TIACam: Text-Anchored Invariant Feature Learning with Auto-Augmentation for Camera-Robust Zero-Watermarking

Supplementary Material

S1. Training Data Diversity and Caption Examples

To illustrate the diversity of text-image supervision used during training, we provide qualitative examples from the Visual Genome and Flickr30k datasets in Fig. 1 and Fig. 2. Visual Genome contains short, object-centric region descriptions, often listing entities or attributes, while Flickr30k provides full natural sentences with richer structure and contextual detail. Despite this difference in caption granularity and linguistic style, TIACam trains stably on the combined dataset and produces consistent invariant features. This demonstrates that the model is not tied to a particular caption format and that its learned feature space generalizes across heterogeneous annotation styles, helping reduce dataset-specific bias and improving robustness in multimodal settings.

S2. Realism of the Learned Auto-Augmentor Distortions

To illustrate the realism and diversity of the distortions learned by our auto-augmentor, Fig. 3 presents representative examples. Each row shows the original input followed by distortions synthesized by the learned modules, including perspective deformation, photometric shifts, additive noise, filtering artifacts, JPEG degradation, and Moiré interference.

These visualizations show that the auto-augmentor produces distortions characteristic of real camera pipelines: such as chromatic imbalance, sensor noise, edge warping, and Moiré patterns, rather than simple digital corruptions. The close resemblance between these synthesized distortions and actual screen- and print-recapture effects supports our claim that the learned augmentations effectively model camera perturbations, enabling TIACam to learn invariance that transfers robustly to real-world camera scenarios.

S3. Additional Ablation Study: Learned Auto-Augmentor vs. Manual Distortions

To assess the impact of the learnable auto-augmentor, we replace it with a fixed set of hand-crafted distortions while keeping the feature extractor and training objectives unchanged. We then compare cosine similarity between invariant features from original and distorted images. As shown in Tables 1 and 2, the manual augmentor produces noticeably lower similarity across both synthetic distortions and real capture settings (screen camera, print camera, and screenshots). In contrast, the learned auto-augmentor consistently achieves much higher similarity, indicating stronger distortion-invariant feature alignment.

Table 1. Cosine similarity between original and distorted invariant features across six distortion types.

Distortion Type	Manual Augmentor (Fixed)	Auto-Augmentor (Ours)
Additive Noise	0.84	**0.98**
Photometric	0.86	**0.96**
Perspective	0.78	**0.92**
JPEG Compression	0.88	**0.97**
Moiré Pattern	0.76	**0.89**
Filtering	0.89	**0.98**

Table 2. Cosine similarity between original and distorted invariant features across Screen Camera, Print Camera, Screenshot

Distortion Type	Manual Augmentor (Fixed)	Auto-Augmentor (Ours)
Screen Camera	0.82	**0.91**
Print Camera	0.76	**0.87**
Screenshot	0.78	**0.85**

We further compare watermark extraction performance under real camera distortions. Table 3 reports bit accuracy for 30-, 100-, and 200-bit payloads under randomly selected six distortions out of Additive, Photometric, Perspective, JPEG, Moiré, and Filtering Noise. Consistent with the feature-level results, models trained with manual distortions degrade as the message length increases, while the learned auto-augmentor maintains high accuracy across all payload sizes.

Table 3. Bit accuracy (%) under randomly applied 6 distortions.

Model Variant	30-bit	100-bit	200-bit
Manual Augmentor (Fixed)	91.2	89.5	89.3
Auto-Augmentor (Ours)	**98.7**	**96.4**	**96.1**

Table 4 provides a breakdown by evaluating the same two models under three specific capture pipelines: screen camera, print camera, and screenshots. The manual augmentor exhibits notable drops, especially under print-camera and high-bit scenarios, while the auto-augmentor consistently achieves strong accuracy across all conditions.

Table 4. Bit accuracy (%) of Manual Augmentor and Auto Augmentor under screen camera, print camera, and screenshot distortions (30-bit and 100-bit).

Method	Screen Camera		Print Camera		Screenshots	
	30 bits	100 bits	30 bits	100 bits	30 bits	100 bits
Manual Augmentor	93.3%	92.6%	90.1%	87.4%	92.7%	87.9%
Auto Augmentor (Ours)	**99.1%**	**98.4%**	**96.9%**	**95.3%**	**97.6%**	**96.5%**

These results show that the learned auto-augmentor provides stronger and more realistic distortion modeling than fixed operators. By better approximating physical camera artifacts, the auto-augmentor enables TIACam to learn more

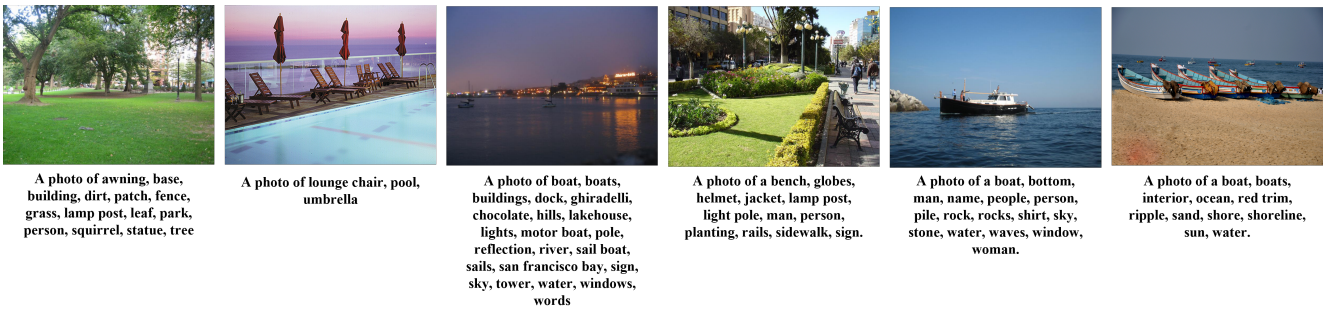

Figure 1. Example of Visual Gnome Training Data used in TIACam.

Figure 2. Example of Flickr training data used in TIACam. The first row shows the images, the second row contains the corresponding captions, and the third row presents paraphrased versions of those captions.

stable invariant features and substantially improves real-world watermark robustness.

S4. Comprehensive Comparison with Existing Digital and Zero-Watermarking Methods

The main paper focuses on camera-based evaluations, where TIACam shows clear advantages over learning-based watermarking baselines. For completeness, we further compare TIACam with a broader set of digital watermarking and zero-watermarking systems under digital appearance distortions, geometric transformations, and real camera capture. These supplementary results confirm that the robustness of TIACam extends well beyond the camera setting.

Table 5 reports bit accuracy under a range of common digital perturbations. While pretrained feature-based zero-watermarking approaches remain reliable under mild settings, they degrade substantially under geometric or severe appearance changes such as rotation, heavy cropping, blur, and screenshot artifacts. In contrast, TIACam maintains consistently high accuracy across all digital distortions, highlighting the benefit of learning distortion-invariant features rather than relying solely on pretrained representations.

We next compare watermarking systems under geometric transformations in Table 6. Non-camera-oriented digital watermarking approaches (DWSF, MuST, WOFA) experience substantial performance drops as geometric severity increases, particularly under large rotations and scale changes. TIACam maintains high accuracy across all geometric settings, demonstrating strong invariance to spatial misalignment and confirming the stability of the learned invariant feature space.

Table 5. Bit accuracy (%) under digital distortions.

Distortion	ZBW [27]	WSSL [7]	TIACam
Identity	1.00	1.00	**1.00**
Rotation (25°)	0.27	1.00	**1.00**
Crop (0.5)	1.00	**1.00**	0.98
Crop (0.1)	0.02	0.98	**1.00**
Resize (0.7)	1.00	1.00	**1.00**
Blur (2.0)	0.25	1.00	**1.00**
JPEG (50)	0.96	0.97	**0.99**
Brightness (2.0)	**0.99**	0.96	0.97
Contrast (2.0)	1.00	1.00	**1.00**
Hue (0.25)	1.00	1.00	**1.00**
Screenshot	0.86	0.97	**0.99**

Table 6. Bit accuracy (%) under geometric distortions.

Distortion	DWSF [11]	MuST [28]	WOFA [21]	TIACam
Translation 10%	52.97	50.00	91.97	**93.9**
Translation 25%	49.87	49.98	**93.25**	92.1
Translation 50%	49.92	49.74	87.93	**90.3**
Rotation 15°	50.21	49.79	95.26	**99.2**
Rotation 30°	49.74	49.73	94.24	**97.8**
Rotation 45°	49.60	49.82	90.63	**95.5**
Scaling ±10%	53.30	49.98	95.72	**97.4**
Scaling ±20%	51.78	49.99	95.50	**96.1**
Scaling ±25%	51.08	50.00	**95.02**	93.4

Finally, Table 7 evaluates insertion-based watermarking methods (DWSF, WOFA, MuST) under real camera distortions. Although these systems are effective under purely digital perturbations, they degrade sharply in real capture scenarios. TIACam consistently achieves higher extraction accuracy across bit lengths and distortion types, further

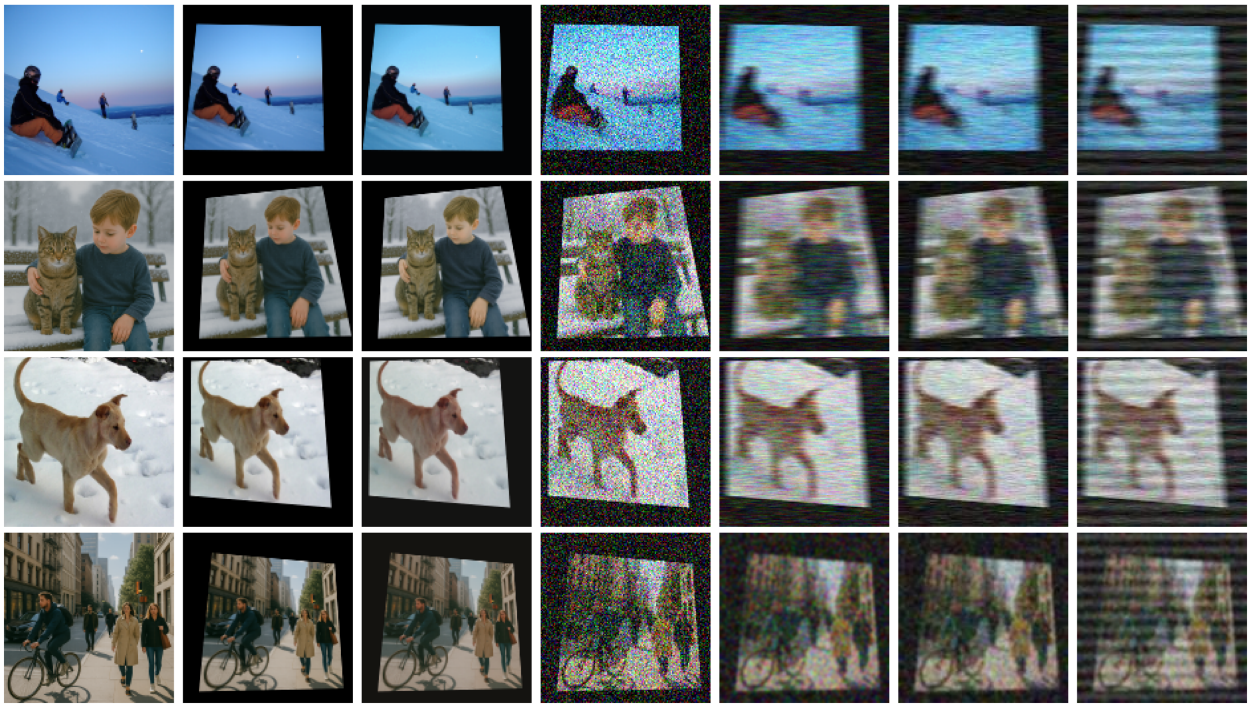

Figure 3. Examples of auto-augmented images.

demonstrating its strong generalization to physical-world artifacts.

Table 7. Bit accuracy (%) of DWSF, WOFA, MuST, and TIACam under screen camera, print camera, and screenshot distortions (30-bit and 100-bit).

Method	Screen Camera		Print Camera		Screenshots	
	30 bits	100 bits	30 bits	100 bits	30 bits	100 bits
DWSF	77.4%	71.8%	69.4%	65.1%	64.9%	62.8%
MuST	81.8%	80.6%	74.9%	72.3%	65.3%	64.1%
WOFA	86.7%	81.1%	71.4%	70.2%	67.7%	64.8%
TIACam	**99.1%**	**98.2%**	**96.6%**	**95.1%**	**97.4%**	**95.2%**

Overall, these supplementary evaluations demonstrate that TIACam provides state-of-the-art robustness not only in camera-based watermarking but also across a broad spectrum of digital and geometric distortions, confirming the general effectiveness of its distortion-invariant feature learning framework.

S5. Architectural Details

Architecture of the Auto-Augmentation Modules. TIACam auto-augmentor contains six learnable distortion modules: Additive, Photometric, Filtering, JPEG, Moiré, and Perspective, that jointly approximate realistic camera degradations. The Additive and Photometric modules share the same architecture: a 512-dimensional latent vector is expanded into a $8 \times 8 \times 128$ tensor through a fully-connected layer, followed by four ConvTranspose2d upsampling blocks that generate a full-resolution residual map using ReLU activations and a final Tanh layer. The Filtering module uses a 3-layer MLP (hidden size 256) to predict a $k \times k$ point-spread function ($k = 3$), enforced to be non-negative via softplus and normalized to sum to one, then applied via depth-wise convolution. Both JPEG and Moiré modules adopt a U-Net with four encoder levels (64, 128, 256, 512 channels) and a 1024-channel bottleneck, paired with symmetric transposed-convolution decoders and skip connections; each predicts a residual that is added to the input image, enabling artifact-specific learning (compression ringing for JPEG, aliasing patterns for Moiré). Finally, the Perspective module uses a ResNet-18 encoder (with the classification layer removed), followed by a 3-layer MLP ($256 \to 128 \to 9$ units) that regresses a 3×3 homography matrix to simulate geometric warping. Together, these six differentiable modules capture noise, blur, photometric variation, JPEG compression, moiré interference, and perspective distortions observed in real camera pipelines.

Architecture of the Feature Extractor and Discriminator. TIACam uses a lightweight MLP-based feature extractor together with a transformer-based discriminator to learn invariant alignment between image–text representations. The feature extractor takes a 768-dimensional CLIP embedding and processes it through an initial Linear–BatchNorm–ReLU–Dropout layer that expands the representation to 1024 units. This is followed by three ResidualBlocks, each containing two Linear layers with BatchNorm1d, ReLU activation, dropout (0.1), and a skip connection. A fusion block (Linear–BN–ReLU–Dropout) refines the intermediate representation, after which a pro-

jection head first reduces the dimensionality to 512 (Linear–BN–ReLU–Dropout) and then maps it to the final 1024-dimensional invariant space through a Linear–BatchNorm layer. Optional ℓ_2 normalization is applied at the output. Overall, the feature extractor forms a 6-layer residual MLP designed to stabilize training and produce distortion-invariant embeddings.

The discriminator jointly processes the extracted image and text features to classify whether the pair originates from a real or adversarially generated match. Both embeddings are projected into a shared 512-dimensional hidden space via a Linear layer. A learnable `[CLS]` token is prepended, forming a 3-token sequence: `[CLS]`, image token, and text token. This sequence is passed through four stacked TransformerBlocks, each consisting of LayerNorm, an 8-head MultiheadAttention module, and a feed-forward MLP with GELU activation and dropout, with residual connections after both attention and MLP sub-layers. A final LayerNorm is applied to the output sequence, and the representation of the `[CLS]` token is fed into a fully connected layer to produce the 2-way classification logits. This combined architecture enables robust adversarial learning by aligning invariant features through the extractor while enforcing semantic consistency through the discriminator.

S6. Detailed Linear Probe Results

Table 8 provides the full numerical results corresponding to the linear probe comparisons summarized in Fig. 4 of the main paper. We report Top-1 and Top-5 accuracy for SimCLR, BYOL, Barlow Twins, VICReg, VIbCReg, and TIACam across four datasets: CIFAR-100, Imagenette, MSCOCO, and Caltech-256, under six distortion types drawn from our camera-style pipeline (additive noise, photometric shift, perspective warp, JPEG compression, Moiré interference, and filtering noise).

These expanded results confirm the trend observed in the main paper: TIACam consistently achieves the highest linear separability across all datasets and distortion categories. In particular, TIACam shows large gains under geometric and appearance-heavy distortions, demonstrating that the learned invariant features retain strong semantic structure even under severe perturbations.

S7. Semantic Sensitivity under Invariance

This experiment complements the "same-caption" analysis in the main paper by examining additional scenario: two visually similar images that differ only in subtle semantic details of their captions. Among 200 such image-caption pairs (e.g., "a photo of cat, child, park bench" vs. "a photo of dog, child, park bench" in Fig. 4), TIACam maintains strong alignment for true image-text pairs, achieving an average cosine similarity of 0.91. In contrast, even small changes in the caption reduce similarity to 0.70 on average.

This behavior confirms that TIACam is not merely invariant to distortions but also sensitive to semantic cues provided by text anchors. The invariant feature space preserves visual robustness while retaining the ability to disambiguate fine-grained linguistic differences, ensuring that each image remains both uniquely represented and semantically grounded.

Figure 4. Semantic disambiguation using controlled captions. Although the two images are visually similar, the model distinguishes them through small caption differences (e.g., "cat" vs. "dog"), yielding higher similarity for matching image–text pairs and lower similarity for mismatched ones.

S8. Failure Cases and Limitations

TIACam is designed for content-preserving distortions: perturbations that may alter the visual appearance but do not change the underlying semantic meaning of the image, which is typically the case for real camera-based watermark extraction. However, when this assumption is violated, the invariant feature space may no longer be recoverable, leading to failures in watermark extraction.

Figure 5. Example failure case caused by severe occlusion. More than two-thirds of the image content is covered, removing or altering the semantic cues required by TIACam to recover the invariant representation. Because the watermark is bound to the image's underlying semantic meaning, a distortion that destroys or changes that meaning essentially yields a new image, making successful extraction impossible.

A representative failure mode occurs under severe occlusion or content removal. When a substantial portion of the image is blocked (empirically, more than two-thirds of the content; see Fig. 5), the semantic cues required by the

Table 8. Linear evaluation (Top-1 / Top-5 accuracy) on CIFAR-100, Imagenette, MSCOCO, and Caltech-256 under six distortion types.

Dataset	Method	Additive		Photometric		Perspective		JPEG		Moiré		Filtering	
		Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
CIFAR-100	SimCLR	72.1	83.3	71.0	79.1	70.5	86.7	62.4	76.5	70.2	81.9	71.3	86.8
	BYOL	75.5	89.2	73.4	88.1	72.6	87.3	61.7	78.0	74.0	86.2	73.9	87.1
	Barlow Twins	71.8	88.7	69.6	85.7	72.9	86.9	65.8	77.6	75.5	87.0	78.2	86.6
	VICReg	74.2	84.5	70.9	90.4	73.9	87.4	65.1	78.6	71.3	83.7	78.1	87.5
	VIbCReg	72.9	88.5	72.9	88.4	76.9	87.4	66.1	82.6	76.3	81.7	77.1	87.5
	TIACam	**81.5**	**95.1**	**77.7**	**94.2**	**83.8**	**93.6**	**73.2**	**85.0**	**83.9**	**89.8**	**82.6**	**93.4**
Imagenette	SimCLR	65.3	80.0	73.1	83.2	72.2	87.5	73.0	88.1	71.7	86.8	72.5	87.3
	BYOL	72.2	86.1	72.0	83.0	73.0	88.2	73.8	88.9	72.4	87.1	73.2	87.9
	Barlow Twins	72.7	87.5	74.6	86.6	72.5	87.8	73.4	88.3	72.0	87.0	72.8	87.5
	VICReg	75.6	90.3	74.4	87.4	73.4	88.5	74.1	89.0	72.7	87.4	73.5	88.1
	VIbCReg	74.3	85.1	72.6	84.4	76.9	87.4	66.1	88.6	76.3	81.7	74.7	84.5
	TIACam	**82.0**	**95.4**	**81.2**	**94.6**	**80.3**	**93.9**	**81.0**	**94.2**	**79.8**	**93.1**	**80.5**	**93.6**
MSCOCO	SimCLR	71.5	87.6	70.7	86.9	70.2	86.3	71.0	84.1	77.0	81.8	70.8	86.5
	BYOL	72.3	88.2	71.4	87.5	70.8	86.8	71.6	87.6	79.5	86.2	71.3	86.9
	Barlow Twins	71.9	87.9	71.1	87.2	70.5	86.6	74.3	83.4	75.2	86.0	71.0	86.7
	VICReg	74.7	88.5	71.7	87.8	71.0	87.1	75.9	85.9	78.7	83.5	71.5	87.2
	VIbCReg	71.7	82.1	76.1	87.7	69.9	82.2	74.3	84.8	77.2	83.7	73.8	84.7
	TIACam	**80.8**	**94.3**	**79.9**	**93.6**	**79.0**	**92.8**	**79.6**	**93.2**	**81.4**	**92.0**	**79.1**	**92.6**
Caltech-256	SimCLR	72.0	87.2	74.2	86.5	70.0	83.0	70.8	86.7	69.8	85.6	70.5	86.3
	BYOL	71.8	89.8	78.0	87.1	73.6	86.5	71.4	88.3	70.2	86.0	71.0	86.6
	Barlow Twins	70.5	86.5	77.7	86.9	70.3	87.3	71.1	85.1	68.9	85.8	70.7	86.5
	VICReg	72.2	90.1	73.4	87.5	72.9	83.8	71.7	87.6	70.5	86.2	71.2	86.9
	VIbCReg	73.7	87.7	74.1	85.1	73.9	87.4	73.3	87.5	68.2	83.7	72.5	84.5
	TIACam	**80.2**	**94.0**	**79.3**	**93.2**	**78.6**	**92.5**	**79.0**	**93.0**	**77.9**	**91.8**	**78.7**	**92.3**

text anchor are no longer visible. In such cases, the feature extractor receives insufficient information to recover the intended invariant representation, causing the extracted features to drift away from the original manifold and leading to reduced or failed watermark decoding.

This limitation is consistent with the design of TIACam: the watermark is associated with the underlying semantics of the image rather than its pixel-level appearance. When those semantics are largely removed or destroyed, the model can no longer align the extracted feature with the registered invariant signature. As a result, TIACam remains robust under content-preserving distortions but cannot recover once the semantic content itself is lost or altered. In such cases, the distortion essentially produces a different new image in terms of meaning, and the model cannot align it with the original invariant signature.

S9. Detailed Comparison with Prior Invariant Feature Learning Methods

Table 9 compares text-guided watermarking [1], InvZW [25], and the proposed TIACam across core design dimensions.

Text-guided watermarking enforces semantic consistency through alignment between image and text embeddings, but does not explicitly model distortions and relies on fixed augmentations. As a result, robustness to complex physical degradations remains limited. In contrast, InvZW formulates invariant feature learning as a distortion-adversarial problem, learning robustness between clean and perturbed images. However, it does not incorporate semantic anchoring, and thus its invariance is not explicitly tied to high-level content.

TIACam unifies these two directions within a single framework. It jointly models distortions through a learnable auto-augmentor and enforces semantic alignment via text anchoring, resulting in invariant representations that are both distortion-robust and semantically grounded. As summarized in Table 9, this integration enables TIACam to combine the strengths of semantic-guided and distortion-adversarial approaches, leading to improved robustness under diverse and realistic perturbations.

S10. Real-World Camera and Print-Capture Datasets Details

To evaluate robustness under real-world acquisition conditions, we construct two datasets: a camera-captured dataset and a print–recapture dataset.

The camera-captured dataset consists of 200 unique images covering diverse indoor and outdoor scenes. These images are displayed and then captured using three different

Table 9. Comparison between Text-Guided Watermarking, InvZW, and the proposed TIACam.

Component	Text-Guided WM	InvZW	TIACam (Ours)
Core idea	Text-guided embedding	Noise-adversarial invariant learning	Text-anchored invariant learning
Text encoder	✓	×	✓
Distortion modeling	×	×	✓
Auto-Augmentor	×	×	✓
Feature alignment	Image ↔ Text	Original ↔ Distorted	Image ↔ Text + Distortion
Architecture	CLIP + watermark head	Generator + Discriminator + Reconstructor	CLIP + invariant projector + watermark head
Invariance source	Semantic guidance	Noise adversarial	Semantic + Auto-Augmentor
Watermark type	Embedded watermark	Zero-watermark	Zero-watermark
Training strategy	Semantic alignment	Adversarial invariant learning	Text-anchored invariant learning

devices, including smartphones and DSLR cameras, under varying illumination conditions, viewing angles, and distances. The resulting samples exhibit realistic distortions such as sensor noise, compression artifacts, color variations, and perspective effects.

The print–recapture dataset contains 200 images that are first printed using three different printers and then recaptured using three cameras. This process introduces additional degradations specific to physical media, including paper texture artifacts, color shifts, and geometric distortions. Each image is captured once using one selected device/setup The final dataset comprises 200 recaptured samples reflecting realistic print-and-capture conditions.

These datasets are used exclusively for evaluation and are never included during training. To ensure fair assessment, there is no overlap between the training data and the real-world camera or print–recapture samples. The model is trained solely on synthetic distortions, while these datasets serve to evaluate generalization to real-world capture scenarios.

S11. Ethical Considerations and Potential Misuse

The proposed TIACam framework is designed to improve the robustness of zero-watermarking under real-world distortions, with primary applications in copyright protection, ownership verification, and content authentication. By enabling reliable watermark extraction without modifying image pixels, the method provides a practical tool for protecting intellectual property in scenarios where images may undergo complex transformations such as camera recapture.

At the same time, we acknowledge that robust watermarking technologies may have broader implications beyond benign use cases. For example, such techniques could potentially be applied to persistent tracking of visual content, surveillance systems, or covert communication. In particular, the ability to reliably associate information with content across transformations may raise concerns regarding user privacy, consent, and misuse in adversarial settings.

To mitigate these risks, we emphasize that TIACam operates under a standard zero-watermarking framework in which watermark generation requires access to a reference codebook that is not publicly shared. This design limits unauthorized extraction or large-scale deployment without controlled access. Furthermore, the method does not embed information directly into image pixels, reducing the risk of unintended propagation or hidden manipulation of visual content.

We believe that responsible deployment of watermarking systems should be guided by clear policies regarding transparency, user consent, and intended use. Future work may explore mechanisms for controlled access, auditability, and usage constraints to ensure that robust watermarking technologies are applied in a manner consistent with ethical standards.

S12. Runtime, Memory, and Deployment Considerations

The full training framework includes a CLIP encoder, invariant feature extractor, Transformer-based discriminator, and a differentiable multi-module Auto-Augmentor. These components are used only during training to learn distortion-robust invariant representations. During inference, the model is significantly simplified: only the CLIP image encoder and the invariant feature extractor are retained. The discriminator and Auto-Augmentor are discarded after training and are not used during deployment.

Training Cost. Training is performed on a single NVIDIA RTX 4090 GPU (24GB). The full model requires approximately 18GB of GPU memory with a batch size of 16. The total training time is approximately 45 hours for 150 epochs on approximately 40K training images. The differentiable Auto-Augmentor introduces additional computation during training, increasing runtime by approximately 10% compared to training without learnable distortions.

Inference Cost. At inference time, only the CLIP image encoder and invariant feature extractor are used. This significantly reduces both memory and runtime overhead. The inference model requires approximately 6GB GPU memory, and processing a single image takes approximately 28

ms on GPU. On CPU, the inference time is approximately 180 ms per image.

Model Size. The CLIP ViT-L/14 encoder contains approximately 304M parameters, while the invariant feature extractor contains approximately 9.2M parameters. The total inference-time model size is therefore approximately 313M parameters. The discriminator and Auto-Augmentor together add approximately 13M additional parameters, but these are used only during training.

Deployment. Since only the invariant feature extractor together with the CLIP image encoder is used at inference time, the deployment cost is comparable to standard CLIP-based feature extraction pipelines. The Auto-Augmentor and discriminator are removed after training, resulting in a lightweight inference pipeline suitable for real-world deployment. In practice, watermark extraction requires a single forward pass through the CLIP encoder and invariant feature extractor, without any additional augmentation modules.