Face Deepfake-aware Recovery via Semantic-driven Facial Representation-based Watermarking

Yuan-Chih Chen
IIS, Academia Sinica
Taipei, Taiwan, ROC
willpower057@iis.sinica.edu.tw

Chun-Shien Lu*
IIS, Academia Sinica
Taipei, Taiwan, ROC
lcs@iis.sinica.edu.tw

Abstract

Existing image watermarking methods typically entangle visual content with spatial structure, making them highly sensitive to geometric transformations and alignment discrepancies, especially in the presence of deepfake manipulations. In this paper, we propose a semantic-driven facial watermarking framework for robust identity recovery. The key idea is to decouple identity-related semantic information from spatial layout and encode it into a compact and spatially robust semantic representation. Specifically, we decompose a face into semantic components and aggregate deep features into component-wise latent representations, which are quantized via independent codebooks and converted into a compact bitstream for embedding. After decoding, the embedded semantic information is recovered and used to reconstruct identity-consistent facial content, even under slight geometric distortions and tampering. Experiments on CelebA-HQ and FFHQ demonstrate that our method significantly outperforms existing watermarking approaches in terms of reconstruction quality, identity preservation, and retrieval accuracy under both photometric and geometric attacks, validating the effectiveness of semantic component-wise encoding for reliable face recovery.

1 Introduction

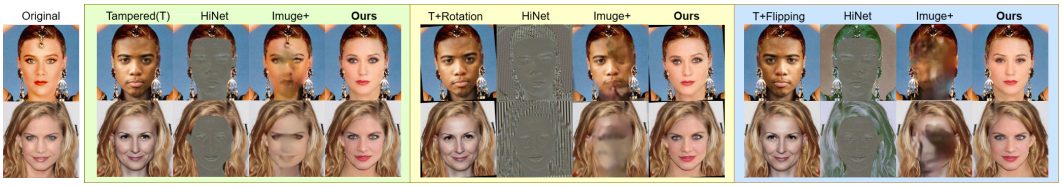


Figure 1: Attacked+Tampered (T) face recovery by HiNet [10], Imuge+ [23], and ours.

Rapid advancement of generative models [17, 9] has significantly lowered the barrier to creating and manipulating digital images, raising serious concerns about image authenticity, particularly for facial content. Since facial images are widely used in identity verification, social media, and digital forensics, recovering original facial information from tampered images has been a paramount issue.

Existing tampered face recovery methods typically follow two paradigms. The first embeds the entire image or facial region into the image using image hiding or reversible embedding techniques [1–3, 18, 23]. Although these methods do not require supervised training with tampered images, they rely on strong spatial correspondence between embedding and decoding. In practice, real-world

*Corresponding author

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

manipulations often involve operations such as cropping, resizing, and translation, which introduce spatial shifts and partial information loss. As a result, the embedded signals may become misaligned or incomplete, leading to degraded recovery performance (Fig. 1). The second paradigm trains recovery models using paired original and tampered images [24]. While supervised learning enables effective restoration, it requires large-scale paired datasets and implicitly assumes prior knowledge of tampering processes, limiting generalization to unseen manipulations. These limitations stem from a common issue: most existing methods encode spatially structured information, making them highly sensitive to spatial transformations and alignment discrepancies. Moreover, directly encoding a full image representation as the watermark is impractical due to limited embedding capacity.

To address this issue, we propose a novel watermarking framework that embeds *facial component semantics* instead of full image content. Specifically, we transform a face image into a compact semantic representation that captures identity-related information. By decoupling semantic content from spatial structure, the proposed representation reduces dependence on precise spatial alignment and improves robustness to geometric distortions (e.g., cropping, resizing, and translation).

Furthermore, we introduce a *semantic token representation* that encodes facial components in a structured and consistent manner. To improve efficiency, we analyze the contribution of different facial components to identity similarity and remove those with minimal impact, enabling a more compact encoding. Finally, we validate the recovered facial information through face retrieval, demonstrating that our representation preserves identity-level semantics for downstream tasks.

Table 1 summarizes the comparison of representative image recovery methods.

Table 1: Existing methods either rely on supervised learning or hide the entire image/face with strong spatial dependency. In contrast, our method encodes compact semantic facial representations and achieves robustness to spatial transformations (ST), including shift, rotation, scaling, cropping, and so on, without requiring supervision or hiding full image content.

Method	**Supervised**	**Entire image hiding**	**Face hiding**	**Facial coding**	**ST-invariance**
HiNet [10]	x	v	x	x	x
Imuge+ [23]	x	v	x	x	x
Bouarroudj [3]	x	v	x	x	x
RecoverMark [1]	x	x	v	x	x
HidingFace [18]	x	x	v	x	x
DFREC [24]	v	x	x	x	x
Ours	x	x	x	v	v

Our contributions are summarized as follows:

- **Watermarking-based face recovery without tampering supervision.** We propose a framework that embeds compact semantic information into images without requiring tampering simulation or paired training data, improving generalization to unseen manipulations.
- **Compact semantic encoding for facial representation.** We encode facial component information into a compact representation (e.g., 128-bit), significantly reducing capacity requirements compared to full image or token-based representations.
- **Facial component importance analysis.** We identify and remove components with limited contribution to identity similarity, enabling more efficient encoding.
- **Identity-level evaluation via face retrieval.** We demonstrate that the recovered faces preserve identity semantics by successfully retrieving the original images from a database.

2 Related Work

2.1 Limitations of Image-wise Hiding ([2, 3, 10, 23])

To our knowledge, existing image hiding methods entangle visual content and spatial structure of an image I in the hidden representation $I_h \equiv (I_{\text{content}}, I_{\text{position}})$, where I_{content} denotes appearance information (e.g., texture and semantics) and I_{position} represents the spatial configuration of pixels. Then, the hidden message I_h is embedded into I via a watermark encoding process $E_w()$ as $I_c = E_w(I, I_h)$ to obtain the container image I_c.



(a) Cropping region estimated from the original image.

(b) Cropping region estimated from the faceswapped image.

(c) Overlay of both cropping regions on the tampered image.

Figure 2: **Alignment discrepancy.** Although the overall head position remains the same (a)(b), the faceswap process alters the geometric relationship between facial components (e.g., eyes, nose, and mouth). As a result, face alignment [13] estimates slightly different cropping regions for the original and tampered images. (c) By overlaying both bounding boxes, spatial discrepancy appears, leading to fragile face recovery due to desynchronization of watermarks.

Such entangled encoding leads to $D_w(T(I_c)) \neq T(I_h)$, violating the equivariance property:

$$D_w\big(T(I_c)\big) \approx T(I_h), \tag{1}$$

where $D_w()$ denotes the watermark decoding process, $T(I)(x, y) = I(x - \Delta x, y - \Delta y)$ denotes a spatial transformation, and $\approx$ accounts for potential degradation. However, Eq. (1) is difficult to achieve in practice due to the entanglement between spatial structure and encoded information, especially when parts of the embedded signal are shifted or removed by image editing.

2.2 Face-wise Hiding and Alignment Discrepancy ([1, 18])

To reduce redundancy between the original image and hidden information, some methods focus on hiding only facial regions instead of the entire image. Let $C(\cdot)$ denote a face cropping operation based on landmark detection. The embedding process becomes $I_c = E_w(I, C(I))$.

However, face cropping is not deterministic, as shown in Fig. 2. Given an original image and its manipulated version (e.g., deepfake), the detected facial regions may differ ($C_o(I) \neq C_t(I_t)$), where C_o and C_t denote cropping operations for the original image I and tampered image I_t, respectively. This discrepancy introduces an **alignment error**, which leads to spatial mismatch between embedding and decoding, *i.e.*, $D_w(C_o(I)) \neq D_w(C_t(I_t))$.

2.3 Token-based Image Representations

Recent image representation methods, such as vector-quantized models [7, 19] and Vision Transformer-based tokenization [5, 15, 25], represent images as sequences of discrete tokens. Although these representations are more compact than raw pixels, they still require a substantial number of bits to encode a full image. For example, representing a 256×256 image with a patch size of 16×16 results in $\frac{256}{16} \times \frac{256}{16} = 256$ tokens. Assuming a codebook size $|V| = 1024$, each token requires $\log_2(|V|) = 10$ bits, leading to a total of approximately $16 \times 16 \times \log_2(|V|) \approx 16 \times 16 \times 10 = 2560$ bits. This requirement significantly exceeds the capacity of typical watermarking systems.

Moreover, token-based representations suffer from the *token mismatch problem* [11, 21]. Due to the non-invertibility of image tokenization and detokenization, the token sequence extracted from a reconstructed image may differ from the original one. This issue becomes more severe under common image distortions, leading to inconsistencies in recovery.

3 Proposed Method

To address the limitations discussed in Sec. 2, particularly the sensitivity to spatial transformations and the inefficiency of dense token representations, we propose a semantic-driven facial representation for

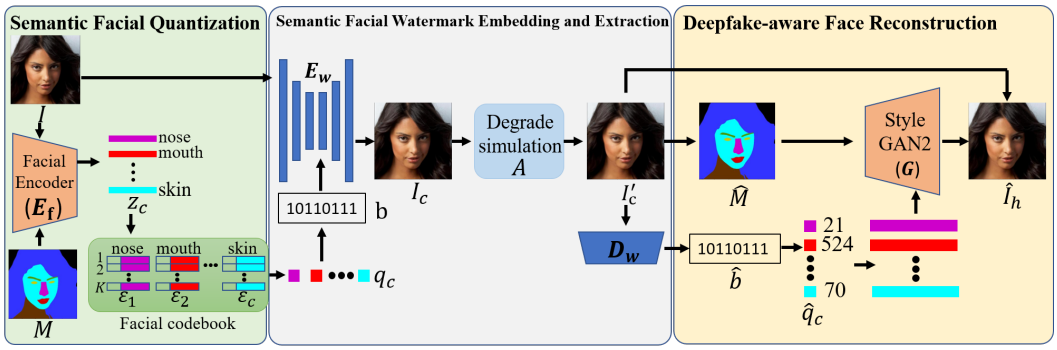


Figure 3: **Overview of the proposed framework.** Given an input image I and segmentation mask M, semantic facial components are aggregated and quantized into discrete indices $\{q_c\}$, which are converted into a compact bitstream $\mathbf{b}$ and embedded into the image to produce the container image I_c. After potential manipulation, a tampered container image I'_c is used to predict the mask $\hat{M}$ and recover the embedded bitstream $\hat{\mathbf{b}}$. The decoded indices $\{\hat{q}_c\}$ are used to reconstruct semantic facial features, which, together with $\hat{M}$, are fed into a generator G for face recovery.

robust and compact watermarking. The core idea is to decouple identity-related semantic information from spatial structure, and encode only the essential facial semantics into a compact bitstream.

As illustrated in Fig. 3, our framework consists of three main stages: (1) Design of adaptive semantic watermark, (2) Watermark emebdding and extraction, and (3) Deepfake-aware recovery, where the embedded information is extracted and used to reconstruct identity-consistent facial content. For clarity, we summarize the complete pipeline of proposed semantic-driven watermarking framework in Algorithm 1 of Appendix A.

3.1 Design of Adaptive Semantic Watermark

3.1.1 Component-wise Representation

To address the limitation found from the existing works, a desirable facial representation should: (1) be invariant or robust to slight spatial transformations, and (2) preserve semantic identity information independent of spatial alignment. We propose decoupling the semantic content I_{semantic} from spatial structure I_{position} of a face within image I and encode only the semantic content as:

$$I_c = E_w(I, I_{\text{semantic}}), \tag{2}$$

where I_{semantic} represents identity-related facial information. The spatial structure I_{position} is not embedded but provided externally (e.g., via face alignment or segmentation). In practice, the semantic representation I_{semantic} will be converted into a compact bitstream $\mathbf{b}$ in the following subsections.

3.1.2 Semantic Facial Component Quantization

Given an input image $I \in \mathbb{R}^{H\times W\times 3}$, we first obtain its corresponding face segmentation map M using an off-the-shelf pre-trained face parsing model M_s [8, 16]. The segmentation map assigns each pixel to one of $|\mathcal{C}| = 12$ predefined semantic facial components, i.e., $M(i,j) \in \mathcal{C}$, where $\mathcal{C}$ includes regions of background, lip, eyebrows, eyes, hair, nose, skin, ears, neck, mouth, eye glass, and ear rings. Subsequently, a facial encoder E_f (IR-SE-50 [6]) extracts dense facial feature maps as:

$$h_f = E_f(I), \quad h_f \in \mathbb{R}^{H'\times W'\times d}, \tag{3}$$

where H' and W' denote the spatial dimensions, and d represents the channel dimension.

Semantic Facial Component Aggregation. To instantiate the abstract semantic representation I_{semantic}, we decompose the face into a set of semantic components and represent each component with a compact latent vector. To address the spatial sensitivity and token inconsistency issues discussed in Sec. 2, we propose to aggregate the facial feature maps h_f into semantic facial component latents using the segmentation map M. This aggregation significantly reduces the number of latent tokens

by summarizing spatial features into a small set of semantic components, which in turn lowers the overall bit requirement for representation. Specifically, for each facial component $c \in \mathcal{C}$, we compute a global average pooling with feature map h_f over the corresponding region defined in Ω_c as:

$$\mathbf{z}_c = \frac{1}{|\Omega_c|} \sum_{(i,j)\in\Omega_c} h_f(i,j), \tag{4}$$

where $\mathbf{z}_c \in \mathbb{R}^d$ and $\Omega_c = \{(i,j) \mid M_{\text{resized}}(i,j) = c\}$ denotes the set of spatial locations belonging to component c in the resized segmentation map, and $|\Omega_c|$ is the number of pixels in that region. This process transforms the spatially dense feature map h_f in a component-wise manner into a compact set of semantic latent vectors: $Z = \{\mathbf{z}_c\}_{c\in\mathcal{C}}$. Thus, the semantic representation is instantiated as $I_{\text{semantic}} \triangleq Z = \{\mathbf{z}_c\}_{c\in\mathcal{C}}$. Unlike fixed spatial patches, this semantic representation is inherently more robust to spatial misalignment and translation, thereby alleviating the violation of Eq. (1).

Component-wise Vector Quantization. To obtain a discrete and compact representation suitable for embedding, each continuous semantic latent $\mathbf{z}_c$ is quantized using a learnable codebook $\mathcal{E} = \{\mathbf{e}_k\}_{k=1}^K$ (see Sec. 3.1.3), where $\mathbf{e}_k \in \mathbb{R}^d$ denotes the k-th codeword. The quantization process assigns each component latent to its nearest codeword in the Euclidean space:

$$q_c = \arg\min_k \|\mathbf{z}_c - \mathbf{e}_k\|_2^2, \qquad \hat{\mathbf{z}}_c = \mathbf{e}_{q_c}, \tag{5}$$

where q_c is the discrete index corresponding to component c, and $\hat{\mathbf{z}}_c$ is the quantized latent vector.

The final discrete representation of a face is thus given by the set of component-wise indices:

$$\mathbf{q} = \{q_c\}_{c\in\mathcal{C}}. \tag{6}$$

Compared with traditional patch-wise tokenization, which represents an image using a large number of spatially localized tokens, our method produces a compact set of component-level tokens (e.g., $|\mathcal{C}| = 12$), where each token encodes a semantically meaningful facial region. This significantly reduces redundancy while improving interpretability.

3.1.3 Facial Bitstream Representation

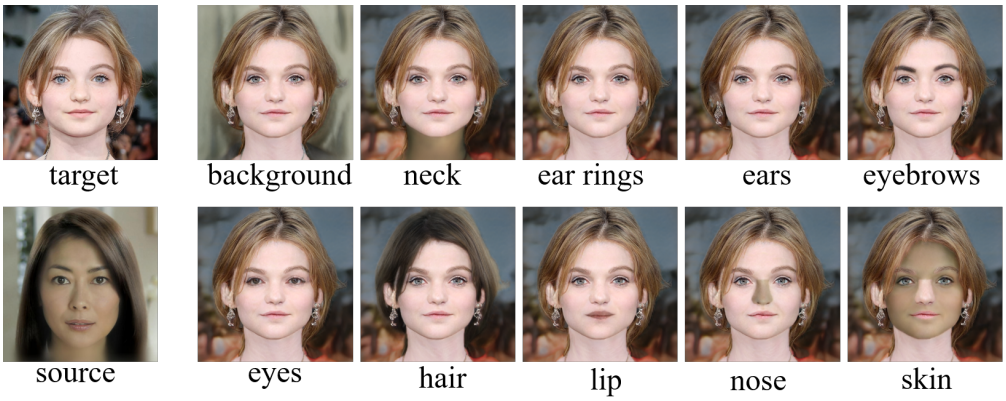
With the quantized semantic facial component indices $\mathbf{q} = \{q_c\}_{c\in\mathcal{C}}$ in Eq. (6), the next step is to convert them into a compact and robust binary representation as the watermark below.

Independent Facial Codebook. Let $\mathcal{C}$ denote the set of facial components and let $\mathcal{E}_c$ be the codebook associated with component $c \in \mathcal{C}$. Instead of using a single shared codebook for all components, we design *independent codebooks* for facial components. This design allows each codebook to specialize in the semantic feature distribution of its corresponding component, rather than requiring a shared codebook to represent heterogeneous facial regions. Please see the quantization loss in Stage 2 of training objectives in Appendix B.

With a independent codebook of size $|\mathcal{E}|$, the total number of bits required to represent a face is: $B_{\text{indep}} = \sum_{c\in\mathcal{C}} \log_2(|\mathcal{E}_c|)$. Assuming there are $|\mathcal{C}| = 12$ facial components and each independent codebook has $|\mathcal{E}_c| = 1024$ entries, the total number of bits is: $B_{\text{indep}} = 12 \times \log_2(1024) = 12 \times 10 = 120$ bits. Furthermore, independent codebooks offer the flexibility to allocate different codebook capacities according to the semantic complexity of each facial component, potentially reducing redundant entries for simpler components while providing greater representational capacity for more complex ones.

This is significantly more compact than traditional patch-wise tokenization methods (e.g., VQGAN [7] or ViT-based encoders [15]), which typically require thousands of bits (e.g., around 2560 bits for a 256×256 image).

Facial Component Importance Analysis While the independent codebook design significantly reduces the bit requirement, not all facial components contribute equally to identity perception. In practical face recognition systems, identity similarity is commonly measured using deep recognition models such as ArcFace [6] and CosFace [20]. Inspired by this, we analyze the importance of each facial component based on identity preservation.



(a) Component-wise replacement visualization.

ArcFace			CosFace		
Rank	**Component**	ΔS_c	**Rank**	**Component**	ΔS_c
1	Nose	0.0996	1	Eyeglasses	0.0910
2	Skin	0.0947	2	Skin	0.0876
3	Eyeglasses	0.0722	3	Nose	0.0784
4	Hair	0.0386	4	Background	0.0688
5	Lip	0.0334	5	Hair	0.0571
6	Background	0.0319	6	Eyebrows	0.0302
7	Eyebrows	0.0309	7	Eyes	0.0237
8	Eyes	0.0285	8	Lip	0.0195
9	Mouth	0.0145	9	Mouth	0.0083
10	Ears	0.0051	10	Neck	0.0072
11	Neck	0.0019	11	Ears	0.0070
12	Earrings	-0.0291	12	Earrings	-0.0069

(b) Facial component importance rankings and identity similarity drops (ΔS_c) under ArcFace and CosFace.

Figure 4: **Facial component importance analysis.** Each facial component is individually replaced using a face swapping method [16], and the resulting identity similarity with the target image is measured using ArcFace [6] and CosFace [20]. (a) Visualization of component-wise replacement. (b) Quantitative identity similarity drop (ΔS_c) and corresponding component rankings under ArcFace and CosFace, averaged over 10,000 samples, where larger values indicate greater importance for identity preservation.

As illustrated in Fig. 4a, we randomly sample two images from the CelebA-HQ dataset [12]. Using a face swapping technique [16], we replace a single facial component c from the source image with that from the target image while keeping other components unchanged. The identity similarity between the manipulated image and the target image is then computed using ArcFace and CosFace. The decrease in similarity (ΔS_c) indicates the importance of the corresponding component: $\Delta S_c = S(I_{\text{target}}, I_{\text{target}}) - S(I_{\text{swap}(c)}, I_{\text{target}})$, where $S(\cdot,\cdot)$ denotes the cosine similarity in the ArcFace/CosFace embedding space, and $I_{\text{swap}(c)}$ represents the image with only component c swapped.

The results, summarized in Fig. 4b, reveal that components such as the nose, skin, eyeglasses, and hair have a significant impact on identity, whereas ears, neck, and earrings contribute minimally. Consequently, less informative components can be excluded to reduce the bit requirement. In particular, background regions are excluded in all settings, as they do not provide reliable identity cues despite potentially affecting similarity scores. For identity-centric tasks (e.g., face retrieval), encoding only highly discriminative facial components is sufficient. For face recovery, we additionally retain structurally supportive components such as the neck to improve visual coherence and natural appearance. This design allows a flexible trade-off between compactness and reconstruction quality.

Compact Bitstream Construction Based on the importance analysis, we define a subset of essential facial components $\mathcal{C}^* \subseteq \mathcal{C}$. The final watermark bitstream is constructed by concatenating

the binary representations of the indices corresponding to these selected components:

$$\mathbf{b} = \bigoplus_{c \in \mathcal{C}^*} \text{bin}(q_c), \tag{7}$$

where $\text{bin}(\cdot)$ converts an index into its fixed-length binary representation and $\bigoplus$ denotes the concatenation operator. For example, if there are $|\mathcal{C}^*| = 8$ important components and each codebook has 1024 entries, the total number of bits required becomes: $B_{\text{final}} = 8 \times \log_2(1024) = 8 \times 10 = 80$ bits.

3.2 Semantic Watermark Embedding and Extraction

After obtaining the compact discrete representation, $\mathbf{b}$ (Eq. (7)), of facial components, it is the semantic watermark. We employ a deep neural watermarking framework consisting of an encoder E_w and a decoder D_w. The embedding process is $I_c = E_w(I, \mathbf{b})$ defined in Eq. (2), where $I \in \mathbb{R}^{H \times W \times 3}$ is an original image and I_c is the container image carrying the hidden facial information.

To simulate real-world distortions and improve robustness, a differentiable attack layer $\mathcal{A}(\cdot)$ is introduced during training: $I'_c = \mathcal{A}(I_c)$, where $\mathcal{A}$ includes geometric and photometric transformations such as cropping, resizing, flipping, translation, and compression. This design encourages the embedded watermark to remain recoverable under practical conditions.

In addition to standard geometric and photometric transformations, the attack layer $\mathcal{A}(\cdot)$ also includes a masking operation guided by the face parsing model M_s. Specifically, the facial region is selectively masked out to simulate partial information loss in identity-related areas. Unlike prior works [24] that explicitly simulate face swapping, our masking-based degradation strategy avoids coupling the recovery model to a particular face-swapping implementation.

At the decoding stage, the watermark decoder D_w predicts the embedded bitstream: $\hat{\mathbf{b}} = D_w(I'_c)$, and the recovered indices are then obtained via:

$$\hat{q}_c = \text{bin}^{-1}(\hat{\mathbf{b}}), \quad c \in \mathcal{C}. \tag{8}$$

3.3 Deepfake-aware Face Reconstruction

Let I_{df} denote the resulting manipulated image. After extracting the embedded watermark and recovering the discrete facial component indices $\{\hat{q}_c\}_{c \in \mathcal{C}^*}$ from I_{df}, the corresponding quantized latent vectors are retrieved from the independent codebooks. These latent representations encode the semantic appearance of the essential facial components.

To synthesize a photorealistic face, we employ a StyleGAN2-based generator [14] as the reconstruction network. Specifically, the recovered component-wise latent codes, together with a predicted facial segmentation mask, are provided as inputs to the generator. The segmentation mask $\hat{M}$ is obtained using the same face parsing model M_s applied to I_{df}, ensuring spatial consistency during reconstruction. The generation process can be formulated as:

$$\hat{I}_{\text{rec}} = G\left(\{\hat{q}_c\}_{c \in \mathcal{C}^*}, \hat{M}\right), \tag{9}$$

where $G(\cdot)$ denotes the StyleGAN2-based generator and $\hat{I}_{\text{rec}}$ is the reconstructed facial image containing the identity information encoded in the watermark.

Component-aware Image Fusion. While the generator produces a complete facial image, not all facial components need to be replaced. Based on the *Facial Component Importance Analysis* described in Sec. 3.1.3, only the identity-sensitive components (e.g., nose, skin, eyes, eyebrows, and lips) are reconstructed, whereas less informative regions (e.g., ears and earrings) are preserved from the deepfake or tampered image to maintain background consistency and visual realism.

Let $\mathcal{C}^*$ denote the set of important facial components. A binary fusion mask M_{imp} is constructed from the predicted segmentation map $\hat{M}$ such that:

$$M_{\text{imp}}(x, y) = \begin{cases} 1, & \text{if } \hat{M}(x, y) \in \mathcal{C}^*, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

The final embedded watermark/image $\hat{I}_h$ can be recovered by blending the reconstructed face $\hat{I}_{\text{rec}}$ with the tampered image I_{df}:

$$\hat{I}_h = M_{\text{imp}} \odot \hat{I}_{\text{rec}} + (1 - M_{\text{imp}}) \odot I_{\text{df}}, \tag{11}$$

where $\odot$ denotes element-wise multiplication. This component-aware fusion strategy ensures accurate identity restoration while preserving non-essential regions, leading to visually coherent and realistic results even in the presence of alignment discrepancies.

4 Experimental Results

4.1 Experimental Setup

Datasets. We first conducted experiments on CelebA-HQ [12], which contains 30,000 high-quality facial images at resolution 1024×1024. We used 28,000 images for training and 2,000 images for testing.[2] For experiments on cross-dataset generalization on FFHQ [13], please see Appendix D.1.

Implementation Details. All images were resized to 1024×1024 during training unless otherwise specified. The identity similarity was measured using a pre-trained ArcFace model [6]. For the semantic quantization, we used independent codebooks with sizes $\{1024, 4096, 8192\}$ for ablation. The watermarking model [4, 22] followed a U-Net-style architecture with skip connections. The total embedded bitstream was fixed to 128 bits unless otherwise stated. The training objectives and loss functions are described in Appendix B.

Evaluation Metrics. We evaluated the performance from three perspectives: (1) **Reconstruction Quality**: Measured by PSNR and SSIM between the recovered image/hidden information $\hat{I}_h$ and the ground-truth I_h; (2) **Identity Consistency**: Measured by cosine similarity in the ArcFace embedding space [6]; and (3) **Retrieval Accuracy**: Top-1|Top-5 accuracy of face retrieval using the recovered image $\hat{I}_h$. The query images are drawn from the test set, while retrieval is performed over the entire dataset (training + test sets) to increase task difficulty.

Attack Settings. To evaluate robustness, we followed [22, 27, 28] to simulate both photometric and geometric distortions that were applied to both the container I_c and original I images: (1) Photometric: JPEG compression (quality=70), Gaussian blur, and Gaussian noise; and (2) Geometric: Rotation ($\pm 10^\circ$), Scaling (crop followed by resize), Translation, and Horizontal flipping. The geometric attacks evaluate robustness to spatial inconsistencies that may arise from repeated face detection, cropping, and alignment operations. The faceswapping approach E4S [16] was adopted as the tampering method. Note that the "tampered" images indicated in the following tables and figures represent those that were face swapped only but did not suffer any further attacks.

4.2 Recovery Comparison with Existing Methods–Evaluation Metric (1)

Our method was compared with representative approaches[3], HiNet [10] and Imuge+ [23], under both photometric and geometric attacks. Table 2 shows the reconstruction performance under different photometric distortions. We can observe that (1) Existing methods suffer significant degradation under blur and noise. and (2) Our method maintains stable performance across all distortions. This is because our representation encodes semantic information rather than pixel-level details, making it inherently robust. For geometric transformatons, Table 3 shows that traditional hiding methods exhibit severe degradation due to spatial misalignment, but our method achieves consistent performance, demonstrating strong robustness to translation, rotation, and scaling.

4.3 Identity Preservation and Retrieval–Evaluation Metrics (2)&(3)

Beyond pixel-level reconstruction, we verified whether the recovered faces preserve identity information and retrieval performance under photometric (Table 4) and geometric (Table 5) distortions.

[2] Actually, our method can be applied to the images with multiple faces, wherein each face is cropped and aligned for watermarking. That is the reason why resolving alignment errors is important. Nevertheless, the existing datasets, including CelebA-HQ [12] and FFHQ [13], are only built with a single face within an image.

[3] The prior studies [1, 18, 24] discussed in Sec. 2 were not included for quantitative comparison because official codes or pretrained models were not publicly available at the time of writing, making reproducible evaluation difficult.

Table 2: **Comparison under photometric distortions on CelebA-HQ [12].** Recovery quality is evaluated using SSIM and PSNR under JPEG compression, Gaussian blur, and Gaussian noise. Higher values indicate better recovery quality. $|\mathcal{E}|$ denotes the codebook size (Sec. 3.1.3).

	Tampered (T) only		T+JPEG		T+Gaussian Blur		T+Gaussian Noise	
Methods \ Metric	SSIM (↑)	PSNR (↑)	SSIM (↑)	PSNR (↑)	SSIM (↑)	PSNR (↑)	SSIM (↑)	PSNR (↑)
Imuge+ [23]	**0.9234**	**29.8463**	0.9021	**28.4641**	0.6963	18.6042	0.8886	27.7838
HiNet [10]	0.8368	22.1404	0.4980	15.9210	0.4745	15.5920	0.1014	15.0147
Ours($|\mathcal{E}| = 1024$)	0.9046	27.5890	0.9009	27.1144	0.9023	27.2763	0.9031	27.4405
Ours($|\mathcal{E}| = 4096$)	0.9158	28.4643	**0.9132**	28.0665	0.9121	27.8947	**0.9121**	27.9031
Ours($|\mathcal{E}| = 8192$)	0.9156	28.6303	**0.9132**	28.2396	**0.9124**	**28.0484**	0.9120	**28.0151**

Table 3: **Comparison under geometric distortions on CelebA-HQ [12].** Recovery quality is evaluated using SSIM and PSNR under rotation, scaling, translation, and horizontal flipping. Higher values indicate better robustness to spatial transformations.

	Tampered (T) only		T+Rotation		T+Scaling		T+Translation		T+Flipping	
Methods \ Metric	SSIM (↑)	PSNR (↑)	SSIM (↑)	PSNR (↑)	SSIM (↑)	PSNR (↑)	SSIM (↑)	PSNR (↑)	SSIM (↑)	PSNR (↑)
Imuge+ [23]	**0.9234**	**29.8463**	0.6482	20.3316	0.8549	24.1396	0.7069	22.4443	0.8066	21.0542
HiNet [10]	0.8368	22.1404	0.3756	15.9873	0.4190	15.9533	0.4862	15.1980	0.7240	20.4477
Ours($|\mathcal{E}| = 1024$)	0.9046	27.5890	0.9058	27.4541	0.8674	26.2212	0.8977	27.2956	0.9025	26.7415
Ours($|\mathcal{E}| = 4096$)	0.9158	28.4643	0.9103	27.8305	0.8723	26.5302	0.9016	27.6201	**0.9061**	27.0009
Ours($|\mathcal{E}| = 8192$)	0.9156	28.6303	**0.9122**	**28.2792**	**0.8744**	**26.6370**	**0.9027**	**28.0627**	0.9057	**27.0572**

Identity preservation (Sim) was measured using ArcFace cosine similarity [6] and retrieval performance was reported using Top1|Top5 face retrieval accuracy. In terms of "Identity Similarity," our method achieves higher identity similarity compared to baselines, especially under distortions. For "Face Retrieval" by matching the recovered image with those from the CelebA-HQ database [12], our method achieves significantly higher retrieval accuracy, indicating that the recovered faces retain sufficient identity semantics. In Appendix C, we demonstrate the visual results for both identity preservation and face retrieval.

Table 4: **Identity preservation and retrieval performance under photometric distortions on CelebA-HQ [12].** Identity similarity (Sim) is measured using ArcFace [6]. Retrieval performance is reported using Top1|Top5 face retrieval accuracy. For Ours (CosFace evaluation), the recovery model trained with ArcFace remains unchanged, while CosFace is used to evaluate identity similarity and face retrieval performance. Higher values indicate better identity preservation and retrieval robustness.

	Tampered (T) only		T+JPEG		T+Gaussian Blur		T+Gaussian Noise	
Methods \ Metric	Sim (↑)	Top1|Top5 (↑)	Sim (↑)	Top1|Top5 (↑)	Sim (↑)	Top1|Top5 (↑)	Sim (↑)	Top1|Top5 (↑)
Imuge+ [23]	0.696	**0.760|0.874**	0.622	0.434|0.598	0.304	0.000|0.000	0.641	**0.737|0.857**
HiNet [10]	0.452	0.010|0.020	0.276	0.000|0.000	0.296	0.000|0.003	0.181	0.000|0.000
Ours($|\mathcal{E}| = 1024$)	0.700	0.481|0.709	0.689	0.435|0.655	0.701	0.489|0.719	0.684	0.457|0.682
Ours($|\mathcal{E}| = 4096$)	0.722	0.592|0.795	0.713	0.517|0.747	0.710	0.593|0.790	0.710	0.572|0.782
Ours($|\mathcal{E}| = 8192$)	**0.724**	0.667|0.840	**0.714**	**0.574|0.758**	**0.722**	**0.658|0.838**	**0.717**	0.623|0.810
Ours (CosFace evaluation)	0.791	0.760|0.878	0.789	0.752|0.882	0.791	0.767|0.886	0.790	0.764|0.881

4.4 Bit Accuracy Analysis

We further evaluate the reliability of watermark decoding using bit accuracy. As shown in Table 6, our method achieves consistently high bit recovery under both photometric and geometric distortions. We compared with EditGuard [27] and InvisMark [22]. While EditGuard performs well under photometric distortions, its bit accuracy drops significantly under geometric transformations (e.g., rotation, scaling, and translation), approaching random guessing (≈ 0.5). This indicates that its embedded signal remains sensitive to spatial misalignment. InvisMark is designed to be robust against geometric distortions. However, under our evaluation setting that includes content-level manipulation (i.e., face swapping) followed by additional attacks, its bit accuracy degrades notably. This suggests that, while InvisMark is effective against geometric transformations, it is less robust to severe semantic changes in identity-centric scenarios. In contrast, our method consistently maintains higher bit accuracy across geometric attacks.

Table 5: **Identity preservation and retrieval performance under geometric distortions on CelebA-HQ [12].** Higher values indicate better robustness under spatial transformations.

Methods \ Metric	Tampered (T) only		T+Rotation		T+Scaling		T+Translation		T+Flipping	
	Sim (↑)	Top1|Top5 (↑)	Sim (↑)	Retr. Acc (↑)	Sim (↑)	Top1|Top5 (↑)	Sim (↑)	Top1|Top5 (↑)	Sim (↑)	Top1|Top5 (↑)
Imuge+ [23]	0.696	**0.760|0.874**	0.421	0.023|0.031	0.504	0.015|0.050	0.476	0.030|0.070	0.411	0.004|0.009
HiNet [10]	0.452	0.009|0.020	0.302	0.000|0.000	0.276	0.000|0.001	0.336	0.004|0.010	0.412	0.004|0.012
Ours($\|\mathcal{E}\| = 1024$)	0.700	0.481|0.709	0.690	0.411|0.618	0.649	0.352|0.541	0.688	0.443|0.631	0.693	0.445|0.657
Ours($\|\mathcal{E}\| = 4096$)	0.722	0.592|0.795	0.705	0.510|0.717	**0.663**	0.438|0.638	0.700	0.537|0.744	0.709	0.526|0.759
Ours($\|\mathcal{E}\| = 8192$)	**0.724**	0.667|0.840	**0.712**	**0.551|0.757**	0.661	**0.479|0.645**	**0.707**	**0.564|0.775**	**0.711**	**0.586|0.782**
Ours (CosFace evaluation)	0.791	0.760|0.878	0.781	0.626|0.775	0.736	0.585|0.747	0.778	0.656|0.811	0.781	0.637|0.798

Table 6: **Bit accuracy under photometric and geometric distortions on CelebA-HQ [12].** "G" indicates Gaussian. Higher values indicate more reliable watermark decoding.

Methods \ Metric	Tampered (T) only Bit. Acc (↑)	T+Rotation Bit. Acc (↑)	T+Scaling Bit. Acc (↑)	T+Translation Bit. Acc (↑)	T+Flipping Bit. Acc (↑)	T+JPEG Bit. Acc (↑)	T+G Blur Bit. Acc (↑)	T+G Noise Bit. Acc (↑)
InvisMark [22]	0.607	0.582	0.583	0.586	0.597	0.578	0.606	0.505
EditGuard [27]	**1.000**	0.503	0.503	0.463	0.669	0.984	0.995	0.984
Bouarroudj et al. [3]	**1.000**	0.4973	0.4946	0.4936	0.4970	0.5054	0.5042	0.5047
Ours	**1.000**	**0.988**	**0.979**	**0.988**	**0.993**	**0.995**	**0.997**	**0.991**

4.5 Ablation Study (in Appendix)

In Appendix D, we further study (1) Cross-dataset generalization, (2) Effect of independent codebooks vs. shared codebook across different facial components, and (3) Effect of different codebook sizes.

5 Conclusion

In this paper, we propose a semantic-driven watermarking framework for robust face recovery under image manipulations. By decoupling identity-related semantic information from spatial structure, our method encodes facial content into a compact and transformation-resilient bitstream. The proposed component-wise representation significantly improves robustness to both photometric and geometric distortions, while maintaining high identity consistency and retrieval performance. Overall, this work highlights the importance of semantic representation in watermarking and provides a practical direction for robust and compact identity-preserving image recovery.

Acknowledgements

This work was supported by the National Science and Technology Council (NSTC) with Grants NSTC 114-2221-E-001-010-MY2 and 114-2634-F-001-001-MBK.

References

[1] Haonan An, Xiaohui Ye, Guang Hua, Yihang Tao, Hangcheng Cao, Xiangyu Yu, and Yuguang Fang. Recovermark: Robust watermarking for localization and recovery of manipulated faces. *arXiv preprint arXiv:2602.20618*, 2026.

[2] Riadh Bouarroudj, Fouzi Harrou, Nabil Zerrouki, Feryel Souami, Fatma Zohra Bellala, and Ying Sun. Medical image authentication and self-recovery using fragile watermarking in the frequency domain. In *2025 IEEE 6th International Conference on Image Processing, Applications and Systems (IPAS)*, pages 1–6. IEEE, 2025.

[3] Riadh Bouarroudj, Feryel Souami, Fatma Zohra Bellala, Nabil Zerrouki, Fouzi Harrou, and Ying Sun. Secure and reversible fragile watermarking for accurate authentication and tamper localization in medical images. *Computers and Electrical Engineering*, 123:110072, 2025.

[4] Tu Bui, Shruti Agarwal, and John Collomosse. Trustmark: Robust watermarking and watermark removal for arbitrary resolution images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18629–18639, 2025.

[5] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11315–11325, 2022.

[6] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019.

[7] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[8] Meng-Hao Guo, Cheng-Ze Lu, Qibin Hou, Zhengning Liu, Ming-Ming Cheng, and Shi-Min Hu. Segnext: Rethinking convolutional attention design for semantic segmentation. *Advances in neural information processing systems*, 35:1140–1156, 2022.

[9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[10] Junpeng Jing, Xin Deng, Mai Xu, Jianyi Wang, and Zhenyu Guan. Hinet: Deep image hiding by invertible network. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4733–4742, 2021.

[11] Nikola Jovanović, Ismail Labiad, Tomas Soucek, Martin Vechev, and Pierre Fernandez. Watermarking autoregressive image generation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

[12] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

[13] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[14] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.

[15] Xiang Li, Kai Qiu, Hao Chen, Jason Kuen, Jiuxiang Gu, Bhiksha Raj, and Zhe Lin. Imagefolder: Autoregressive image generation with folded tokens. *arXiv preprint arXiv:2410.01756*.

[16] Zhian Liu, Maomao Li, Yong Zhang, Cairong Wang, Qi Zhang, Jue Wang, and Yongwei Nie. Fine-grained face swapping via regional gan inversion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8578–8587, 2023.

[17] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[18] Xiaofeng Shen, Heng Yao, Shunquan Tan, and Chuan Qin. Hiding face into background: A proactive countermeasure against malicious face swapping. *IEEE Transactions on Industrial Informatics*, 20(8):10613–10623, 2024.

[19] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024.

[20] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5265–5274, 2018. doi: 10.1109/CVPR.2018.00552.

[21] Yihan Wu, Georgios Milis, Ruibo Chen, and Heng Huang. Robust distortion-free watermark for autoregressive audio generation models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

[22] Rui Xu, Mengya Hu, Deren Lei, Yaxi Li, David Lowe, Alex Gorevski, Mingyu Wang, Emily Ching, and Alex Deng. Invismark: Invisible and robust watermarking for ai-generated image provenance. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 909–918. IEEE, 2025.

[23] Qichao Ying, Hang Zhou, Zhenxing Qian, Sheng Li, and Xinpeng Zhang. Learning to immunize images for tamper localization and self-recovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):13814–13830, 2023.

[24] Peipeng Yu, Hui Gao, Jianwei Fei, Zhitao Huang, Zhihua Xia, and Chip-Hong Chang. Dfrec: Deepfake identity recovery based on identity-aware masked autoencoder. *arXiv preprint arXiv:2412.07260*, 2024.

[25] Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An image is worth 32 tokens for reconstruction and generation. *Advances in Neural Information Processing Systems*, 37:128940–128966, 2024.

[26] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[27] Xuanyu Zhang, Runyi Li, Jiwen Yu, Youmin Xu, Weiqi Li, and Jian Zhang. Editguard: Versatile image watermarking for tamper localization and copyright protection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11964–11974, 2024.

[28] Xuanyu Zhang, Zecheng Tang, Zhipei Xu, Runyi Li, Youmin Xu, Bin Chen, Feng Gao, and Jian Zhang. Omniguard: Hybrid manipulation localization via augmented versatile deep image watermarking. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3008–3018, 2025.

A Pseudocode of the Proposed Framework

To provide a clearer and more structured understanding of the proposed method, we present the full pipeline in Algorithm 1. The algorithm summarizes the three key stages of our framework.

Semantic Watermark Generation. Given an input image I, a face parsing model M_s is first applied to obtain the semantic segmentation map. A facial encoder E_f extracts dense feature maps, which are then aggregated into component-wise latent vectors based on the segmentation regions. Each latent vector is quantized using a learnable codebook, and the resulting indices are converted into a compact bitstream $\mathbf{b}$ by selecting the most identity-informative components.

Watermark Embedding and Extraction. The bitstream $\mathbf{b}$ is embedded into the image via a watermark encoder E_w. To improve robustness, a differentiable attack layer $\mathcal{A}(\cdot)$ simulates real-world distortions during training. The embedded information is later recovered using a decoder D_w, producing the estimated bitstream $\hat{\mathbf{b}}$.

Deepfake-aware Reconstruction. The recovered bitstream is decoded into component-wise indices, which are mapped back to quantized latent representations. A generator G reconstructs the facial image using these semantic representations along with a predicted segmentation mask. Finally, a component-aware fusion strategy combines the reconstructed identity-relevant regions with the input image to produce the final output.

Algorithm 1 Semantic-driven Facial Watermarking Framework

Require: Image I
Ensure: Recovered image $\hat{I}_h$

// Semantic watermark generation (Sec. 3.1)
$M \leftarrow M_s(I)$ ▷ Face parsing
$h_f \leftarrow E_f(I)$ ▷ Feature extraction (Eq. 3)
for each component $c \in \mathcal{C}$ **do**
  $\Omega_c \leftarrow \{(i,j) \mid M(i,j) = c\}$
  $\mathbf{z}_c \leftarrow \frac{1}{|\Omega_c|} \sum_{(i,j)\in\Omega_c} h_f(i,j)$ ▷ Component aggregation (Eq. 4)
  $q_c \leftarrow \arg\min_k \|\mathbf{z}_c - \mathbf{e}_k\|_2^2$ ▷ Vector quantization (Eq. 5)
end for
$\mathbf{b} \leftarrow \bigoplus_{c\in\mathcal{C}^*} \text{bin}(q_c)$ ▷ Bitstream construction (Eq. 7)
// Watermark embedding and extraction (Sec. 3.2)
$I_c \leftarrow E_w(I, \mathbf{b})$ ▷ Embedding (Eq. 2)
$I'_c \leftarrow \mathcal{A}(I_c)$ ▷ Attack simulation
$\hat{\mathbf{b}} \leftarrow D_w(I'_c)$
$\{\hat{q}_c\} \leftarrow \text{bin}^{-1}(\hat{\mathbf{b}})$ ▷ Index recovery (Eq. 8)
// Deepfake-aware reconstruction (Sec. 3.3)
$\hat{M} \leftarrow M_s(I'_c)$
$\hat{I}_{\text{rec}} \leftarrow G(\{\hat{q}_c\}_{c\in\mathcal{C}^*}, \hat{M})$ ▷ Face reconstruction (Eq. 9)
$M_{\text{imp}}(x,y) \leftarrow \begin{cases} 1, & \hat{M}(x,y) \in \mathcal{C}^* \\ 0, & \text{otherwise} \end{cases}$
$\hat{I}_h \leftarrow M_{\text{imp}} \odot \hat{I}_{\text{rec}} + (1 - M_{\text{imp}}) \odot I'_c$ ▷ Fusion (Eq. 11)
return $\hat{I}_h$

B Training Objectives and Loss Functions

The proposed framework is trained in two separate stages to ensure stable optimization and effective learning of both watermarking and facial reconstruction capabilities.

Watermark Embedding and Extraction. The first stage focuses on training the neural watermarking model, which consists of an encoder E_w and a decoder D_w. The objective is to embed the

facial bitstream into the container image while enabling reliable recovery under various distortions, as shown in Fig. 5.

1) Bit Recovery Loss. To ensure accurate extraction of the embedded watermark, we employ a binary cross-entropy (BCE) loss between the embedded bitstream $\mathbf{b}$ and the predicted bitstream $\hat{\mathbf{b}}$:

$$\mathcal{L}_{\text{bit}} = \text{BCE}(\hat{\mathbf{b}}, \mathbf{b}). \tag{12}$$

2) Container Image Reconstruction Loss. To maintain high visual fidelity and imperceptibility of the watermark, the similarity between the container image I_c and the original image I is enforced using an ℓ_2 loss:

$$\mathcal{L}_{\text{wm-rec}} = \|I_c - I\|_2^2. \tag{13}$$

The overall loss for the watermarking stage is defined as:

$$\mathcal{L}_{\text{WM}} = \lambda_{\text{bit}}\mathcal{L}_{\text{bit}} + \lambda_{\text{wm}}\mathcal{L}_{\text{wm-rec}}, \tag{14}$$

where λ_{bit} and λ_{wm} are weighting coefficients.

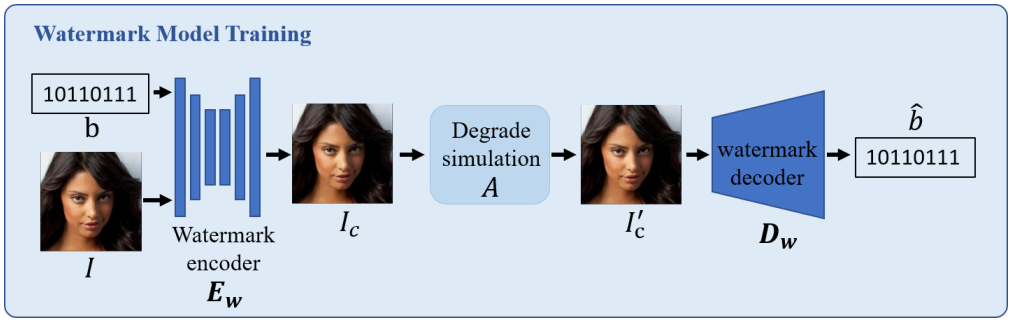


Figure 5: **Illustration of watermarking module training.**

Facial Feature Quantization and Reconstruction. The second stage trains the facial encoder E, the independent codebooks $\{\mathcal{E}_c\}$, and the StyleGAN2-based [14] generator G to accurately reconstruct the hidden facial image from the quantized semantic representations, as shown in Fig. 6. To reduce training costs and improve stability during optimization, we used a pre-trained StyleGAN2 [14].

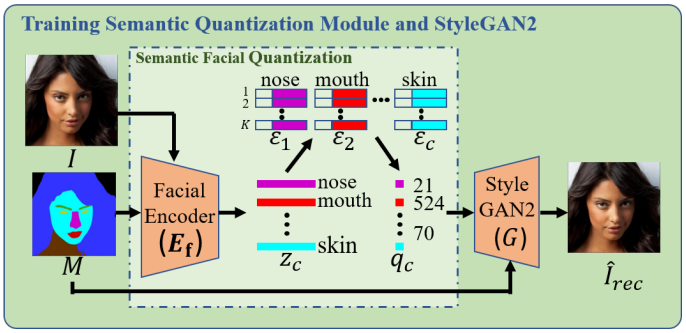


Figure 6: **Illustration of jointly training the facial codebook module and StyleGAN2.**

1) Quantization Loss. To stabilize codebook learning, we adopt the standard vector quantization loss:

$$\mathcal{L}_{\text{VQ}} = \sum_{c \in \mathcal{C}^*} \left(\|\text{sg}[\mathbf{z}_c] - \mathbf{e}_{q_c}\|_2^2 + \beta \|\mathbf{z}_c - \text{sg}[\mathbf{e}_{q_c}]\|_2^2 \right), \tag{15}$$

where $\text{sg}[\cdot]$ denotes the stop-gradient operator and β controls the commitment strength.

2) Pixel-wise Reconstruction Loss. To enforce low-level visual similarity between the reconstructed image $\hat{I}_{\text{rec}}$ and the original image I, we employ a mean squared error (MSE) loss:

$$\mathcal{L}_{\text{pix}} = \|\hat{I}_{\text{rec}} - I\|_2^2. \tag{16}$$

3) Perceptual Loss. To enhance perceptual quality and preserve high-level semantic features, we incorporate the LPIPS loss [26]:

$$\mathcal{L}_{\text{perc}} = \text{LPIPS}(\hat{I}_{\text{rec}}, I). \tag{17}$$

4) Identity Preservation Loss. Maintaining the identity of the reconstructed face is crucial. Therefore, we utilize a pre-trained ArcFace model [6] to compute the cosine similarity between identity embeddings:

$$\mathcal{L}_{\text{id}} = 1 - \cos\left(F_{\text{id}}(\hat{I}_{\text{rec}}), F_{\text{id}}(I)\right), \tag{18}$$

where $F_{\text{id}}(\cdot)$ denotes the ArcFace feature extractor.

5) Adversarial Loss. To enhance the photorealism of the reconstructed facial images, we incorporate an adversarial learning framework based on a discriminator D with a StyleGAN2-based generator G.

Let I denote the ground-truth facial image and let $\hat{I}_{\text{rec}}$ denote the reconstructed image. The adversarial loss is defined as:

$$\mathcal{L}_{\text{adv}} = \mathbb{E}[\log D(I)] + \mathbb{E}[1 - \log D(\hat{I}_{\text{rec}})]. \tag{19}$$

Overall Objective. The total loss for the facial quantization and reconstruction stage is defined as:

$$\mathcal{L}_{\text{FR}} = \lambda_{\text{VQ}}\mathcal{L}_{\text{VQ}} + \lambda_{\text{pix}}\mathcal{L}_{\text{pix}} + \lambda_{\text{perc}}\mathcal{L}_{\text{perc}} + \lambda_{\text{id}}\mathcal{L}_{\text{id}} + \lambda_{\text{adv}}\mathcal{L}_{\text{adv}}, \tag{20}$$

where λ_{VQ}, λ_{pix}, λ_{perc}, λ_{id}, and λ_{GAN} are balancing coefficients.

Training Strategy. The semantic reconstruction network and the watermarking encoder/decoder are trained independently with separate objectives and without sharing gradients. For semantic reconstruction, the facial feature extractor, independent codebooks, and StyleGAN2-based decoder are optimized to achieve high-quality face reconstruction. The StyleGAN2-based decoder is initialized from the pretrained StyleGAN2 model [16], with only the first 13 layers fine-tuned while the remaining layers are frozen. For watermark training, randomly sampled 128-bit semantic codes are used as payloads, and attack-augmented images are employed to improve watermark robustness. This process does not involve the StyleGAN2 generator, GAN losses, or reconstruction-related supervision. Since the two training procedures are independent, their training order does not affect the final model.

Training Cost. All models were trained on NVIDIA V100 GPUs. The semantic quantization model and watermarking network were trained separately. Training the semantic quantization stage takes approximately 2 days on 1×V100 GPUs, while the watermark embedding and extraction network requires around 2 days on 1×V100 GPUs. Overall, the full training pipeline requires about 4 GPU days.

C Visual Results

C.1 Visual Results for Identity Preservation

Fig. 7 presents qualitative comparisons of face recovery with geometric attack setting and without geometric attack setting ("Tampered" only). For the recovery from tampered images without additional distortions, actually only our method can produce visually pleasing quality.

However, when geometric transformations such as rotation and horizontal flipping were further applied to tamppered images, Imuge+ suffers from severe spatial misalignment. Specifically, under rotation attacks, the recovered regions are dominated by the tampered facial content instead of the hidden identity information, indicating failure of hidden information synchronization. Similarly, under horizontal flipping, Imuge+ fails to reconstruct meaningful facial structures and produces heavily blurred recovery results. For HiNet, it entirely fails in recovery.

In contrast, our method consistently preserves facial identity and visual coherence across all attack settings. This demonstrates that the proposed semantic component-wise representation is substantially more robust to spatial transformations than conventional image-wise hiding approaches.

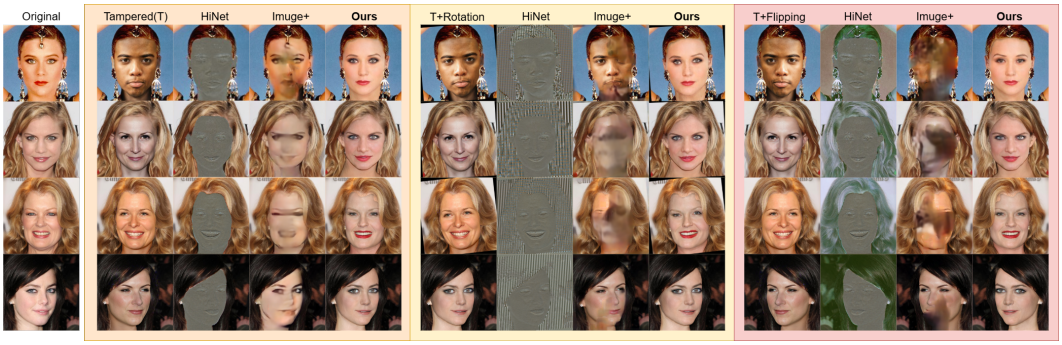


Figure 7: **Qualitative comparison under geometric attacks.** Each group shows the tampered (T) input and the corresponding recovery results of HiNet, Imuge+, and our method under different attack settings, where rotation and horizontal flipping are additionally applied after tampering. While Imuge+ produces more or less degraded recovery quality under tampering only, it fails under geometric transformations due to spatial misalignment, resulting in recovery dominated by tampered content or severe blur artifacts. For HiNet, it fails in recovery entirely. In contrast, our method maintains consistent identity recovery and visual quality under both rotation and horizontal flipping attacks.

C.2 Visual Results for Face retrieval

Fig. 8 presents qualitative face retrieval results under different geometric attack settings. To isolate the influence of transformations, we apply different attacks to the same original image, including tampering only (T), T+Rotation, and T+Flipping. For each case, the recovered image is used as the query to retrieve the Top-5 nearest identities from the CelebA-HQ [12] database using ArcFace embeddings [6].

As expected, the similarity scores gradually decrease when additional geometric distortions are introduced. Nevertheless, the correct identity consistently remains within the top-ranked retrieval results across all attack settings. This demonstrates that the proposed semantic representation preserves discriminative identity information even under spatial transformations.

To further demonstrate more results across different subjects, including failed examples, we provide additional retrieval results using different images under individual attack settings. Specifically, Fig. 9 shows retrieval results under tampering only, Fig. 10 shows results under T+Rotation, and Fig. 11 shows results under T+Flipping. Each figure contains six different query examples to illustrate the consistency and generalization ability of the proposed approach across diverse facial identities.

D Ablation Study

We further study the issues of (1) Cross-dataset generalization on FFHQ, (2) Effect of independent codebooks vs. shared codebook across different facial components, and (3) Effect of different codebook sizes.

D.1 Cross-Dataset Generalization on FFHQ [13]

To further evaluate generalization under a more challenging setting, we conducted experiments on the FFHQ dataset [13]. Specifically, the facial codebook, StyleGAN2 generator, and the watermarking model were trained on CelebA-HQ. The face parsing network,[4] pretrained on CelebA-HQ, is directly applied without fine-tuning, introducing additional cross-dataset discrepancy.

As shown in Tables 7 and 8, our method maintains strong robustness under both photometric and geometric distortions. Notably, compared to prior methods, our approach exhibits substantially smaller performance degradation under spatial transformations, demonstrating that the proposed semantic representation generalizes effectively beyond the training distribution.

[4] https://github.com/VisionSystemsInc/face-parsing.PyTorch

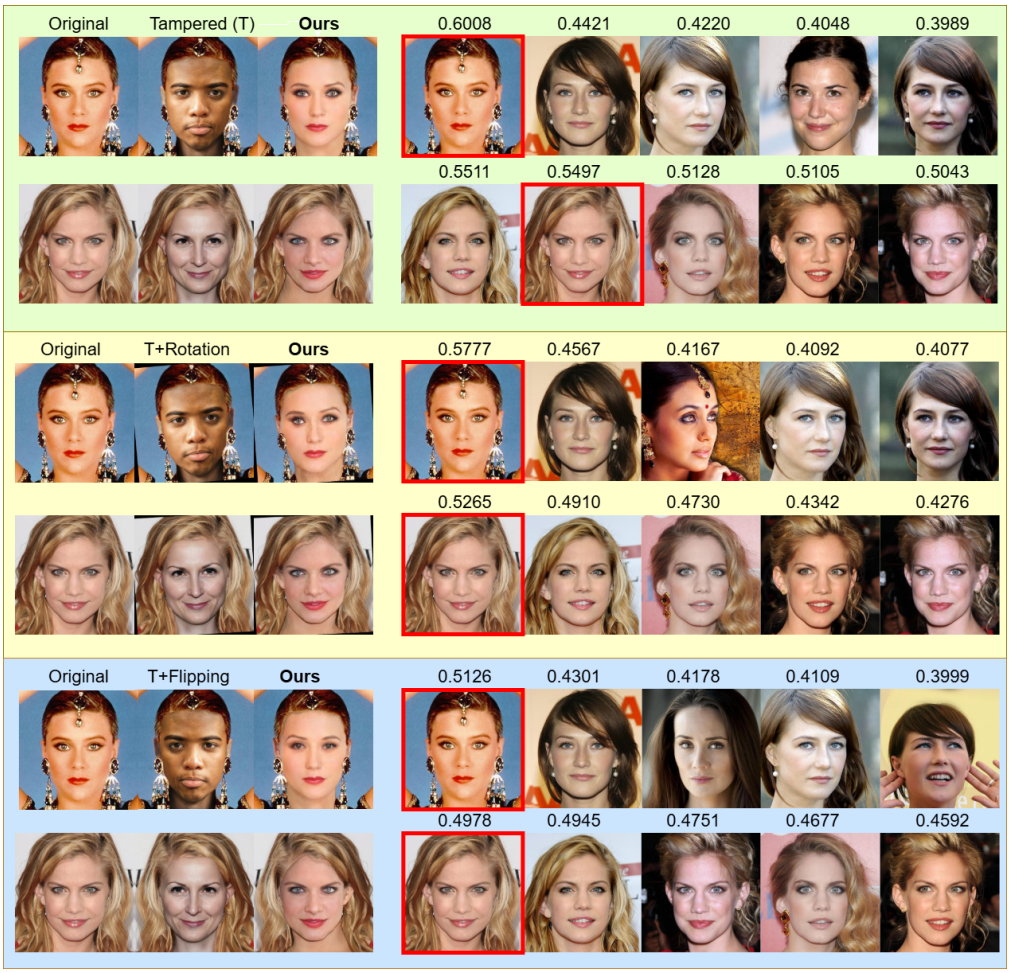


Figure 8: **Top-5 face retrieval results and corresponding similarity scores.** From left to right: the original image, its tampered version, the recovered image, and the top-5 retrieved images using the recovered image as the query. The scores above the retrieved images indicate the face similarity between the recovered image and each retrieved candidate. The ground-truth image (identical to the original image) is highlighted with a red bounding box in the top-5 results.

Despite the increased difficulty, Tables 9 and 10 show that identity similarity and retrieval performance remain competitive. This suggests that the proposed semantic representation captures identity-relevant information in a robust and transferable manner, even under imperfect semantic decomposition.

Furthermore, in comparison with Table 6, the consistently high bit accuracy across all distortion settings in Table 11 indicates that the watermarking module is largely invariant to dataset shifts. This highlights that the stability of the embedded bitstream is decoupled from reconstruction quality, enabling reliable identity recovery even in cross-dataset scenarios.

D.2 Effect of Independent Codebooks vs. Shared Codebook Across Facial Components

We examine the impact of using a shared codebook across all facial components versus assigning independent codebooks to each component. To ensure a fair comparison, we control the embedding capacity by fixing the codebook size per component. Specifically, for a codebook of size $|\mathcal{E}|$, both shared and independent designs encode the same number of bits, i.e., $\log_2(|\mathcal{E}|) \times N_{\text{comp}}$, where $N_{\text{comp}} = |\mathcal{C}^*|$ is the number of facial components. Therefore, the independent design does not introduce additional bit overhead, and any performance difference cannot be attributed to increased capacity.



Figure 9: **Top-5 face retrieval results (tampered only) and corresponding similarity scores.** From left to right: the original image, its tampered version, the recovered image, and the top-5 retrieved images using the recovered image as the query. The scores above the retrieved images indicate the face similarity between the recovered image and each retrieved candidate.

Table 7: **Comparison under photometric distortions on FFHQ [13].** Recovery quality is evaluated using SSIM and PSNR under JPEG compression, Gaussian blur, and Gaussian noise. Higher values indicate better recovery quality. $|\mathcal{E}|$ denotes the codebook size (Sec. 3.1.3).

	Tampered (T) only		T+JPEG		T+Gaussian Blur		T+Gaussian Noise	
Methods \ Metric	SSIM (↑)	PSNR (↑)	SSIM (↑)	PSNR (↑)	SSIM (↑)	PSNR (↑)	SSIM (↑)	PSNR (↑)
Imuge+ [23]	**0.918**	**29.505**	**0.899**	**28.393**	0.698	18.781	**0.916**	**29.243**
HiNet [3]	0.834	22.773	0.514	16.805	0.500	16.423	0.066	17.481
Ours($|\mathcal{E}| = 1024$)	0.892	27.052	0.889	26.795	0.892	27.056	0.890	26.870
Ours($|\mathcal{E}| = 4096$)	0.898	27.827	0.895	27.425	**0.899**	**27.844**	0.895	27.589
Ours($|\mathcal{E}| = 8192$)	0.897	27.792	0.894	27.331	0.897	27.794	0.895	27.482

Tables 12 and 13 report identity preservation and retrieval performance under photometric and geometric distortions. Overall, independent codebooks consistently outperform or match shared codebooks, with more stable behavior under challenging conditions, particularly geometric transformations.

A key observation is that the **independent** configuration with $|\mathcal{E}| = 4096$ achieves performance comparable to, or even better than, the **shared** configuration with a larger codebook size ($|\mathcal{E}| = 8192$), especially under geometric distortions. This demonstrates that component-wise quantization improves representation efficiency: instead of requiring a larger shared vocabulary to model diverse facial regions, independent codebooks enable more effective specialization without increasing the bit budget.

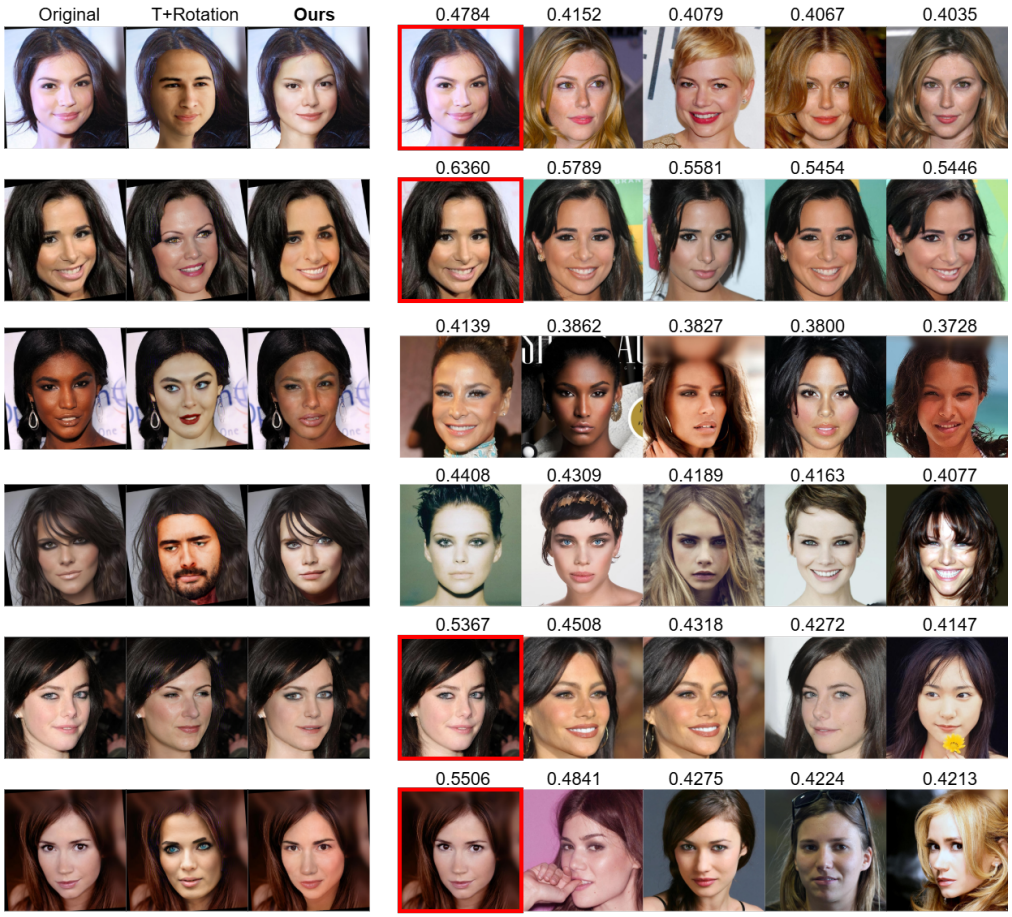


Figure 10: **Top-5 face retrieval results (tampered+rotation) and corresponding similarity scores.** From left to right: the original image, its tampered version, the recovered image, and the top-5 retrieved images using the recovered image as the query. The scores above the retrieved images indicate the face similarity between the recovered image and each retrieved candidate. For each row without the red box indicates a failed retrieval within top-5 retrieval.

Table 8: **Comparison under geometric distortions on FFHQ [13].** Recovery quality is evaluated using SSIM and PSNR under rotation, scaling, translation, and horizontal flipping. Higher values indicate better robustness to spatial transformations.

	Tampered (T) only		T+Rotation		T+Scaling		T+Translation		T+Flipping	
Methods \ Metric	SSIM (↑)	PSNR (↑)	SSIM (↑)	PSNR (↑)	SSIM (↑)	PSNR (↑)	SSIM (↑)	PSNR (↑)	SSIM (↑)	PSNR (↑)
Imuge+ [23]	**0.918**	**29.505**	0.644	20.363	0.848	24.214	0.703	22.469	0.803	21.274
HiNet [10]	0.834	22.773	0.401	16.708	0.449	16.768	0.495	17.072	0.719	20.685
Ours($\lvert\mathcal{E}\rvert = 1024$)	0.892	27.052	0.889	26.843	0.858	25.775	0.883	26.749	0.883	26.0799
Ours($\lvert\mathcal{E}\rvert = 4096$)	0.898	27.827	**0.895**	**27.584**	0.860	26.197	**0.888**	**27.412**	**0.888**	26.5247
Ours($\lvert\mathcal{E}\rvert = 8192$)	0.897	27.792	**0.895**	27.489	**0.895**	**27.489**	0.861	26.182	**0.888**	**27.328**

Importantly, since both designs encode the same number of bits at a given codebook size, the improved performance of independent codebooks does not come at the cost of reduced watermark robustness. In contrast, for shared codebooks to reach similar performance, the codebook size must be increased (e.g., from 4096 to 8192), which directly increases the bit length of the embedded watermark. This expansion can potentially make the watermark more vulnerable under distortions due to the higher information density.

We further observe that increasing the codebook size benefits both designs; however, independent codebooks consistently achieve better identity similarity and retrieval accuracy under spatial transfor-

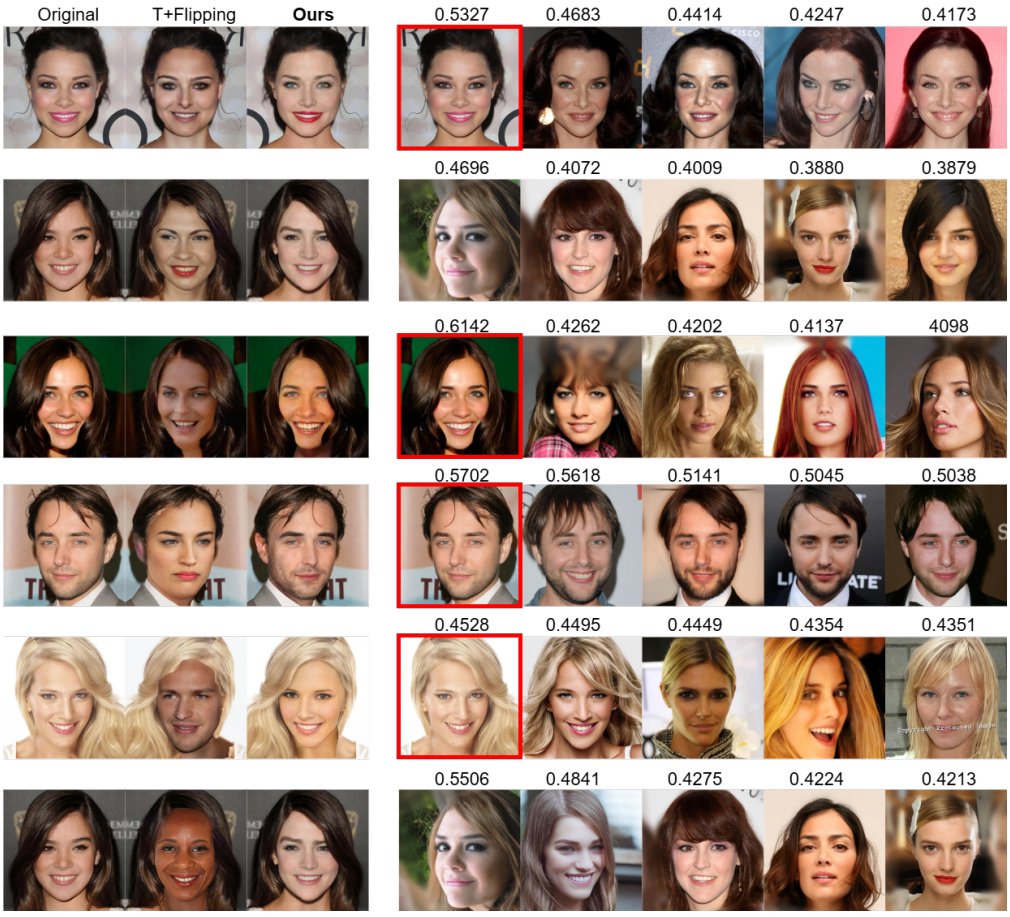


Figure 11: **Top-5 face retrieval results (tampered+flipping) and corresponding similarity scores.** From left to right: the original image, its tampered version, the recovered image, and the top-5 retrieved images using the recovered image as the query. The scores above the retrieved images indicate the face similarity between the recovered image and each retrieved candidate. For each row without the red box indicates a failed retrieval within top-5 retrieval.

Table 9: **Identity preservation and retrieval performance under photometric distortions on FFHQ [13].** Identity similarity (Sim) is measured using ArcFace [6]. Retrieval performance is reported using Top1|Top5 face retrieval accuracy. Higher values indicate better identity preservation and retrieval robustness.

	Tampered (T) only		T+JPEG		T+Gaussian Blur		T+Gaussian Noise	
Methods \ Metric	Sim (↑)	Top1|Top5 (↑)	Sim (↑)	Top1|Top5 (↑)	Sim (↑)	Top1|Top5 (↑)	Sim (↑)	Top1|Top5 (↑)
Imuge+ [23]	**0.695**	**0.736|0.853**	0.625	0.419|0.578	0.300	0.000|0.000	**0.692**	**0.723|0.837**
HiNet [3]	0.448	0.003|0.009	0.281	0.000|0.000	0.304	0.000|0.001	0.357	0.000|0.000
Ours($\|\mathcal{E}\| = 1024$)	0.659	0.387|0.561	0.649	0.338|0.490	0.659	0.384|0.543	0.653	0.346|0.502
Ours($\|\mathcal{E}\| = 4096$)	0.683	0.504|0.707	**0.672**	**0.443|0.623**	**0.683**	0.500|**0.697**	0.676	0.472|0.655
Ours($\|\mathcal{E}\| = 8192$)	0.683	0.512|0.695	0.670	0.441|0.617	**0.683**	**0.511**|0.695	0.676	0.468|0.655

mations such as rotation and scaling. This suggests that disentangling facial components reduces cross-region interference and enhances robustness to spatial misalignment.

In summary, independent codebooks provide a more efficient representation by achieving stronger performance under the same bit budget. In contrast, shared codebooks require larger codebook sizes—and thus longer bitstreams—to reach comparable performance, leading to a less favorable trade-off between representation capacity and robustness.

Table 10: **Identity preservation and retrieval performance under geometric distortions on FFHQ [13].** Higher values indicate better robustness under spatial transformations.

	Tampered (T) only		T+Rotation		T+Scaling		T+Translation		T+Flipping	
Methods \ Metric	Sim (↑)	Top1|Top5 (↑)	Sim (↑)	Retr. Acc (↑)	Sim (↑)	Top1|Top5 (↑)	Sim (↑)	Top1|Top5 (↑)	Sim (↑)	Top1|Top5 (↑)
Imuge+ [23]	**0.695**	**0.736|0.853**	0.442	0.017|0.031	0.424	0.018|0.041	0.472	0.018|0.048	0.403	0.000|0.004
HiNet [3]	0.448	0.003|0.009	0.306	0.000|0.000	0.289	0.000|0.000	0.339	0.002|0.006	0.401	0.002|0.006
Ours($\|\mathcal{E}\| = 1024$)	0.659	0.387|0.561	0.648	0.279|0.433	0.606	0.232|0.383	0.647	0.291|0.471	0.647	0.323|0.505
Ours($\|\mathcal{E}\| = 4096$)	0.683	0.504|0.707	**0.671**	**0.420|0.593**	0.622	**0.324|0.486**	**0.670**	**0.456|0.632**	**0.667**	**0.448|0.623**
Ours($\|\mathcal{E}\| = 8192$)	0.683	0.512|0.695	0.670	0.411|0.579	**0.623**	0.307|0.473	**0.670**	0.438|0.616	**0.667**	0.430|0.604

Table 11: **Bit accuracy under photometric and geometric distortions on FFHQ [13] in terms of cross-dataset evaluation, where the watermarking model was trained on CelebA-HQ but directly evaluated on FFHQ without fine-tuning.** "G" denotes Gaussian. Higher values indicate more reliable watermark decoding, demonstrating strong cross-dataset generalization.

Dataset \ Metric	Tampered (T) only Bit. Acc (↑)	T+Rotation Bit. Acc (↑)	T+Scaling Bit. Acc (↑)	T+Translation Bit. Acc (↑)	T+Flipping Bit. Acc (↑)	T+JPEG Bit. Acc (↑)	T+G Blur Bit. Acc (↑)	T+G Noise Bit. Acc (↑)
FFHQ	**0.996**	**0.989**	**0.979**	**0.992**	**0.994**	**0.976**	**0.997**	**0.984**

D.3 Effect of Different Codebook Sizes

We verify the impact of codebook sizes within the range of $\{1024, 4096, 8192\}$. The corresponding results have been summarized in Tables 2–5, 7–10, and 12–13.

From Tables 7–10, we see that the codebook size of 4096 achieves better performance of cross-dataset generalization. For other results, increasing the codebook size generally improves reconstruction quality, identity preservation, and retrieval performance since a larger codebook provides stronger representational capacity for semantic facial components, enabling more accurate quantization and reconstruction of identity-related features. Similar trends are also observed under both photometric and geometric distortions.

Therefore, by considering the tradeoff between model performance and representational capacity, the codebook size of either 4096 or 8192 is a good choice.

Table 12: **Identity preservation and retrieval performance undershared vs. independent codebooks (photometric distortions). on CelebA-HQ [12].** Identity similarity (Sim) is measured using ArcFace [6]. Retrieval performance is reported using Top1|Top5 face retrieval accuracy. Higher values indicate better identity preservation and retrieval robustness.

Methods \ Metric	Tampered (T) only		T+JPEG		T+Gaussian Blur		T+Gaussian Noise	
	Sim (↑)	Top1|Top5 (↑)	Sim (↑)	Top1|Top5 (↑)	Sim (↑)	Top1|Top5 (↑)	Sim (↑)	Top1|Top5 (↑)
Independent($\|\mathcal{E}\| = 1024$)	0.700	0.481|0.709	0.689	0.435|0.655	0.701	0.489|0.719	0.684	0.457|0.682
Independent($\|\mathcal{E}\| = 4096$)	0.722	0.592|0.795	0.713	0.517|**0.747**	0.710	0.593|0.790	0.710	0.572|0.782
Independent($\|\mathcal{E}\| = 8192$)	**0.724**	**0.667|0.840**	**0.714**	**0.574|0.758**	**0.722**	**0.658|0.838**	**0.717**	**0.623|0.810**
Shared($\|\mathcal{E}\| = 1024$)	0.697	0.480|0.705	0.678	0.410|0.638	0.697	0.479|0.699	0.668	0.348|0.528
Shared($\|\mathcal{E}\| = 4096$)	0.707	0.540|0.727	0.699	0.500|0.703	0.710	0.549|0.755	0.680	0.390|0.590
Shared($\|\mathcal{E}\| = 8192$)	0.723	0.619|0.807	0.711	0.555|0.745	0.711	0.620|0.820	0.709	0.599|0.779

Table 13: **Identity preservation and retrieval performance under shared vs. independent codebooks (geometric distortions) on CelebA-HQ [12].** Higher values indicate better robustness under spatial transformations.

Methods \ Metric	Tampered (T) only		T+Rotation		T+Scaling		T+Translation		T+Flipping	
	Sim (↑)	Top1|Top5 (↑)	Sim (↑)	Retr. Acc (↑)	Sim (↑)	Top1|Top5 (↑)	Sim (↑)	Top1|Top5 (↑)	Sim (↑)	Top1|Top5 (↑)
Independent($\|\mathcal{E}\| = 1024$)	0.700	0.481|0.709	0.690	0.411|0.618	0.649	0.352|0.541	0.688	0.443|0.631	0.693	0.445|0.657
Independent($\|\mathcal{E}\| = 4096$)	0.722	0.592|0.795	0.705	0.510|0.717	**0.663**	0.438|0.638	0.700	0.537|0.744	0.709	0.526|0.759
Independent($\|\mathcal{E}\| = 8192$)	**0.724**	**0.667|0.840**	**0.712**	**0.551|0.757**	0.661	**0.479|0.645**	**0.707**	**0.564|0.775**	**0.711**	**0.586|0.782**
Shared($\|\mathcal{E}\| = 1024$)	0.697	0.480|0.705	0.683	0.396|0.591	0.637	0.310|0.504	0.683	0.401|0.603	0.689	0.412|0.643
Shared($\|\mathcal{E}\| = 4096$)	0.707	0.540|0.727	0.701	0.456|0.678	0.651	0.368|0.576	0.696	0.482|0.684	0.700	0.473|0.718
Shared($\|\mathcal{E}\| = 8192$)	0.723	0.619|0.807	0.710	0.517|0.722	0.662	0.437|0.637	0.702	0.538|0.744	0.710	0.540|0.769

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the main contributions, including the proposed semantic-driven watermarking framework, robustness to geometric distortions, and improved identity-preserving recovery. These claims are supported by both quantitative and qualitative experiments (Sec. 4).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses limitations such as dependence on accurate face parsing, and potential degradation under large occlusions (Sec. D.1). These factors may affect robustness and generalization.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not focus on formal theoretical derivations or proofs; instead, it presents a method supported by empirical validation.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper provides detailed descriptions of datasets (CelebA-HQ), evaluation metrics (PSNR, SSIM, ArcFace similarity, retrieval accuracy, bit accuracy), model components, and training strategy (Sec. 3 and Sec. 4), enabling reproduction of results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: While the implementation details are provided, the code and trained models are not publicly released at submission time due to anonymity constraints. We plan to release them upon acceptance.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies training details, datasets, evaluation protocols, and architectural components. Additional implementation details are provided in the appendix to ensure clarity and reproducibility.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The experiments report deterministic metrics (e.g., PSNR, SSIM, bit accuracy, retrieval accuracy) without error bars. Running multiple trials is computationally expensive due to the training cost of watermarking and generative models.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper reports the hardware configuration (NVIDIA V100 GPUs), training time for two-stage training (semantic quantization and watermarking), and total computational cost (approximately 4 GPU days).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research complies with the NeurIPS Code of Ethics. It focuses on improving robustness and security of watermarking systems and does not involve unethical data usage.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The paper discusses both positive impacts (e.g., deepfake detection, media authentication) and potential risks (e.g., misuse of identity recovery or privacy concerns), along with the importance of responsible deployment.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [No]

Justification: The paper does not release models or datasets at submission time. Potential misuse risks are acknowledged, but no specific safeguards are implemented yet.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Public datasets (e.g., CelebA-HQ) and pretrained models (e.g., ArcFace, StyleGAN2) are properly cited, and their usage follows standard academic practice.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [No]

Justification: The paper does not introduce new datasets. The proposed model will be documented and released after publication.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The work does not involve human subjects or crowdsourcing.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: No human subject research is conducted.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not used as part of the core methodology. They are only used for writing and editing assistance, which does not affect the scientific contributions.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.