.0385	1.0000
GN	0.2064	0.9178	0.1658	0.9925
SN	0.0942	0.9979	0.1164	0.9914
SPN	0.1259	0.9825	0.1208	0.9935
JC	0.0583	0.9954	0.0579	0.9879
PT	0.1852	0.9563	0.2404	0.8480
RS	0.0121	1.0000	0.0028	1.0000
RT	0.1341	1.0000	0.1332	1.0000
SHR	0.1335	0.9984	0.1369	0.9984
TR	0.1974	0.9688	0.2081	0.9507
RT + JC	0.1456	1.0000	0.1395	1.0000
RS + BA	0.2122	0.9542	0.2415	0.9428
TR + GN	0.2612	0.6799	0.3012	0.7758
FL + SH	0.0286	1.0000	0.0264	1.0000
GF + PT + CA	0.1971	0.9872	0.1941	0.9887
ST + RT + SHR	0.1802	0.9953	0.1750	0.9976
SH + TR + SN	0.1438	0.9953	0.1522	0.9992
RS + SPN + AF	0.1618	0.9203	0.0624	1.0000
CIE	0.0485	1.0000	0.0472	1.0000
CIC	0.0549	1.0000	0.0503	1.0000
RA	0.1360	1.0000	0.1405	1.0000
Average	0.1321	0.9693	0.1155	0.9794

Table 8: Effect of different backbone network on CMAA for CASIA dataset. The best results in each column are highlighted in **bold**.

	ViT-small	ResNet-18	VGG
NHD_d	0.4947	0.4879	0.4837
NHD_r	0.9870	0.1133	0.1182
F_1	0.9946	0.9836	0.9830

forming attack types not used during training as well as attacks used during training but with different parameters. Based on the attacks detailed in Table 6, the obtained results of generalization evaluation are given in Table 7.

As shown in Table 7, the results demonstrate outstanding generalization ability of our scheme with the average F_1 score of 0.9693 and 0.9794 for CASIA and Copydays dataset. Especially when faced with attacks that have not been encountered during training, CMAA still exhibits remarkable performance. At the same time, the average NHD_r value of 0.1321 for CASIA dataset and 0.1155 for Copydays dataset indicate that our approach is still effective.

Time complexity

The time complexity is evaluated by calculating the average time of hash generation based on the CASIA database. The results show that the computational time of our image hashing is 0.0315 seconds and the computational time of ProtoHash, TMICH, Swin-T, GLIF and ISM is 0.0345, 0.0401,

0.0291, 0.1716 and 0.1172 seconds, respectively. It can be seen that our CMAA is faster than ProtoHash, TMICH, GLIF and ISM, but it is little slower than Swin-T. These demonstrate the competitive time performance of our approach, indicating promising practical application prospects.

Theoretical discussion

In our paper, the weak strong augmentation alignment (WSA) is designed to further enhance robustness against the strong attacks inspired by (Wang and Qi 2022). (Wang and Qi 2022) has fully demonstrated the phenomenon that directly using stronger attacks in contrastive learning cannot improve its performance in classification tasks. For our task, the weak augmented view allows the model to learn the discrimination between different images and the robustness against weak attacks more easily. Specially, by using the KL divergence, we supervise the features of the strong augmented view with those of the weak augmented view. This mutual supervision allows both features to improve each other, meaning that the strong augmented view can learn discrimination and robustness from the weak augmented view, while the strong augmented view enhances the model's robustness against more complex attacks.

References

- Bottou, L. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, 177–186. Springer.
- Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.
- He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 16000–16009.
- He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9729–9738.
- Loshchilov, I.; and Hutter, F. 2016. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.
- Wang, X.; and Qi, G.-J. 2022. Contrastive learning with stronger augmentations. *IEEE transactions on pattern analysis and machine intelligence*, 45(5): 5549–5560.