Group-wise Scaling and Orthogonal Decomposition for Domain-Invariant Feature Extraction in Face Anti-Spoofing

Supplementary Material

A. Additional Visual Analysis

In this section, we describe additional visual analyses that were not covered in detail in Section 4.6 of the main paper. Specifically, we provide experimental validation of our method using PCA-based visualizations.

A.1. Feature Orthogonal Decomposition

For PCA analysis and visualization, we calculate the 2D principal components based on the source dataset (training data) which indicates the (seen) domains. The target dataset (test data) which indicates the (unseen) domains is then projected onto these 2D principal components. Finally, we visualize the features of the source and target datasets projected onto the derived principal components.

As shown in Fig. A-(a), (b), (c), and (d), all cases demonstrate strong learning consistency. However, there are some mismatches between the domain-separable feature space and the decision boundary, indicating that they are not parallel to each other. Specifically, Fig. A-(b) exhibits little skewed decision boundary in the Idaip domain, (c) in the CASIA domain, and (d) in the MSU domain. Referring to Tab 1 in the main paper, the alignment quality correlates with the HTER scores. The most aligned case, (a), achieves the lowest HTER of 0.42, followed by (b) with 0.93, (d) with 2.64, and finally, (c), which has the highest HTER of 3.33.

Moreover, Fig. A-(e), (f), (g), and (h) further confirm that better alignment quality enhances the alignment of the domain-specific separable space in the target dataset.

This analysis validates that aligning the domain-specific feature space with the decision boundary significantly contributes to improving domain generality.

A.2. Decomposition into Invariant and Specific Features

For PCA analysis and visualization, we conduct the 2D principal components analysis for invariant features which are projected onto the invariant basis features. The invariant basis features are driven by Gram-Schmidt process [1]. Specific features are then obtained by subtracting the invariant features from the original features. Finally, we visualize the invariant features and specific features separately to analyze their behavior.

The domain-invariant features should demonstrate that they are challenging to distinguish across different domains. In Fig. B-(a), (b), (c), and (d), this is evident as the liveness features are difficult to associate with specific domains. In contrast, spoofness features show some level of domain separability. This might be from the fact that liveness typically represents a single characteristic, while spoofness encompasses a variety of attack types, such as print attacks, video attacks, and partial attacks. However, they still prove their invariance to domains.

Specific features, on the other hand, should neither clearly distinguish spoofness from liveness nor differentiate between domains, as they are unrelated to domain invariance. Moreover, since they are projected onto the principal components of domain-invariant features, domain separability should also be minimal. As expected, Fig. B-(e), (f), (g), and (h) depict indistinct patterns, where neither spoofness nor domain information can be identified.

The overall visualizations confirm that our method effectively and explicitly decomposes domain-invariant components from domain-specific components, achieving the intended decomposition.

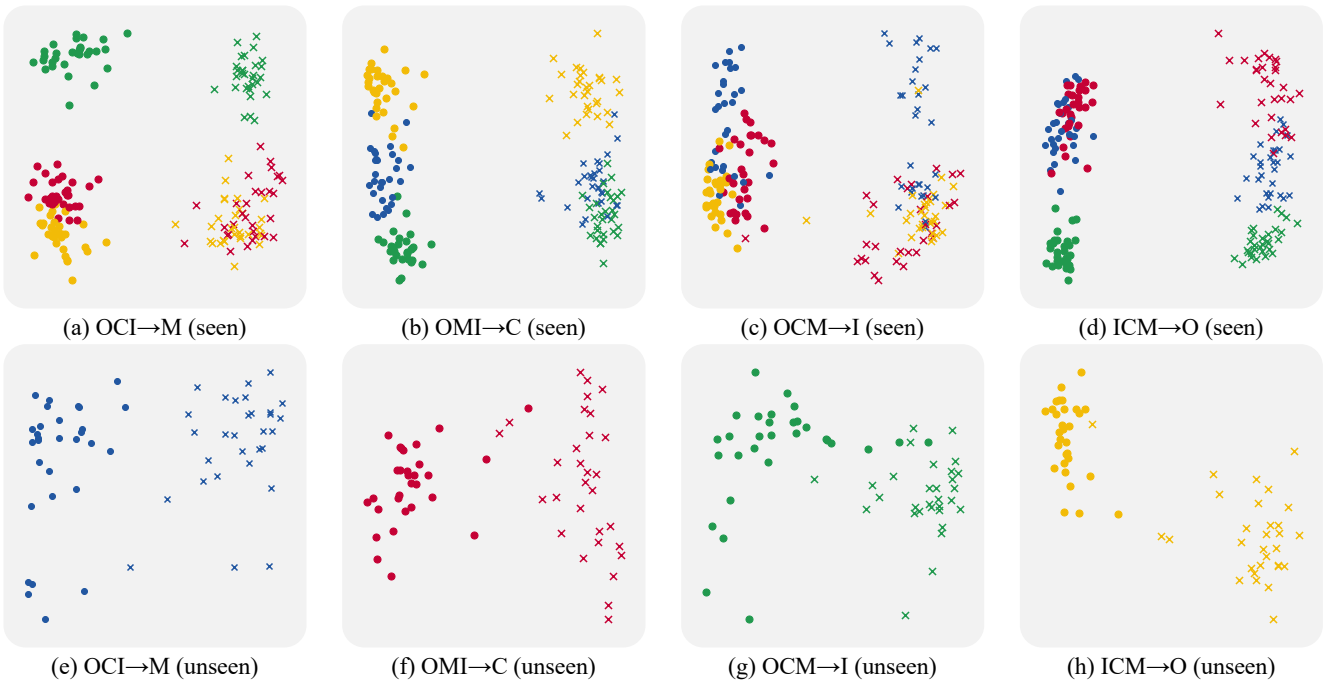
A.3. 3D visualization

We aim to intuitively examine the domain-invariant and domain-specific components. To achieve this, we further conduct 3D principal components analysis and project the features onto them for visualization, as shown in Fig. C. In the figure, the x-axis refers to the horizontal axis when the sphere is viewed from the front. Accordingly, at 0º on the x-axis, the domain-invariant components are highlighted; at 90º, the domain-specific components are emphasized; and at 40º, both components are visible. Additionally, the seen visualization represents data from the training set, while the unseen visualization includes both target and training datasets.

As proposed in Sec. A.1, Fig. C (seen, 0º) demonstrates that learning consistency is well-aligned. Furthermore, as discussed in Sec A.2, Fig. C (seen, 90º) shows that the domain-specific components can distinguish domains but cannot differentiate between liveness and spoofness. These results confirm that the domain-invariant and domain-specific components are effectively and explicitly separated. We also provide 3D visualized GIF images, including ours, FLIP, and SAFAS.

B. Model Reliability

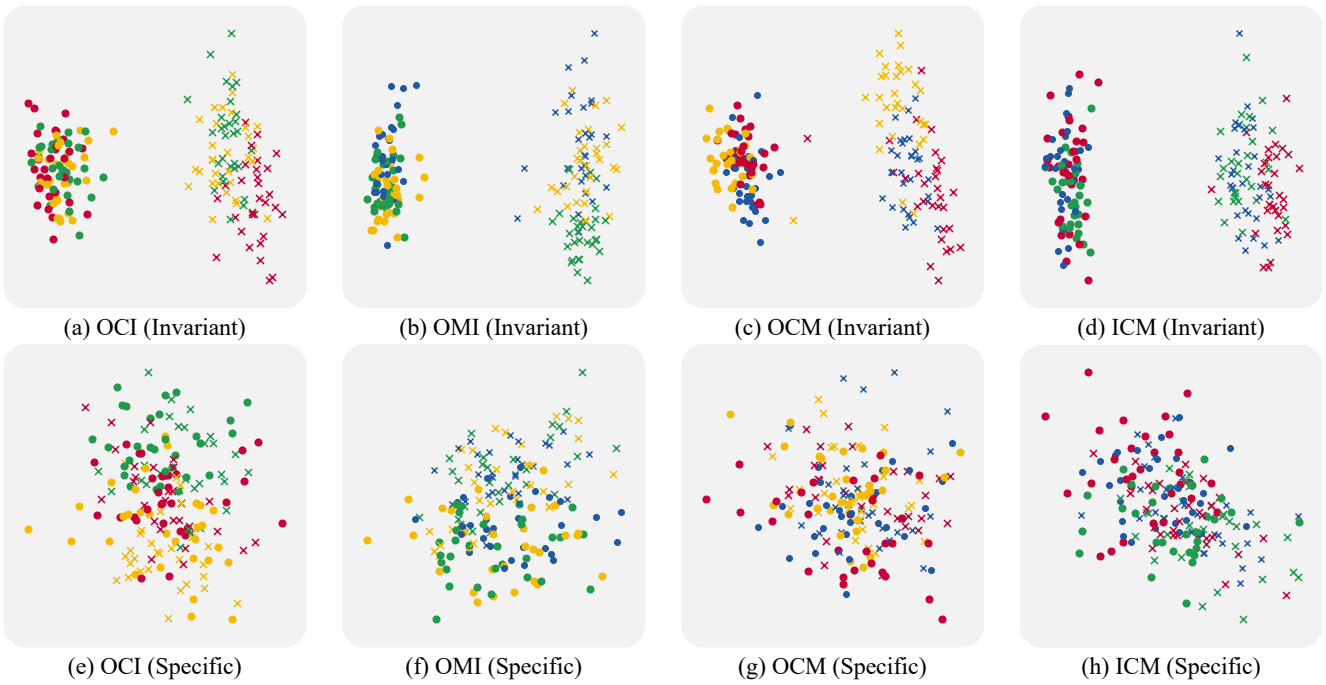
Regardless of how advanced AI models become, the implementation of fundamental safety mechanisms remains essential for their deployment in real-world scenarios, particularly within security systems. This necessity arises be-



(a) OCI→M (seen) (b) OMI→C (seen) (c) OCM→I (seen) (d) ICM→O (seen)

(e) OCI→M (unseen) (f) OMI→C (unseen) (g) OCM→I (unseen) (h) ICM→O (unseen)

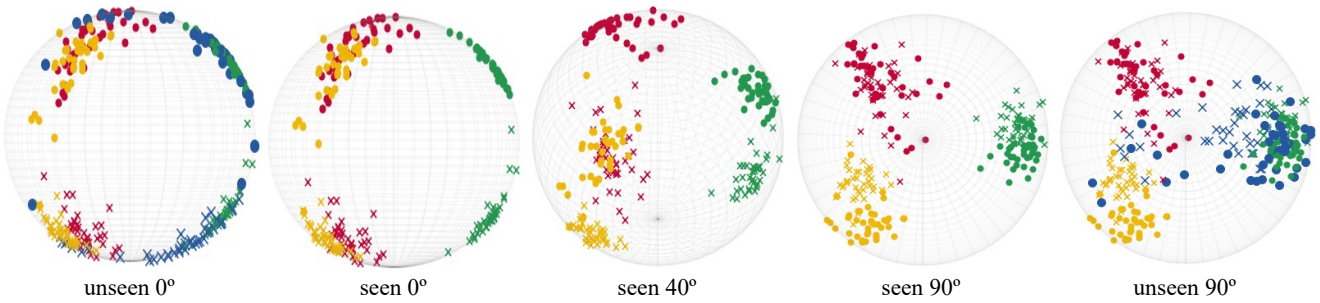
Figure A. **Visual analysis**: 2D PCA Visualization across domains. The term (seen) refers to the training dataset, while (unseen) refers to the target dataset that was not encountered during training. ○ and × represent liveness and spoofness, respectively. The datasets are represented as follows: ○,×: CASIA, ○,×: Idaip Replay-Attack, ○,×: MSU-MFSD, ○,×: OULU-NPU.



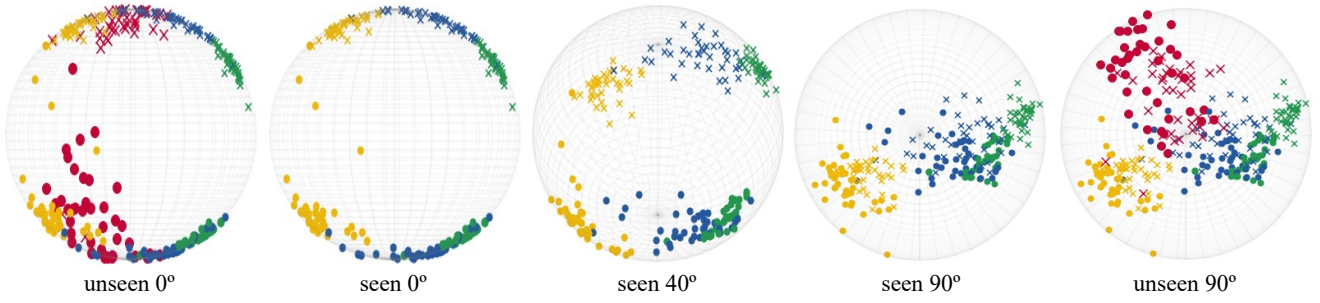
(a) OCI (Invariant) (b) OMI (Invariant) (c) OCM (Invariant) (d) ICM (Invariant)

(e) OCI (Specific) (f) OMI (Specific) (g) OCM (Specific) (h) ICM (Specific)

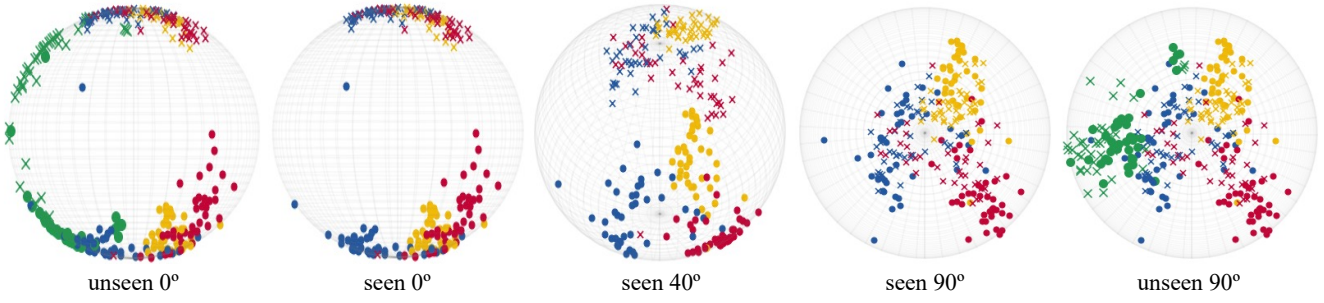
Figure B. **Decomposition into invariance and specificity for domain**: 2D PCA Visualization across domains. The term (Invariant) refers to features projected onto the invariant space, while (specific) refers to features projected onto the domain-specific separable space. ○ and × represent liveness and spoofness, respectively. The datasets are represented as follows: ○,×: CASIA, ○,×: Idaip Replay-Attack, ○,×: MSU-MFSD, ○,×: OULU-NPU.



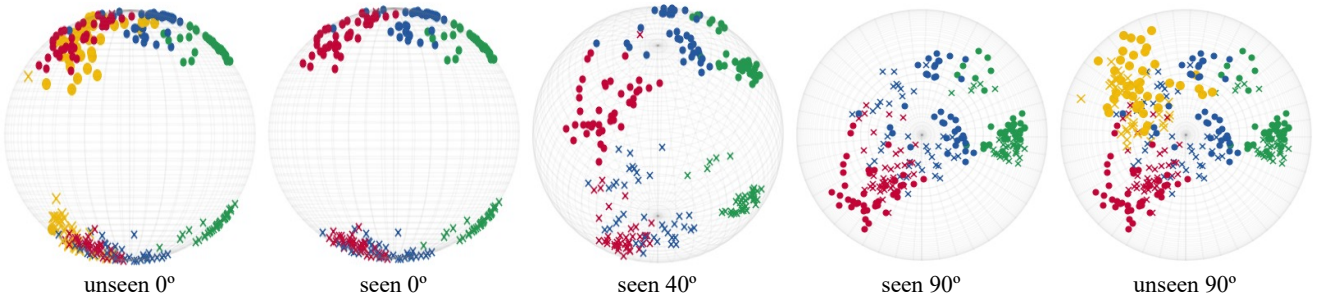
(a) OCI→M



(b) OMI→C



(c) OCM→I



(d) ICM→O

Figure C. **3D visualization projected on sphere**: PCA 3D Visualization across domains. The terms (0º), (40º), and (90º) represent the angles of rotation about the vertical axis. ○ and × represent liveness and spoofness, respectively. The datasets are represented as follows: ○,×: CASIA, ○,×: Idaip Replay-Attack, ○,×: MSU-MFSD, ○,×: OULU-NPU.

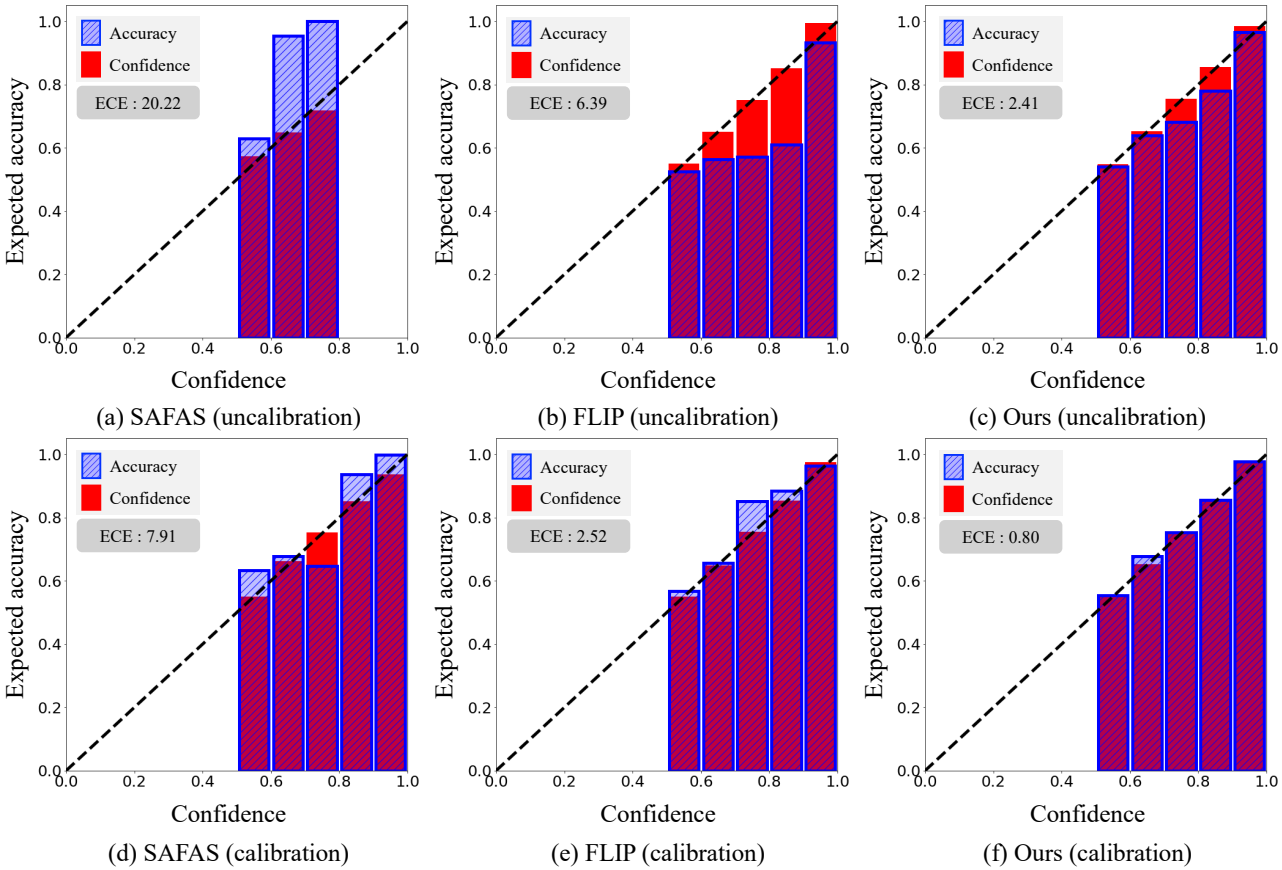


Figure D. **Reliable Diagram**: Comparison with Previous Methods. The terms (uncalibration) and (calibration) are distinguished based on whether the uncertainty has been calibrated.

Methods	C→I	C→M	C→O	I→C	I→M	I→O	M→C	M→I	M→O	O→C	O→I	O→M
FLIP-MCL† [10]	10.57	7.15	3.91	0.68	7.22	4.22	0.19	5.88	3.95	0.19	5.69	8.40
BUDoPT† [9]	4.33	2.62	4.98	0.48	1.83	4.14	**0.00**	2.45	1.87	0.44	2.53	1.43
TF-FAS† [11]	3.06	**1.59**	3.78	0.69	**1.34**	**2.50**	0.11	2.31	**1.40**	1.5	4.02	1.59
GD-FAS†	**2.98**	3.92	**3.36**	**0.19**	2.50	3.36	**0.00**	**2.23**	2.59	**0.00**	**1.83**	**0.42**

Table A. **Leave-three-domains-out protocol: † indicates the use of an extra source dataset (CelebA-Spoof [13]]).**

cause model reliability is often expressed through the probability values associated with predictions; however, recent AI models have demonstrated a tendency toward overconfidence [5]. Unfortunately, this critical aspect has been largely overlooked in prior studies.

In this section, we aim to underscore the importance of model reliability and delineate how our approach differs from previous methods by discussing the Expected Calibration Error (ECE) and the reliability diagrams introduced in Sec 4.1 of this paper.

A reliability diagram is an analytical tool that visually displays the discrepancy between predicted probability values and the actual probabilities. We conduct a visual analysis of the dataset both with and without uncertainty calibration, utilizing a post-hoc uncertainty calibration method that employs the temperature scaling technique. For this purpose, we split the test dataset into a validation set and a test set with a 2:8 ratio. The validation set is used to calibrate the reliability, while the test set is used to evaluate.

As shown in Fig. D, our method maintains a slightly overconfident state, whereas FLIP is significantly overconfident and, conversely, SAFAS is underconfident. In the calibrated diagrams below, the slightly under and overconfident regions are notably reduced. Here, we observe that our method before calibration outperforms other methods even after calibration.

Methods		SSAN-R [12]	SSDG-R [7]	DGUA-FAS [6]	BUDoPT [9]	GD-FAS
MI→C	HTER	25.56	19.86	19.22	5.33	**2.22**
	AUC	83.89	86.46	86.81	98.92	**99.15**
MI→O	HTER	24.44	27.92	20.05	5.94	**3.75**
	AUC	82.86	78.72	88.75	98.37	**98.75**

Table B. **Leave-two-domains-out protocol**

Methods	CS→W		SW→C		CW→S		Average	
	HTER	AUC	HTER	AUC	HTER	AUC	HTER	AUC
FLIP-MCL† [10]	4.46	**99.16**	9.66	96.69	11.71	95.21	8.61	97.02
CFPL† [8]	**4.40**	99.11	8.13	96.70	8.50	97.00	7.01	97.60
GD-FAS†	7.88	97.60	**4.01**	**98.59**	**5.87**	**98.04**	**5.92**	**98.08**

Table C. **Leave-one-domain-out protocol with different datasets**: † indicates the use of an extra source dataset (CelebA Spoof)

C. Additional Quantitative Experiments

We conducted experiments under more challenging settings, including the leave-two-domain-out (Sec. C.1) and leave-three-domain-out (Sec. C.2) protocols. Additionally, we evaluated our method on a different DGFAS dataset to assess its generalizability beyond the original benchmark(Sec. C.3). We evaluate all experiments using two standard metrics: Half Total Error Rate (HTER) and Area Under the ROC Curve (AUC).

C.1. Leave-three-domains-out protocol

Tab. A presents the results under the leave-three-domains-out protocol, where the model is trained on a single domain and evaluated on the remaining unseen domains. Our method, GD-FAS, consistently outperforms existing state-of-the-art approaches across most domain shifts. Notably, it achieves the best performance in particularly challenging scenarios such as O→C and M→C(HTER: 0.00%) and I→C (HTER: 0.19%). These results highlight the strong generalizability of our approach across highly diverse and difficult domain pairs.

C.2. Leave-two-domains-out protocol

Tab. B presents the performance under the leave-two-domains-out protocol, where the model is trained on two domains and evaluated on the remaining unseen domains—consistent with the experimental setup used in BUDoPT [9]. In both MI→C scenarios, GD-FAS achieves the lowest HTER (2.22% and 3.75%) and the highest AUC (99.15% and 98.75%), significantly outperforming prior methods such as BUDoPT [9], DGUA-FAS [6], and SSDG-R [7]. These results demonstrate the effectiveness of our group-wise scaling and orthogonal decomposition techniques in capturing domain-invariant features, even under limited training conditions and complex inter-domain shifts.

C.3. Different dataset

Our experiments were conducted on three diverse datasets—Surf (S) [4], CeFA (C) [3], and WMCA (W) [2]—under the leave-one-domain-out protocol to evaluate cross-dataset generalization.

As shown in Tab. C, while FLIP-MCL and CFPL achieve high AUC scores, GD-FAS demonstrates the most balanced performance, achieving the best average results across all domains (HTER: 5.92%, AUC: 98.08%). Notably, in the SW→C setting, GD-FAS outperforms all baselines with a significantly lower HTER (4.01%) and higher AUC (98.59%), highlighting its robustness against challenging cross-domain shifts.

D. Discussion of Computational Cost

In this section, we discuss the computational cost of GD-FAS.

Training Efficiency The additional computational overhead introduced by the GS-RM and FOD losses is approximately 10M FLOPs, which is negligible compared to the overall training complexity of FLIP-MCL (88.6G FLOPs). GD-FAS also adopts the same backbone architecture as FLIP-MCL and simply replaces the original MSE loss with the proposed GS-RM and FOD losses. As a result, GD-FAS maintains a comparable training cost to FLIP-MCL. In contrast, prior methods often rely on significantly more resource-intensive components, such as large language models (LLMs) or two-stage training pipelines.

Inference Efficiency GD-FAS achieves an inference time of 0.01 seconds per frame, matching the computational cost of FLIP-MCL due to their shared inference pipeline. In comparison, other baseline methods typically incur slightly higher computational overhead and latency.

Prediction	OCI→M HTER / AUC / ECE	OMI→C HTER / AUC / ECE	OCM→I HTER / AUC / ECE	ICM→O HTER / AUC / ECE
$\mathbf{w_I} \otimes \mathbf{w_T}$	**0.42 / 99.88 / 3.39**	**0.93 / 99.99 / 1.82**	**3.33 / 98.61 / 2.37**	**2.64 / 99.34 / 2.86**
Φ_C	2.50 / 99.44 / 5.02	2.22 / 99.71 / 5.15	4.83 / 98.75 / 3.51	3.33 / 99.28 / 2.41

Table D. **Embedding Features ($\mathbf{w_I} \otimes \mathbf{w_T}$) vs. Classifier ($\phi_C$)**

batch size	OCI→M HTER / AUC / ECE	OMI→C HTER / AUC / ECE	OCM→I HTER / AUC / ECE	ICM→O HTER / AUC / ECE
3	5.00 / 98.77 / 6.38	3.15 / 99.24 / 15.81	6.67 / 98.72 / 2.24	2.96 / 99.24 / 9.41
8	2.50 / 99.85 / 3.60	2.22 / 98.84 / 8.67	5.00 / 98.94 / 2.78	3.01 / 99.20 / 5.41
16	**0.42** / 99.88 / 3.39	**0.93** / **99.99** / 1.82	**3.33** / 98.61 / 2.37	**2.64 / 99.34 / 2.86**
24	**0.42** / **99.90** / 2.79	**0.93** / 99.86 / 0.80	**3.33** / **99.25** / 2.79	2.96 / **99.34** / 5.77
32	**0.42** / 99.73 / **2.53**	1.11 / 99.76 / **0.72**	3.50 / 99.24 / **1.84**	3.01 / 99.07 / 6.72

Table E. **Influence of Batch-size**

Method	Target	OCI→M	OMI→C	OCM→I	ICM→O
Gaussian Noise	Image	7.5	3.52	3.5	4.77
Random Vector	Text	7.08	4.44	11.83	7.36
a Pohoto of {dog or cat}	Text	3.3	12.04	5.60	15.09
a Photo of {live of fake} face	Text	2.5	1.11	5.17	4.07
Baseline (FLIP-MCL)	None	**0.42**	**0.93**	**3.33**	**2.64**

Table F. **Robustness across Image and Text Quality**

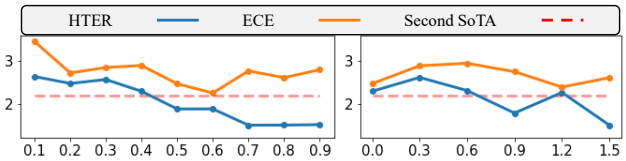
E. Classifier vs Embedding Features

In this section, we explain why GD-FAS detects spoofing attacks using embedding features rather than relying on the classifier output, as illustrated in Fig. **??**. As shown in Tab. D, predictions based on embedding features outperform those based on the classifier. This performance gap arises because the embedding space explicitly separates domain-invariant and domain-specific information through orthogonal decomposition. In contrast, the classifier operates on features prior to this decomposition, failing to fully exploit the decomposed representations.

F. Analysis of Batch-Size Influence

We analyze the influence of batch size on model performance. CLIP-based methods typically use a batch size of 3 per domain due to the large model size, resulting in an effective batch size equal to the per-domain batch size multiplied by the number of domains.

As shown in Tab. E, a batch size of 16 is sufficient for stable and effective training. Increasing the batch size beyond 16 yields marginal gains while requiring GPU memory in excess of 48GB, which may not be practical for most setups.



(a) Hyperparameter λ_1 (b) Hyperparameter β

Figure E. **Hyperparameter Sensitivity**: Default λ_1=0.8, β=1.5

G. Robustness across Image and Text Quality

We analyze the robustness of GD-FAS to variations in both image and text quality. GD-FAS utilizes the same text templates as FLIP-MCL, comprising six distinct templates for each class—spoof and live—to guide representation learning. To evaluate robustness, we conduct experiments under four distinct conditions: one related to image quality and three to text quality. For the image case, we apply Gaussian noise to the input images to simulate visual degradation. For the text cases: (1) we replace text embeddings with random vectors, (2) we use only a single incorrect-class template, and (3) we use only a single correct-class template.

As descriebed in Tab. F, while both image and text degradations lead to moderate performance drops, degradation in text quality—particularly the use of random vectors or incorrect-class templates—results in a more substantial de-

cline. These findings suggest that text embeddings play a crucial role in enabling GD-FAS to extract domain-invariant features effectively.

H. Sensitivity of Hyperparameter

We performed a series of analytical experiments to examine how FOD and GS-RM contribute to overall performance, as shown in Fig. E. The hyperparameters λ_1 and β correspond to FOD and GS-RM, respectively. While our method exhibits slight sensitivity to hyperparameter variations, the combined use of FOD and GS-RM consistently improves performance as their contributions are strengthened. We selected the final hyperparameter values based on HTER, as it directly reflects the model's domain generalization ability.

References

[1] Ward Cheney and David Kincaid. Linear algebra: Theory and applications. *The Australian Mathematical Society*, 110: 544–550, 2009. 1

[2] George et al. Biometric face presentation attack detection with multi-channel convolutional neural network. *TIFS*, 2019. 5

[3] Liu et al. Casia-surf cefa: A benchmark for multi-modal cross-ethnicity face anti-spoofing. In *WACV*, 2021. 5

[4] Zhang et al. Casia-surf: A large-scale multi-modal benchmark for face anti-spoofing. *TBBIS*, 2020. 5

[5] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017. 4

[6] Zong-Wei Hong, Yu-Chen Lin, Hsuan-Tung Liu, Yi-Ren Yeh, and Chu-Song Chen. Domain-generalized face anti-spoofing with unknown attacks. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 820–824. IEEE, 2023. 5

[7] Yunpei Jia, Jie Zhang, Shiguang Shan, and Xilin Chen. Single-side domain generalization for face anti-spoofing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8484–8493, 2020. 5

[8] Ajian Liu, Shuai Xue, Jianwen Gan, Jun Wan, Yanyan Liang, Jiankang Deng, Sergio Escalera, and Zhen Lei. Cfpl-fas: Class free prompt learning for generalizable face anti-spoofing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 222–232, 2024. 5

[9] Si-Qi Liu, Qirui Wang, and Pong C Yuen. Bottom-up domain prompt tuning for generalized face anti-spoofing. In *European Conference on Computer Vision*. Springer, 2025. 4, 5

[10] Koushik Srivatsan, Muzammal Naseer, and Karthik Nandakumar. Flip: Cross-domain face anti-spoofing with language guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19685–19696, 2023. 4, 5

[11] Xudong Wang, Ke-Yue Zhang, Taiping Yao, Qianyu Zhou, Shouhong Ding, Pingyang Dai, and Rongrong Ji. Tf-fas: twofold-element fine-grained semantic guidance for generalizable face anti-spoofing. In *European Conference on Computer Vision*, pages 148–168. Springer, 2025. 4

[12] Zhuo Wang, Zezheng Wang, Zitong Yu, Weihong Deng, Jiahong Li, Tingting Gao, and Zhongyuan Wang. Domain generalization via shuffled style assembly for face anti-spoofing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4123–4133, 2022. 5

[13] Di Wen, Anil. K Jain, and Hu Han. Face Spoof Detection with Image Distortion Analysis. *IEEE Trans. Information Forensic and Security*, 2015 (To Appear). 4