DGS-Net: Distillation-Guided Gradient Surgery for CLIP Fine-Tuning in AI-Generated Image Detection

Yan et al., ICML 2026 Spotlight

Presenter: CHANWOO KIM

July 16, 2026

Seoul National University

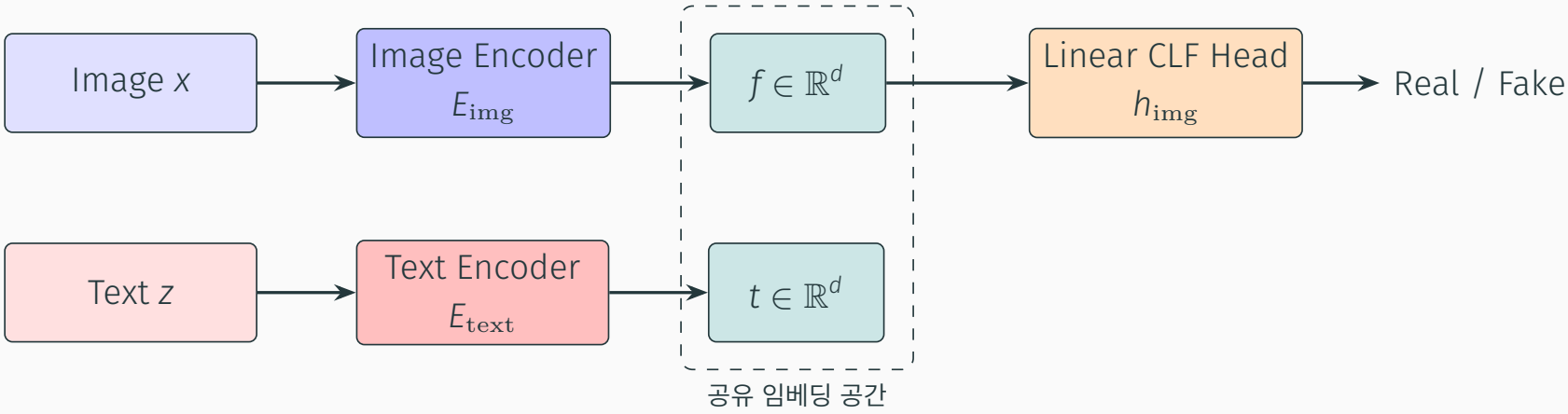
Research Summary

한줄 요약

CLIP fine-tuning 과정에서 발생하는 두 가지 문제를 해결하기 위해, 학습 단계에서 gradient를 조작하는 방법을 제안.

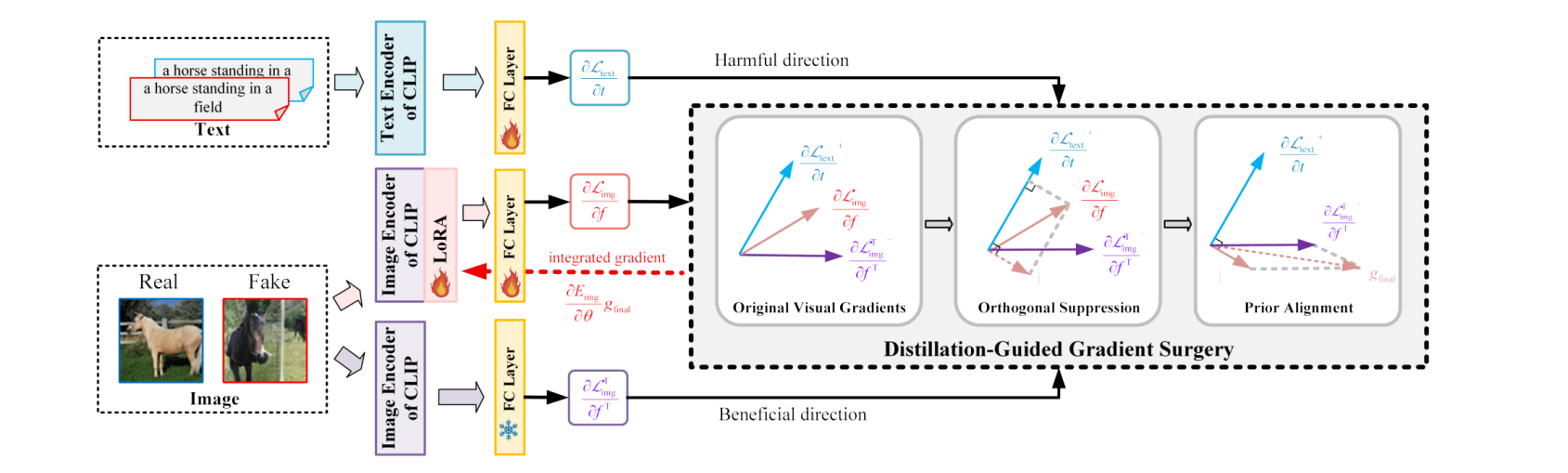
- **문제 1:** Fine-tuning으로 인해 pre-trained representation이 손실되는 **catastrophic forgetting**.
- **문제 2:** Detection과 무관한 **semantic representation**이 일반화를 저해하는 문제.
- **해결 방법:** Teacher model에서 얻은 gradient 정보를 이용하여 student model의 gradient를 조작하는 **gradient-level distillation**.

Preliminaries: CLIP Encoder



- CLIP은 **두 개의 encoder**(image / text)로 구성되며, 이미지–텍스트 pair **데이터**와 **contrastive loss**로 같은 의미의 텍스트와 이미지가 **공유 임베딩 공간**에서 유사하게 위치하도록 학습됨.
- Detector로 사용하기 위해서는 image encoder에 **linear classification head**를 붙이고 **fine-tuning** (real / fake 이진 분류).

Method: DGS-Net Overview



- Image Encoder + clf layer: 학습 대상
- Text Encoder + clf layer: harmful direction 제공
- Image Encoder + clf layer (Frozen): beneficial direction 제공 (파인튜닝 전 CLIP 복사본)
- 두 방향으로 이미지 gradient를 surgery: **Orthogonal Suppression** + **Prior Alignment**.

Method: Notation and Setup

- x: 입력 이미지
- $y \in \{0, 1\}$: 레이블 (real $= 0$, fake $= 1$)
- z: BLIP으로 생성한 x의 텍스트 설명
- $f = E_{\text{img}}(x; \theta) \in \mathbb{R}^d$: 학습 중인 이미지 인코더의 feature
- $t = E_{\text{text}}(z; \phi) \in \mathbb{R}^d$: 텍스트 인코더의 feature ($\phi$: frozen)
- $f^T = E^T_{\text{img}}(x) \in \mathbb{R}^d$: frozen 이미지 인코더의 feature
- $h_{\text{img}},\ h_{\text{text}},\ h^T_{\text{img}}$: 각 인코더의 선형 분류 head
- $\mathcal{L}_{\text{img}} = \ell(h_{\text{img}}(f), y)$: 학습 중인 이미지 인코더 Loss (BCEWithLogits)
- $\mathcal{L}_{\text{text}} = \ell(h_{\text{text}}(t), y)$: 텍스트 인코더 Loss
- $\mathcal{L}^T_{\text{img}} = \ell(h^T_{\text{img}}(f^T), y)$: frozen 이미지 인코더 Loss
- $g_{\text{task}} = \nabla_f \mathcal{L}_{\text{img}} \in \mathbb{R}^d$: 이미지 feature f에 대한 task gradient
- $g_{\text{text}} = \nabla_t \mathcal{L}_{\text{text}} \in \mathbb{R}^d$: 텍스트 feature t에 대한 gradient
- $g_{\text{img}} = \nabla_{f^T} \mathcal{L}^T_{\text{img}} \in \mathbb{R}^d$: frozen feature f^T에 대한 gradient

Method: Orthogonal Suppression

목적

Semantic 정보를 **전부 버리는 것이 아니라**, semantic 정보 중 detection task에 **도움이 되지 않는 성분(harmful direction)만 제거**.

- 이미지의 semantic 정보 추출: **BLIP**으로 이미지에 대응되는 캡션 z를 생성하고, text encoder에 넣어 text feature t를 추출.
- 텍스트 인코더 clf에서 feature gradient g_{text}를 계산한 뒤, **양수 원소는 그대로, 음수 원소는 0**으로 매핑하여 g_{harm} 구성:

$$g_{\text{harm}} \triangleq [g_{\text{text}}]_+ \in \mathbb{R}^d, \qquad ([a]_+)_j = \max(a_j, 0)$$

- Task gradient를 g_{harm}의 **orthogonal space로 projection**:

$$\tilde{g} = \left(I - \hat{g}_{\text{harm}}\, \hat{g}_{\text{harm}}^{\top}\right) g_{\text{task}}, \qquad \hat{g}_{\text{harm}} = \frac{g_{\text{harm}}}{\|g_{\text{harm}}\|_2}$$

$\Rightarrow$ 업데이트 방향에서 harmful semantic 성분이 제거되어, task와 무관한 semantic 학습을 차단.

Method: Prior Alignment

목적

Fine-tuning **이전의 frozen 모델(teacher)의 학습 방향을 따라가도록** 하여, feature 구조를 보존하고 catastrophic forgetting 방지.

- Frozen image encoder branch에서 feature gradient g_{img}를 계산한 뒤, **음수 원소는 그대로, 양수 원소는 0**으로 매핑하여 g_{help} 구성:

$$g_{\text{help}} \triangleq [g_{\text{img}}]_{-} \in \mathbb{R}^d, \qquad ([a]_{-})_j = \min(a_j, 0)$$

- 앞의 projection **과정을 거친 벡터** $\tilde{g}$에 g_{help}를 λ **scale로 더함**:

$$g_{\text{final}} = \tilde{g} + \lambda\, g_{\text{help}}$$

Method: Final Gradient Update

- **전체 학습 목적 함수**:

$$\mathcal{L} = \mathcal{L}_{\text{img}} + \mathcal{L}_{\text{text}} + \lambda \mathcal{L}_{\text{align}}$$

- 역전파 시 $\frac{\partial \mathcal{L}_{\text{img}}}{\partial f}$를 다음으로 **교체**:

$$g_{\text{final}} = \underbrace{\left(I - \hat{g}_{\text{harm}} \hat{g}_{\text{harm}}^{\top}\right) \nabla_f \mathcal{L}_{\text{img}}}_{\text{Orthogonal Suppression } (\tilde{g})} + \underbrace{\lambda\, g_{\text{help}}}_{\text{Prior Alignment}}$$

- **파라미터 업데이트** (chain rule, 학습률 μ):

$$\theta \leftarrow \theta - \mu \left(\frac{\partial E_{\text{img}}(x; \theta)}{\partial \theta} \right)^{\top} g_{\text{final}}$$

Experiments: t-SNE Visualization (Figure 1)

- 강한 real/fake 분리와 feature 구조를 **동시에** 유지.

Experiments: AIGCDetectBench (Table 1, 2)

Table 1. Cross-model Accuracy (Acc.)

Detection Method	Real Image	Generative Adversarial Networks							Other		Diffusion Models								mAcc.
		Pro-GAN	Cycle-GAN	Big-GAN	Style-GAN	Style-GAN2	Gau-GAN	Star-GAN	WFIR	Deep-fake	SDv1.4	SDv1.5	ADM	GLIDE	Mid-journey	Wukong	VQDM	DALLE2	
CNN-Spot (Wang et al., 2020)	99.0	95.3	18.7	1.8	36.5	22.0	2.5	23.1	1.1	29.3	55.9	55.6	1.8	4.8	5.2	27.6	0.7	4.5	29.0
UnivFD (Ojha et al., 2023)	92.3	98.9	90.5	79.2	55.7	48.7	91.1	96.9	92.8	26.9	96.3	96.0	12.7	75.6	61.2	84.7	45.6	62.3	72.7
FreqNet (Tan et al., 2024a)	89.9	99.4	57.1	51.0	75.1	67.5	9.9	88.4	95.4	35.8	99.9	99.8	37.7	78.9	80.8	98.0	34.1	88.8	71.7
NPR (Tan et al., 2024b)	99.3	98.9	29.3	16.5	67.1	58.7	1.7	6.5	7.9	0.1	100.0	99.9	26.5	69.2	71.0	97.7	15.4	89.8	53.1
Ladeda (Cavia et al., 2024)	99.8	99.7	7.2	29.0	95.6	98.3	4.9	0.0	19.2	0.1	100.0	99.9	27.3	79.7	88.8	97.9	15.5	92.4	58.6
AIDE (Yan et al., 2024a)	93.0	95.3	88.0	96.9	89.6	97.0	90.0	97.0	42.9	9.2	99.8	99.7	92.4	98.7	63.7	98.9	90.1	98.9	85.6
C2P-CLIP* (Tan et al., 2025)	99.8	100.0	99.8	99.8	72.2	78.0	87.7	100.0	34.9	67.6	100.0	99.9	50.5	73.3	28.4	99.7	93.3	98.1	82.4
DFFreq (Yan et al., 2026)	98.0	98.0	70.9	93.6	99.0	98.9	93.8	97.4	3.5	76.0	99.9	99.8	65.9	87.8	**94.8**	99.1	76.0	95.9	85.2
SAFE (Li et al., 2025b)	99.2	99.7	85.6	83.5	88.1	96.7	89.0	99.9	8.4	17.3	99.8	99.6	36.8	90.5	86.3	98.5	84.0	92.0	80.8
Effort (Yan et al., 2024b)	**99.8**	100.0	100.0	100.0	79.6	82.1	100.0	100.0	**99.9**	76.6	100.0	99.9	53.1	81.2	44.2	99.9	96.3	75.8	88.1
NS-Net (Yan et al., 2025b)	97.7	100.0	99.9	100.0	95.7	98.6	98.5	100.0	68.9	60.6	100.0	99.9	85.3	97.2	77.4	100.0	98.9	98.8	93.2
Ours	95.3	**100.0**	**100.0**	**100.0**	**99.5**	**99.8**	**100.0**	**100.0**	84.6	**96.7**	**100.0**	**99.9**	**95.2**	**99.0**	88.0	**100.0**	**99.7**	**99.6**	**97.6**

Table 2. Cross-model Average Precision (A.P.)

Detection Method	Generative Adversarial Networks							Other		Diffusion Models								mA.P.
	Pro-GAN	Cycle-GAN	Big-GAN	Style-GAN	Style-GAN2	Gau-GAN	Star-GAN	WFIR	Deep-fake	SDv1.4	SDv1.5	ADM	GLIDE	Mid-journey	Wukong	VQDM	DALLE2	
CNN-Spot(Wang et al., 2020)	99.9	91.0	59.2	94.8	94.2	86.1	82.5	65.5	74.2	97.6	97.6	66.6	83.1	80.4	88.9	57.0	87.8	84.3
UnivFD (Ojha et al., 2023)	99.9	96.2	94.7	90.9	91.1	97.6	99.7	83.0	83.5	99.3	99.1	65.2	95.3	91.7	97.4	87.2	89.8	91.9
FreqNet (Tan et al., 2024a)	100.0	97.5	87.9	95.9	94.2	54.0	100.0	57.0	88.5	99.9	99.9	73.9	94.1	94.4	99.1	74.9	92.3	88.4
NPR (Tan et al., 2024b)	100.0	98.6	79.4	98.6	99.6	71.3	97.6	65.5	92.5	100.0	99.9	74.1	97.4	97.8	99.9	81.5	99.3	91.4
Ladeda (Cavia et al., 2024)	100.0	99.0	89.5	100.0	100.0	96.4	90.3	93.6	92.6	95.5	95.8	84.8	96.6	93.5	93.0	84.5	93.9	94.0
AIDE (Yan et al., 2024a)	99.6	98.2	91.6	99.4	99.9	74.1	99.7	90.8	66.5	99.9	99.9	99.4	99.9	90.3	99.9	98.9	99.9	94.5
C2P-CLIP* (Tan et al., 2025)	100.0	100.0	99.9	99.3	99.7	98.8	100.0	99.6	98.3	100.0	100.0	96.3	98.9	90.4	100.0	99.9	100.0	98.8
DFFreq (Yan et al., 2026)	100.0	98.7	98.4	100.0	100.0	98.1	99.8	89.0	87.5	100.0	99.9	98.7	99.5	99.7	100.0	99.3	99.9	98.1
SAFE (Li et al., 2025b)	100.0	99.5	97.8	99.9	100.0	96.6	100.0	73.8	91.0	100.0	100.0	98.1	99.9	**99.9**	100.0	99.9	100.0	97.4
Effort (Yan et al., 2024b)	100.0	100.0	**100.0**	99.3	99.2	**100.0**	100.0	91.1	97.9	100.0	100.0	97.5	99.8	98.9	100.0	100.0	99.9	99.0
NS-Net (Yan et al., 2025b)	100.0	100.0	99.7	99.8	99.9	97.0	100.0	99.5	95.2	100.0	100.0	**100.0**	99.8	97.0	100.0	100.0	100.0	99.2
Ours	**100.0**	**100.0**	99.9	**100.0**	**100.0**	98.8	**100.0**	**99.8**	**98.3**	**100.0**	**100.0**	99.8	**100.0**	99.6	**100.0**	**100.0**	**100.0**	**99.8**