Supplementary Material of IDFace: Face Template Protection for Efficient and Secure Identification

Sunpill Kim[1†] Seunghun Paik[1†] Chanwoo Hwang[1] Dongsoo Kim[1] Junbum Shin[2] Jae Hong Seo[1*]
[1]Department of Mathematics & Research Institute for Natural Sciences, Hanyang University
[2]CryptoLab Inc.
{ksp0352, whitesoonguh, aa5568, frds37, jaehongseo}@hanyang.ac.kr, junbum.shin@cryptolab.co.kr

In this supplementary material, we provide some details on both theoretical and empirical analyses that were omitted in the main paper. This document provides the following contents:

Contents

*Corresponding author
†These co-first authors contributed equally to this work

A. Additional Related Works

We briefly survey other types of biometric template protection (BTP) methods than using homomorphic encryption (HE), which were omitted in the main text. We divide the taxonomy of BTPs with respects to whether cryptographic tools are used or not during construction. For BTPs with cryptographic tools, we will focus on fuzzy commitment (FC) [28] constructions, which is considered one of typical solutions for constructing BTPs.

BTPs without Cryptographic Tools. In fact, several BTPs without using cryptographic primitives have been proposed. For example, one famous approach is to employ locality-sensitive hashing (LSH) [21], which is a (non-cryptographic) hash function that makes a collision on the hashed value if two input vectors are sufficiently close enough. Starting from [24], several LSH-based BTPs [7, 23, 26, 35, 38] have been proposed under the assumption that finding a pre-image of a LSH is computationally infeasible. However, their security proofs are less robust than BTPs whose security relies on cryptographic assumptions, as shown in several cryptanalyses [14, 58, 64] that successfully recover a pre-image of the protected template.

On the other hand, there are several attempts to design BTPs through obfuscating the feature templates [25, 49, 50, 65]. More precisely, the goal of these works is to modify the face templates so that the reconstruction attack adversary cannot train and infer the reconstruction model as desired while maintaining the accuracy of the recognition model through these templates. To this end, [65] utilized the idea of adversarial attack on the shadow face reconstruction model owned by the database server to protect face templates. On the other hand, other works attempted to delete and recover visual information of the queried facial images on frequency domain [25, 49] or using an auxiliary neural network [50], in order to disturb the training of reconstruction attack models. Although all these works provided their own security analyses, they are all empirical; there are no theoretical or provable security guarantees.

FC-based BTPs. A FC scheme is a famous solution for constructing BTP using error-correcting code (ECC). In this approach, they view biometric templates as a fuzzy data, coping with the inherent noise of the biometrics via error-correction. Since the hashed values of the noise-corrected templates are stored to the database, the security of FC-based BTPs relies on the onewayness of the cryptographic hash function.

The accuracy degradation accompanied from the FC-based BTP depends on the noise-correction capacity of the underlying ECC. In addition, since many classical ECCs, such as BCH-codes or Reed-Solomon codes. were defined over binary or finite fields, they cannot be directly adopted for correcting noises in the real-valued face templates. For this reason, recent studies elaborated on designing ECCs suitable for face templates. For example, some studies [8, 27, 34, 43, 62] designed neural network-based decoders for classical ECCs. [51] proposed a novel representation of real-valued feature vectors tailored for classical ECCs. On the other hand, [32] proposed a novel ECC defined over the unit hypersphere, proposing a FC-based BTP called IronMask using this ECC.

Many FC-based BTPs were proposed for face verification tasks, showing a reasonable performance with respect to efficiency. However, we note that they are not suitable for large-scale identification because of the difficulty of leveraging parallel computation, such as a batched computation of the matching score.

Privacy-Preserving Face Recognition. Other than BTPs, it is worthwhile to mention that there have been studies focused on information leakage during the communication in the matching process [12, 13, 18, 20, 36]. In their works, they regard each process of enrollment and identification as a protocol between the database server and the client. From this setting, they utilized several cryptographic tools on secure multi-party computation to construct secure two (or multi) party protocols. Nevertheless, we note that protecting templates is not a primary goal of theirs; in fact, the stored face templates in the database are unencrypted in some works [12, 13]. For this reason, we focused on BTP methods only throughout this paper.

B. Full Statement of Proposition 1 and its Proof

In this section, we give our theoretical analysis of the proposed almost isometric transformation. For this, we first give the full statement of Proposition 1 in the main text. Furthermore, we give a mathematical proof on the Proposition 1, which demonstrates that our proposed transformation is indeed an almost isometric transformation.

B.1. Full Statement of Proposition 1

The following proposition is the full statement of the Proposition 1. Note that the result given in the main text was derived by numerically calculating the following integrals when $(d, \alpha) = (512, 341)$. Throughout this section, $o(1)$ denotes a little-oh notation with respect to the dimension d.

Proposition 1. *For $d \in \mathbb{N} \setminus \{1\}$ and $\theta \in (0, \pi/2) \cup (\pi/2, \pi)$,*

$$T_\alpha : \mathbb{S}^{d-1} \to \mathcal{Z}_\alpha^d \text{ is a } (\epsilon_{\alpha,\theta} + o(1), \delta_\alpha, \theta)\text{-isometry for } \epsilon_{\alpha,\theta} = \left|\cos\theta - \frac{2P(\theta)\cdot d}{\alpha}\right|, \delta_\alpha = o(1)$$

where,

$$P(\theta) = \int_c^\infty \int_{\frac{c}{\cos\theta}-u}^\infty f_{UV}(u,v)dvdu - \int_c^\infty \int_{-\infty}^{-\frac{c}{\cos\theta}-u} f_{UV}(u,v)dvdu,$$

$f_{UV}(u,v)$ is a joint pdf of bivariate normal distribution with two independent random variables $U \sim N(0,1)$ and $V \sim N(0, (\tan\theta)^2)$, $c = \sqrt{2}\text{erf}^{-1}(1-\frac{\alpha}{d})$ for error function erf.

Since the closed form of $f_{UV}(u,v)$ is well-known, we can compute $P(\theta)$ via numerical methods implemented in various libraries, *e.g.*, SciPy [63].

B.2. Proof of Proposition 1

We will prove this proposition by (1) replacing the condition of sampling from $\mathbb{S}^{d-1}$ with sampling from $N(0, I_d)$ and (2) explicitly calculating $\langle T_\alpha(\mathbf{x}), T_\alpha(\mathbf{x}')\rangle$ from order statistics. To this end, we provide 6 lemmas, each of which plays a role as follows:

- *Lemma 1* is a somewhat weaker version of *Proposition 1.*, which introduces some additional parameters related to ϵ and δ. This lemma implies *Proposition 1.* with the careful choice of such parameters.
- *Lemma 2, 3* give our analysis of the relation between the uniform distribution on $\mathbb{S}^{d-1}$ and the normal distribution $N(0, I_d)$. Thanks to these lemmas, it is enough to analyze $N(0, I_d)$ during the proof.
- *Lemma 4, 5, 6* give our analysis of the random variable $T_\alpha(\mathbf{X})$ for $\mathbf{X} \sim N(0, I_d)$. Note that our transformation has a strong relationship with order statistics.

A simple schematic diagram of our proof is given in Figure 1.

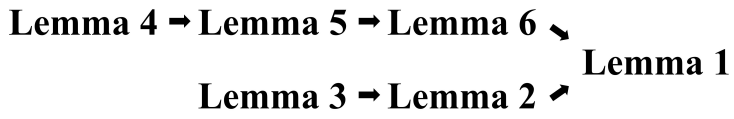


Figure 1. Schematic diagram for our strategy on the proof of *Proposition 1*.

Now, we first introduce our *Lemma 1*.

Lemma 1. *Let $\mathbf{X}, \mathbf{Y}$ be two independent random variables following $N(0, I_d)$. Then for fixed $\theta \in (0, \pi/2) \cup (\pi/2, \pi)$ and the random variable $\mathbf{W} = \frac{\mathbf{X} + \tan\theta \mathbf{Y}}{\sqrt{1+\tan^2\theta}}$, let us define the random variable E as*

$$E = \left|\langle T_\alpha(\mathbf{X}), T_\alpha(\mathbf{W})\rangle - \frac{\langle \mathbf{X}, \mathbf{W}\rangle}{||\mathbf{X}||_2 \cdot ||\mathbf{W}||_2}\right|$$

Then, for $\xi_1 > 0, \xi_2 = d^{-1/2+\epsilon}$ for $\epsilon \in (0, 1/2)$ the following holds:

$$\Pr\left[\left|E - |\cos\theta - \frac{2P(\theta,\xi_2)d}{\alpha}|\right| < \xi_1 + \xi_2 + o(1)\right] > 1 - \delta,$$

where $\delta = d^{-1}(3\xi_1^{-2} + 64\xi_2^{-2})$, and

$$P(\theta,\xi_2) = \int_{c+\xi_2}^\infty \int_{\frac{c+\xi_2}{\cos\theta}-u}^\infty f_{UV}(u,v)dvdu - \int_{c+\xi_2}^\infty \int_{-\infty}^{-\frac{c+\xi_2}{\cos\theta}-u} f_{UV}(u,v)dvdu,$$

$f_{UV}(u,v)$ is a joint pdf of bivariate normal distribution with two independent random variables $U \sim N(0,1)$ and $V \sim N(0, (\tan\theta)^2)$, and $c = \sqrt{2}\text{erf}^{-1}(1-\frac{\alpha}{d})$ for error function erf.

We can derive *Proposition 1* directly from *Lemma 1*. This is because we have that

$$\begin{aligned}1-\delta &< \Pr\left[\left|E-\left|\cos\theta-\frac{2P(\theta,\xi_2)d)}{\alpha}\right|\right|<\xi_1+\xi_2+o(1)\right]\\ &= \Pr\left[E<\epsilon_{\alpha,\theta}+(\xi_1+\xi_2+o(1))\right].\end{aligned}$$

In this case, if we choose $\xi_1=\xi_2=d^{-1/3}$, then $\xi_1+\xi_2=O(d^{-1/3})=o(1)$ and $\delta=O(d^{-1/3})=o(1)$, showing our T_α is $(\epsilon_{\alpha,\theta}+o(1)),o(1),\theta)$-isometry.

Our strategy for proving *Lemma 1* is as follows: We first apply the triangular inequality to the involved term on the R.H.S. of the inequality in order to approximate each term contained in E separately and merge them together:

$$\left|E-\left|\cos\theta-\frac{2P(\theta,\xi_2)d}{\alpha}\right|\right|\leq\left|\frac{\langle\mathbf{X},\mathbf{W}\rangle}{||\mathbf{X}||_2\cdot||\mathbf{W}||_2}-\cos\theta\right|+\left|\langle T_\alpha(\mathbf{X}),T_\alpha(\mathbf{W})\rangle-\frac{2P(\theta,\xi_2)d}{\alpha}\right|. \tag{1}$$

We will bound each term in Eq. (1) with some real numbers ξ_1,ξ_2 respectively. Precisely, if we show that

$$\Pr\left[\left|\frac{\langle\mathbf{X},\mathbf{W}\rangle}{||\mathbf{X}||_2\cdot||\mathbf{W}||_2}-\cos\theta\right|<\xi_1+o(1)\right]>1-\delta_1, \tag{2}$$

$$\Pr\left[\left|\langle T_\alpha(\mathbf{X}),T_\alpha(\mathbf{W})\rangle-\frac{2P(\theta,\xi_2)d}{\alpha}\right|<\xi_2\right]>1-\delta_2 \tag{3}$$

for $\delta_1=O(\xi_1^{-2}d^{-1})$ and $\delta_2=O(\xi_2^{-2}d^{-1})$, then the R.H.S. of Eq. (1) is bounded by $\xi_1+\xi_2+o(1)$ with probability at least $1-(\delta_1+\delta_2)=1-O(d^{-1}(\xi_1^{-2}+\xi_2^{-2}))$. To this end, we first give a *Lemma 2* for dealing with the former term.

Lemma 2. *Let $\mathbf{X},\mathbf{Y}\sim N(0,I_d)$ be two independent random variables, and define random variables $\mathbf{W}=\frac{\mathbf{X}+\tan\theta\mathbf{Y}}{\sqrt{1+\tan^2\theta}}$, $\mathbf{C}=\frac{\langle\mathbf{X},\mathbf{W}\rangle}{||\mathbf{X}||_2\cdot||\mathbf{W}||_2}$. Then for $\xi_1>0$, the following holds:*

$$\Pr\left[|C-\cos\theta|<\xi_1+o(1)\right]>1-3\xi_1^{-2}d^{-1}.$$

Prior to proving **Lemma 2**, we provide a useful fact that gives a relation between the uniform distribution on $\mathbb{S}^{d-1}$ and the normal distribution $N(0,I_d)$.

Fact 1 ([54])**.** *Let $\mathbf{X}\sim N(0,I_d)$ be a random variable. Then the random variable $\mathbf{Y}:=\frac{\mathbf{X}}{||\mathbf{X}||_2}$ follows the uniform distribution on $\mathbb{S}^{d-1}$.*

In addition, we give another lemma showing that two vectors uniformly sampled in $\mathbb{S}^{d-1}$ are "almost" orthogonal.

Lemma 3. *Let $\mathbf{X},\mathbf{Y}\sim N(0,I_d)$ be two independent random variables. For the random variable C defined as $C=\frac{\langle\mathbf{X},\mathbf{Y}\rangle}{||\mathbf{X}||_2\cdot||\mathbf{Y}||_2}$, then $\mathbb{E}[C]=0$ and $\mathrm{Var}[C]=\frac{1}{d}$.*

Proof. We will explicitly calculate $\mathbb{E}[C]$ and $\mathrm{Var}[C]$, as follows. For simplicity, we will denote $\mathbf{X}=(X_1,\ldots,X_d)$, and $\mathbf{Y}=(Y_1,\ldots,Y_d)$. Then for $\tilde{X}_1=\frac{X_1}{\sqrt{\sum_{i=1}^d X_i^2}}$ and $\tilde{Y}_1=\frac{Y_1}{\sqrt{\sum_{i=1}^d Y_i^2}}$,

$$\mathbb{E}[C]=\mathbb{E}\left[\frac{\sum_{i=1}^d X_iY_i}{||\mathbf{X}||_2\cdot||\mathbf{Y}||_2}\right]=d\cdot\mathbb{E}[\tilde{X}_1]\cdot\mathbb{E}[\tilde{Y}_1].$$

By the *Fact 1*, $\tilde{X}_1$ follows the distribution of the 1st component of the vector sampled from $\mathbb{S}^{d-1}$. By the radial symmetry of $\mathbb{S}^{d-1}$, $\mathbb{E}[\tilde{X}_1]=0$ holds, so $\mathbb{E}[C]=0$. We also compute $\mathrm{Var}[C]$ by

$$\begin{aligned}\mathrm{Var}[C] &= d\cdot\mathbb{E}[\tilde{X_1}^2\tilde{Y_1}^2]+d(d-1)\cdot\mathbb{E}[\tilde{X}_1\tilde{X}_2\tilde{Y}_1\tilde{Y}_2]\\ &= d\cdot\mathbb{E}[\tilde{X_1}^2]\cdot\mathbb{E}[\tilde{Y_1}^2]+d(d-1)\cdot\mathbb{E}[\tilde{X}_1\tilde{X}_2]\mathbb{E}[\tilde{Y}_1\tilde{Y}_2].\end{aligned}$$

Note that $\tilde{X_1}^2$ follows beta distribution $\mathrm{Beta}(1/2,(d-1)/2)$ and $\mathbb{E}[\tilde{X_1}^2] = 1/d$. For two random variables U, V such that $U = \frac{X_1+X_2}{\sqrt{2}}, V = \frac{X_1-X_2}{\sqrt{2}}$, we can deduce that

$$\tilde{X_1}\tilde{X_2} = \frac{1}{2}\frac{U^2}{U^2+V^2+\sum_{i=3}^d X_i^2} - \frac{1}{2}\frac{V^2}{U^2+V^2+\sum_{i=3}^d X_i^2}.$$

Since all of both term on L.H.S. follows $\mathrm{Beta}(1/2,(d-1)/2)$, we finally obtain $\mathbb{E}[\tilde{X_1}\tilde{X_2}] = 0$, and $\mathrm{Var}[C] = 1/d$, as claimed. □

By using the same technique as *Lemma 3*, it seems that *Lemma 2* can be easily proved. However, we need to consider the dependence of two random variables $\mathbf{X}$ and $\mathbf{W}$. To deal with such dependence, we need to introduce an assumption as follows:

Assumption 1. *Let $\mathbf{X}$ and $\mathbf{Y}$ be two independent random variables following $N(0, I_d)$. For fixed θ, let us define the random variable $\mathbf{W} = \frac{X+\tan\theta\mathbf{Y}}{\sqrt{1+\tan^2\theta}}$. Then we have that $\mathbb{E}\left[\frac{\langle \mathbf{X},\mathbf{W}\rangle}{||\mathbf{X}||_2\cdot||\mathbf{W}||_2}\right] = \cos\theta + o(1)$.*

For justification, we will experimentally verify this assumption in the next section. By admitting that the *Assumption 1* holds, we can prove the *Lemma 2* as follows.

Proof of Lemma 2. By the same strategy as *Lemma 3*, we will calculate $\mathbb{E}[C]$ and $\mathrm{Var}[C]$ directly. From the *Assumption 1*, $\mathbb{E}[C]$ is approximated as $\cos\theta$ with an error term $o(1)$. Therefore, it suffices to compute the reasonable bound of $\mathrm{Var}[C]$. By using the same technique and notation as *Lemma 3*,

$$\begin{aligned}\mathrm{Var}[C] &= \mathbb{E}[C^2] - \mathbb{E}[C]^2 \\ &= d\mathbb{E}[\tilde{X_1}^2\tilde{W_1}^2] + d(d-1)\mathbb{E}[\tilde{X_1}\tilde{W_1}]\mathbb{E}[\tilde{X_2}\tilde{W_2}] - d^2\mathbb{E}[\tilde{X_1}\tilde{W_1}]^2 \\ &= d(\mathbb{E}[\tilde{X_1}^2]\mathbb{E}[\tilde{W_1}^2] + \mathrm{Cov}(\tilde{X_1}^2,\tilde{W_1}^2) - \mathbb{E}[\tilde{X_1}\tilde{W_1}]^2),\end{aligned}$$

Since $\mathrm{Cov}(\tilde{X_1}^2,\tilde{W_1}^2) \le \sqrt{\mathrm{Var}[\tilde{X_1}^2]\cdot\mathrm{Var}[\tilde{W_1}^2]}$, along with the fact that $\tilde{X_1}^2$ follows beta distribution $\mathrm{Beta}(1/2,(d-1)/2)$, we obtain

$$\mathrm{Var}[C] \le d(\mathbb{E}[\tilde{X_1}^2]^2 + \mathrm{Var}[\tilde{X_1}] - \mathbb{E}[\tilde{X_1}\tilde{W_1}]^2) \le d\left(\frac{1}{d^2} + \frac{2(d-1)}{d^2(d+1)}\right) \le 3d^{-1}.$$

By Chebyshev's inequality, for $\xi_1 > 0$, $\xi_E = \mathbb{E}[C] - \cos\theta$,

$$\Pr[|C - \cos\theta| < \xi_1 + \xi_E] > 1 - \mathrm{Var}[C]\xi_1^{-2} > 1 - 3\xi_1^{-2}d^{-1}.$$

Since $\xi_E = o(1)$ by *Assumption 1*, we have the desired result. □

On the other hand, we need to handle the latter term of Eq. (1). Our observation is the following: let $\mathbf{x},\mathbf{y} \in \mathbb{S}^{d-1} \subset \mathbb{R}^d$. Then, when we denote $(I_{\mathbf{x}}^+, I_{\mathbf{x}}^-)$, $(I_{\mathbf{y}}^+, I_{\mathbf{y}}^-)$ as the sets of indices corresponding components of $T_\alpha(\mathbf{x}), T_\alpha(\mathbf{y})$ are $\frac{+1}{\sqrt{\alpha}}, \frac{-1}{\sqrt{\alpha}}$, respectively, we have that

$$\alpha\cdot\langle T_\alpha(\mathbf{x}), T_\alpha(\mathbf{y})\rangle = (|I_{\mathbf{x}}^+ \cap I_{\mathbf{y}}^+| + |I_{\mathbf{x}}^- \cap I_{\mathbf{y}}^-|) - (|I_{\mathbf{x}}^+ \cap I_{\mathbf{y}}^-| + |I_{\mathbf{x}}^- \cap I_{\mathbf{y}}^+|). \tag{4}$$

We focus on each term in Eq. (4). For two random variables $\mathbf{X} = (X_1,\dots,X_d)$ and $\mathbf{Y} = (Y_1,\dots,Y_d)$, we introduce new random variables $I_{\mathbf{XY}}^{\mathrm{pp}}, I_{\mathbf{XY}}^{\mathrm{mm}}, I_{\mathbf{XY}}^{\mathrm{mp}}, I_{\mathbf{XY}}^{\mathrm{pm}}$ such that

$$I_{\mathbf{XY}}^{\mathrm{pp}} = |I_{\mathbf{X}}^+ \cap I_{\mathbf{Y}}^+|, \quad I_{\mathbf{XY}}^{\mathrm{mm}} = |I_{\mathbf{X}}^- \cap I_{\mathbf{Y}}^-|, \quad I_{\mathbf{XY}}^{\mathrm{pm}} = |I_{\mathbf{X}}^+ \cap I_{\mathbf{Y}}^-|, \quad I_{\mathbf{XY}}^{\mathrm{mp}} = |I_{\mathbf{X}}^- \cap I_{\mathbf{Y}}^+|. \tag{5}$$

To analysis the non-zero components after applying T_α, we need to consider the $(d-\alpha)$th order statistics $\tilde{X}_{(d-\alpha)}$ of $|X_1|,\dots,|X_d|$. By using this notation, we can derive that for fixed $i \in [d]$,

$$\Pr[i \in I_{\mathbf{X}}^+ \cap I_{\mathbf{Y}}^+] = \Pr[(|X_i| > \tilde{X}_{(d-\alpha)}) \wedge (|Y_i| > \tilde{Y}_{(d-\alpha)}) \wedge (X_i > 0) \wedge (Y_i > 0)]. \tag{6}$$

To compute this probability, we first need to study the property of order statistics. In fact, there is a lemma from [53], which tells us the asymptotic behavior of order statistics.

Lemma 4 ([53]). *Let $X_1, \ldots, X_d$ be i.i.d. random variables following the distribution D and let us denote $X_{(d-\alpha)}$ as an αth order statistics of them. Then for $\alpha = Cd \in [d]$ for constant $C \in (0,1)$, $X_{(d-\alpha)}$ asymptotically follows a normal distribution, whose mean and variance are $\mathbb{E}[X_{(d-\alpha)}] = F_X^{-1}(1-\alpha/d)$ and $\mathrm{Var}[X_{(d-\alpha)}] = \frac{\alpha(d-\alpha)}{d^2(f_X(F_X^{-1}(1-\alpha/d)))^2}$, where f_X, F_X is the pdf, cdf of the distribution D, respectively.*

In fact, we used $\alpha = \lfloor \frac{2d}{3} \rfloor$ for our transformation, so this lemma is applicable to our setting. From this lemma, we can obtain a reasonable threshold to estimate the output of the transform. More precisely, we can estimate the probability in Eq. (7) for a fixed i as the following lemma.

$$\Pr[i \in I_{\mathbf{X}}^{+}] = \Pr[(|X_i| > \tilde{X}_{(d-\alpha)}) \wedge (X_i > 0)]. \tag{7}$$

Lemma 5. *Let $X_1, \ldots, X_d$ be i.i.d. random variables following the distribution D, and let us denote $X_{(\alpha)}$ as αth order statistics of $X_1, \ldots, X_d$. Then we have that for $i \in [d]$, $\alpha = Cd \in [d]$ for constant $C \in (0,1)$ and $\xi_2 > 0$,*

$$\left| \Pr\left[X_i > X_{(d-\alpha)}\right] - \Pr\left[X_i > F^{-1}\left(1 - \frac{\alpha}{d}\right) + \xi_2\right] \right| < \max\left\{\xi_2, \frac{1}{8\xi_2^2 d(f(F^{-1}(1-\alpha/d)))^2}\right\},$$

where f, F is the pdf, cdf corresponding to D, respectively.

Proof. Let us denote $\mathcal{E}_1$, $\mathcal{E}_2$, and $\mathcal{E}_3$ as events corresponding to $X_i > X_{(d-\alpha)}$, $X_i > F^{-1}(1-\alpha/d) + \xi_2$, and $X_{(d-\alpha)} > F^{-1}(1-\alpha/d) + \xi_2$, respectively. In this setting, we can deduce that if $\mathcal{E}_2$ holds but $\mathcal{E}_1$ does not, then $\mathcal{E}_3$ holds automatically. This gives $\Pr[\neg\mathcal{E}_1 \wedge \mathcal{E}_2] \leq \Pr[\mathcal{E}_3]$. We will focus on calculating $\Pr[\mathcal{E}_3]$. According to *Lemma 4* and Chebyshev's inequality, we have that

$$\Pr\left[\left|X_{(d-\alpha)} - F^{-1}\left(1 - \frac{\alpha}{d}\right)\right| > \xi_2\right] < \frac{\xi_2^{-2} d^{-1}}{4(f(F^{-1}(1-\alpha/d)))^2}.$$

Here, we used the fact that $\frac{\alpha}{d}(1-\frac{\alpha}{d}) \leq \frac{1}{4}$ because $0 < \frac{\alpha}{d} < 1$. Since $X_{(d-\alpha)}$ asymptotically follows normal distribution, we can expect that $X_{(d-\alpha)}$ has a symmetry on the line $x = F^{-1}(1-\alpha/d)$. Therefore, we have

$$\begin{aligned} \Pr[\mathcal{E}_1] - \Pr[\mathcal{E}_2] &= \Pr[\mathcal{E}_1 \wedge \neg\mathcal{E}_2] - \Pr[\neg\mathcal{E}_1 \wedge \mathcal{E}_2] \\ &\geq -\Pr[\neg\mathcal{E}_1 \wedge \mathcal{E}_2] \geq -\frac{1}{8\xi_2^2 d} \cdot \frac{1}{(f(F^{-1}(1-\alpha/d)))^2}. \end{aligned} \tag{8}$$

Also, if we calculate $\Pr[\mathcal{E}_1]$ and $\Pr[\mathcal{E}_2]$ explicitly, we have that $\Pr[\mathcal{E}_1] = \alpha/d$, and

$$\Pr[\mathcal{E}_2] = \int_{\tau}^{\infty} f(x)dx = \lim_{t\to\infty} F(t) - F(\tau) = 1 - F(\tau),$$

where $\tau = F^{-1}(1-\alpha/d) + \xi_2$. If we consider first order Taylor approximation of F on τ, we obtain

$$\Pr[\mathcal{E}_2] = \alpha/d - \xi_2 f(F^{-1}(1-\alpha/d)) - O(\xi_2^2)$$

Therefore, we have that $\Pr[\mathcal{E}_1] - \Pr[\mathcal{E}_2] < \xi_2$. By combining this with Eq. (8), we finally obtain the following inequality:

$$|\Pr[\mathcal{E}_1] - \Pr[\mathcal{E}_2]| < \max\left\{\xi_2, \frac{1}{8\xi_2^2 d(f(F^{-1}(1-\alpha/d)))^2}\right\}. \tag{9}$$

This completes the proof. □

If we set ξ_2 such that $\xi_2 = d^{-1/2+\epsilon}$ for some $\epsilon \in (0, 1/2)$, then as $d \to \infty$, the R.H.S of the Eq. (9) tends to 0. That is, the probability calculated from the threshold ($\Pr[\mathcal{E}_2]$) is a good approximation for the probability of whether the given entry surpasses the $(d-\alpha)$th order statistics or not ($\Pr[\mathcal{E}_1]$). We can apply the same argument on estimating Eq. (6), as the following lemma. We will denote the upper bound on the probability obtained in *Lemma 5* as $\mathcal{U}_{\alpha,\xi_2}$.

Lemma 6. *Let* $\mathbf{X}, \mathbf{Y}$ *be two independent random variables such that* $\mathbf{X} \sim N(0, I_d)$ *and* $\mathbf{Y} \sim N(0, (\tan^2\theta) I_d)$*. Then for the random variable* $\mathbf{W} = \frac{X+Y}{\sqrt{1+\tan^2\theta}}$*,* $\alpha = Cd \in [d]$ *for constant* $C \in (0,1)$*, fixed* $i \in [d]$ *and* $\xi_2 \in (0, C)$*, define*

$$P^+(\theta, \xi_2) = \int_{c+\xi_2}^{\infty} \int_{\frac{c+\xi_2}{\cos\theta} - u}^{\infty} f_{UV}(u,v) dv du, \quad \hat{P}^+(\theta) = \Pr\left[i \in I_{\mathbf{X}}^+ \cap I_{\mathbf{W}}^+\right],$$

where f_{UV} *is the joint pdf of two independent random variables* $U \sim N(0,1)$ *and* $V \sim N(0, \tan^2\theta)$*, and* $c = \sqrt{2}\mathrm{erf}^{-1}(1 - \frac{\alpha}{d})$*. Then we have that*

$$|P^+(\theta, \xi) - \hat{P}^+(\theta)| < \left(1 + \frac{2C - \xi_2 + 1}{C^2 - C\xi_2}\right) \mathcal{U}_{\alpha, \xi_2},$$

for pdf of the random variable $|X|$ *for* $X \sim N(0,1)$ *calculated as*

$$f_{|X|}(x) = \begin{cases} \frac{2}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}, & \text{if } x \geq 0 \\ 0, & \text{Otherwise} \end{cases}$$

Proof. Let us denote $\mathbf{X} = (X_1, \ldots, X_d)$ and $\mathbf{W} = (W_1, \ldots, W_d)$, and let $\tilde{X}_{(\alpha)}$ and $\tilde{W}_{(\alpha)}$ be the order statistics of $|X_1|, \ldots, |X_d|$ and $|W_1|, \ldots, |W_d|$, respectively. Also, we define events $\hat{\mathcal{E}_1}, \hat{\mathcal{E}_2}, \mathcal{E}_1$ and $\mathcal{E}_2$ as

$$(X_i > \tilde{X}_{(d-\alpha)}),\ (W_i > \tilde{W}_{(d-\alpha)}), (X_i > c + \xi_2),\ (W_i > c + \xi_2),$$

respectively, where $c = \sqrt{2}\mathrm{erf}^{-1}(1 - \alpha/d)$. Then $\hat{P}^+(\theta) = \Pr[\hat{\mathcal{E}_1} \wedge \hat{\mathcal{E}_2}]$, and for joint pdf $f_{UV}(u,v)$ of two independent random variables $U \sim N(0,1)$ and $V \sim N(0, \tan^2\theta)$,

$$\Pr[\mathcal{E}_1 \wedge \mathcal{E}_2] = \Pr[(X_i > c + \xi_2) \wedge (W_i > c + \xi_2)] = \int_{c+\xi_2}^{\infty} \int_{\frac{c+\xi_2}{\cos\theta} - u}^{\infty} f_{UV}(u,v) dv du = P^+(\theta, \xi_2).$$

In fact, c corresponds to the $F^{-1}(1 - \alpha/d)$ term on *Lemma 4*. Note that the cdf $F_{|X|}$ of $|X|$ is

$$F_{|X|}(x) = \begin{cases} \frac{2}{\sqrt{2\pi}} \int_0^x e^{-\frac{t^2}{2}} dt, & \text{if } x \geq 0 \\ 0, & \text{Otherwise} \end{cases},$$

and can be expressed by the error function defined as $\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$.

Now, to estimate the probability $|P^+(\theta, \xi_2) - \hat{P}^+(\theta)|$, we first derive that

$$\left|P^+(\theta, \xi_2) - \hat{P}^+(\theta)\right| = \left|\Pr[\hat{\mathcal{E}_1} \wedge \hat{\mathcal{E}_2}] - \Pr[\mathcal{E}_1 \wedge \mathcal{E}_2]\right| \leq \left|\Pr[\hat{\mathcal{E}_2}|\hat{\mathcal{E}_1}] - \Pr[\mathcal{E}_2|\mathcal{E}_1]\right| + \left|\Pr[\mathcal{E}_1] - \Pr[\hat{\mathcal{E}_1}]\right|.$$

For the former term, we can obtain

$$\begin{aligned}
&\left|\Pr[\hat{\mathcal{E}_2}|\hat{\mathcal{E}_1}] - \Pr[\mathcal{E}_2|\mathcal{E}_1]\right| \leq \left|\Pr[\hat{\mathcal{E}_2}|\hat{\mathcal{E}_1}] - \Pr[\mathcal{E}_2|\hat{\mathcal{E}_1}]\right| + \left|\Pr[\mathcal{E}_2|\hat{\mathcal{E}_1}] - \Pr[\mathcal{E}_2|\mathcal{E}_1]\right| \\
&\leq \frac{\left|\Pr[\hat{\mathcal{E}_2} \wedge \hat{\mathcal{E}_1}] - \Pr[\mathcal{E}_2 \wedge \hat{\mathcal{E}_1}]\right|}{\Pr[\hat{\mathcal{E}_1}]} + \left|\frac{\Pr[\mathcal{E}_2 \wedge \hat{\mathcal{E}_1}]}{\Pr[\hat{\mathcal{E}_1}]} - \frac{\Pr[\mathcal{E}_2 \wedge \hat{\mathcal{E}_1}]}{\Pr[\mathcal{E}_1]}\right| + \left|\frac{\Pr[\mathcal{E}_2 \wedge \hat{\mathcal{E}_1}]}{\Pr[\mathcal{E}_1]} - \frac{\Pr[\mathcal{E}_2 \wedge \mathcal{E}_1]}{\Pr[\mathcal{E}_1]}\right| \\
&\leq \frac{\left|\Pr[\hat{\mathcal{E}_2}] - \Pr[\mathcal{E}_2]\right|}{\Pr[\hat{\mathcal{E}_1}]} + \left|\Pr[\mathcal{E}_2 \wedge \hat{\mathcal{E}_1}]\right| \left|\frac{\Pr[\mathcal{E}_1] - \Pr[\hat{\mathcal{E}_1}]}{\Pr[\mathcal{E}_1] \cdot \Pr[\hat{\mathcal{E}_1}]}\right| + \frac{\left|\Pr[\mathcal{E}_1] - \Pr[\hat{\mathcal{E}_1}]\right|}{\Pr[\mathcal{E}_1]} \\
&= \left(\frac{1}{\Pr[\mathcal{E}_1]} + \frac{1}{\Pr[\hat{\mathcal{E}_1}]} + \frac{\Pr[\mathcal{E}_2 \wedge \hat{\mathcal{E}_1}]}{\Pr[\mathcal{E}_1] \cdot \Pr[\hat{\mathcal{E}_1}]}\right) \left|\Pr[\mathcal{E}_1] - \Pr[\hat{\mathcal{E}_1}]\right|.
\end{aligned}$$

Since $\Pr[\hat{\mathcal{E}}_1] = \alpha/d$ and $\Pr[\mathcal{E}_1] > \alpha/d - \xi_2$, we have that for the constant $C = \alpha/d$,

$$\begin{aligned}\left|P^+(\theta,\xi_2) - \hat{P}^+(\theta)\right| &\le \left(1 + \frac{1}{\Pr[\mathcal{E}_1]} + \frac{1}{\Pr[\hat{\mathcal{E}}_1]} + \frac{\Pr[\mathcal{E}_2 \wedge \hat{\mathcal{E}}_1]}{\Pr[\mathcal{E}_1]\cdot\Pr[\hat{\mathcal{E}}_1]}\right)\cdot\left|\Pr[\mathcal{E}_1] - \Pr[\hat{\mathcal{E}}_1]\right| \\ &\le \left(1 + \frac{1}{C} + \frac{1}{C-\xi_2} + \frac{1}{C(C-\xi_2)}\right)\mathcal{U}_{\alpha,\xi_2} \\ &= \left(1 + \frac{2C - \xi_2 + 1}{C^2 - C\xi_2}\right)\mathcal{U}_{\alpha,\xi_2}. \end{aligned} \tag{10}$$

Therefore, we obtained the desired bound. □

Remark that in Eq. (10), if we assume that $\xi_2 \in (d^{-1}, d^{-1/2})$, then for sufficiently large d, $\xi_2 < C/2$. Since the function $f(x) = \frac{2C+1-x}{C^2-Cx}$ monotonically increases for $x < C$, under the condition of *Lemma 1*, we obtain

$$\left|P^+(\theta,\xi_2) - \hat{P}^+(\theta)\right| \le \frac{(C+1)(C+2)}{C^2}\mathcal{U}_{\alpha,\xi_2}$$

Now we are ready to complete the proof of *Lemma 1*.

Proof of Lemma 1. It suffices to derive Eq. (3). For two independent random variables $\mathbf{X}, \mathbf{Y} \sim N(0, I_d)$ and a fixed $\theta \in [0, \pi/2)$, let $\mathbf{W} = \frac{\mathbf{X} + \tan\theta \mathbf{Y}}{\sqrt{1+\tan^2\theta}}$. From this, we define the random variables $I^{\mathrm{pp}}_{\mathbf{XW}}$, $I^{\mathrm{mm}}_{\mathbf{XW}}$, $I^{\mathrm{pm}}_{\mathbf{XW}}$, $I^{\mathrm{mp}}_{\mathbf{XW}}$ in same way as Eq. (5). We will estimate

$$I^{\mathrm{pp}}_{\mathbf{XW}} + I^{\mathrm{mm}}_{\mathbf{XW}} - I^{\mathrm{pm}}_{\mathbf{XW}} - I^{\mathrm{mp}}_{\mathbf{XW}}$$

by using previous lemmas. Recall that

$$\alpha \cdot \langle T_\alpha(\mathbf{X}), T_\alpha(\mathbf{W})\rangle = (|I^+_{\mathbf{X}} \cap I^+_{\mathbf{W}}| + |I^-_{\mathbf{X}} \cap I^-_{\mathbf{W}}|) - (|I^+_{\mathbf{X}} \cap I^-_{\mathbf{W}}| + |I^-_{\mathbf{X}} \cap I^+_{\mathbf{W}}|),$$

and for each indices $i \in [d]$,

$$\Pr[i \in I^+_{\mathbf{X}} \cap I^+_{\mathbf{W}}] = \Pr[(|X_i| > \tilde{X}_{(d-\alpha)}) \wedge (|W_i| > \tilde{W}_{(d-\alpha)}) \wedge (X_i > 0) \wedge (W_i > 0)].$$

By using the result of *Lemma 6*, we can approximate this probability from

$$\Pr[(X_i > c + \xi_2) \wedge (W_i > c + \xi_2) \wedge (X_i > 0) \wedge (W_i > 0)]$$

within $\mathcal{U}_{\alpha,\xi_2}$ error, where c = $\mathrm{erf}(1 - \alpha/d)$. One may observe that the events in the above probability term are mutually independent with respect to each index. That is, when we define a random variable $\mathbf{Z} \sim N(dP^+(\theta,\xi_2), dP^+(\theta,\xi_2)(1 - P^+(\theta,\xi_2)))$ and $\hat{\mathbf{Z}} \sim N(d\hat{P}^+(\theta), d\hat{P}^+(\theta)(1 - \hat{P}^+(\theta)))$, $I^{pp}_{\mathbf{XW}}$ first can be approximated by $\hat{\mathbf{Z}}$, which again well approximated by $\mathcal{Z}$. To be precise, from the result of *Lemma 6*, with Chebyshev's inequality for $\xi_3 > C^*\mathcal{U}_{\alpha,\xi_2}d$ and $C^* = \frac{(C+1)(C+2))}{C^2}$,

$$\Pr\left[\left|I^{\mathrm{pp}}_{\mathbf{XW}} - P^+(\theta,\xi_2)d\right| < \xi_3\right] \ge 1 - \frac{4}{d\left(\xi_3 - C^*\mathcal{U}_{\alpha,\xi_2}d\right)^2},$$

because $\hat{P}^+(\theta,\xi_2)(1 - \hat{P}^+(\theta,\xi_2)) < \frac{1}{4}$. By the symmetry of the normal distribution, we can do the same approximation to $I^{\mathrm{mm}}_{\mathbf{XW}}$ which results in the same bound. Also, for

$$P^-(\theta,\xi) = \int_{c+\xi}^{\infty}\int_{-\infty}^{-\frac{c+\xi}{\cos\theta} - u} f_{UV}(u,v)dvdu,$$

we can apply a similar argument for $I^{\mathrm{pm}}_{\mathbf{XW}}$ and $I^{\mathrm{mp}}_{\mathbf{XW}}$, namely,

$$\Pr\left[\left|I^{\mathrm{pm}}_{\mathbf{XW}} - P^-(\theta,\xi_2)d\right| < \xi_3\right] > 1 - \frac{4}{d\left(\xi_3 - C^*\mathcal{U}_{\alpha,\xi_2}d\right)^2}.$$

By applying triangular inequality, for

$$P(\theta,\xi) = P^+(\theta,\xi) - P^-(\theta,\xi) = \int_{c+\xi}^{\infty}\int_{\frac{c+\xi}{\cos\theta}-u}^{\infty} f_{UV}(u,v)dvdu - \int_{c+\xi}^{\infty}\int_{-\infty}^{-\frac{c+\xi}{\cos\theta}-u} f_{UV}(u,v)dvdu,$$

we obtain

$$\begin{aligned}&|I^{\mathrm{pp}}_{\mathbf{XW}} + I^{\mathrm{mm}}_{\mathbf{XW}} - I^{\mathrm{pm}}_{\mathbf{XW}} - I^{\mathrm{mp}}_{\mathbf{XW}} - 2P(\theta,\xi_2)d| \\ &\leq |I^{\mathrm{pp}}_{\mathbf{XW}} - P^+(\theta,\xi_2)| + |I^{\mathrm{mm}}_{\mathbf{XW}} - P^+(\theta,\xi_2)| + |I^{\mathrm{pm}}_{\mathbf{XW}} - P^-(\theta,\xi_2)| + |I^{\mathrm{mp}}_{\mathbf{XW}} - P^-(\theta,\xi_2)|. \end{aligned} \tag{11}$$

In this inequality, we can calculate the probability that each term of the R.H.S. in Eq. (11) is bounded by $\xi_3/4$. Therefore, we have that

$$\Pr\left[\left|\langle T_\alpha(\mathbf{X}), T_\alpha(\mathbf{W})\rangle - \frac{2P(\theta,\xi_2)d}{\alpha}\right| < \xi_3\right] > \left(1 - \frac{16}{d(\alpha\xi_3 - 4d\mathcal{U}_{\alpha,\xi_2})^2}\right)^4 \geq 1 - \frac{64}{d(\alpha\xi_3 - 4dC^*\mathcal{U}_{\alpha,\xi_2})^2},$$

where

$$P(\theta,\xi) = P^+(\theta,\xi) - P^-(\theta,\xi) = \int_{c_\xi}^{\infty}\int_{\frac{c+\xi}{\cos\theta}-u}^{\infty} f_{UV}(u,v)dvdu - \int_{c+\xi}^{\infty}\int_{-\infty}^{-\frac{c_\xi}{\cos\theta}-u} f_{UV}(u,v)dvdu.$$

Therefore, by combining the result of *Lemma 2* and taking $\alpha\xi_3 = 4dC^*C^{-1}\mathcal{U}_{\alpha,\xi_2} + \xi_2 > C^*\mathcal{U}_{\alpha,\xi_2}d$, we finally obtain

$$\begin{aligned}&\Pr\left[\left|E - |\cos\theta - \frac{2P(\theta,\xi_2)d}{\alpha}|\right| < \xi_1 + \xi_2 + 4C^*C^{-1}\mathcal{U}_{\alpha,\xi_2} + o(1)\right] \\ &\geq \Pr\left[\left|E - |\cos\theta - \frac{2P(\theta,\xi_2)d}{\alpha}|\right| < \xi_1 + \alpha^{-1}\xi_2 + 4C^*C^{-1}\mathcal{U}_{\alpha,\xi_2} + o(1)\right] > 1-\delta,\end{aligned}$$

where $\delta = d^{-1}(64\xi_2^{-2} + 3\xi_1^{-2})$. Under our condition that $\xi_2 = d^{-1/2+\epsilon}$ for $\epsilon \in (0, 1/2)$,

$$\mathcal{U}_{\alpha,\xi_2} = \max\left\{\xi_2, \frac{\xi_2^{-2}d^{-1}}{8(f(F^{-1}(1-\alpha/d)))^2}\right\},$$

tends to 0, as d becomes ∞. This completes the proof. □

Verification of the Assumption 1. To complete the proof, it suffices to check whether *Assumption 1* holds or not. To this end, we conducted the following experiment: Let us denote $\mathbf{X}$ and $\mathbf{Y}$ as two independent random variables following the standard normal distribution $N(0, I_d)$. For fixed θ, define $\mathbf{W} = \frac{\mathbf{X}+\tan\theta Y}{\sqrt{1+\tan^2\theta}}$. We will check that $C = \mathbb{E}\left[\frac{\langle \mathbf{X},\mathbf{W}\rangle}{||\mathbf{X}||_2\cdot||\mathbf{W}||_2}\right]$ is estimated by $\cos\theta$ with error $o(1)$. For this, we sample 1,000 pairs of vectors $(\mathbf{x},\mathbf{y})$ corresponding to random variables $(\mathbf{X},\mathbf{Y})$ and calculate the sample mean $\overline{C}$ of $\frac{\langle \mathbf{X},\mathbf{W}\rangle}{||\mathbf{X}||_2\cdot||\mathbf{W}||_2}$. In this setting, we compute $|\overline{C} - \cos\theta|$ with various d and θ, and visualize it as Figure 2, showing that as d larger, the error term $|C - \cos\theta|$ tends to 0. This is the evidence showing the *Assumption 1* holds.

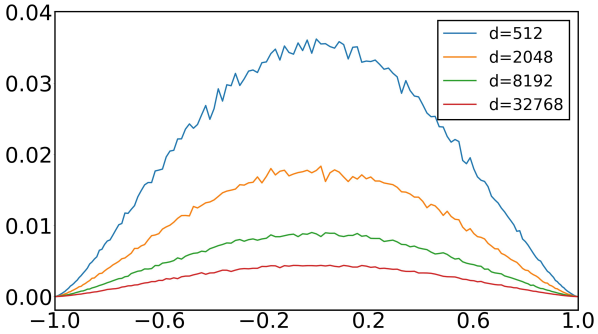


Figure 2. Empirical justification of *Assumption 1*.

C. Analysis on the Parameter Selection

In fact, Proposition 1 does not tell us the explicit value of $\epsilon_{\alpha,\theta}$ and δ_α, or at least an upper bound of them. Since our main interest is the extent to preserve the distance relationship, we provide our experimental results with respect to $\epsilon_{\alpha,\theta}$, along with the desirable choice of α that minimizes $\epsilon_{\alpha,\theta}$ for all $\theta \in [0,\pi]$.

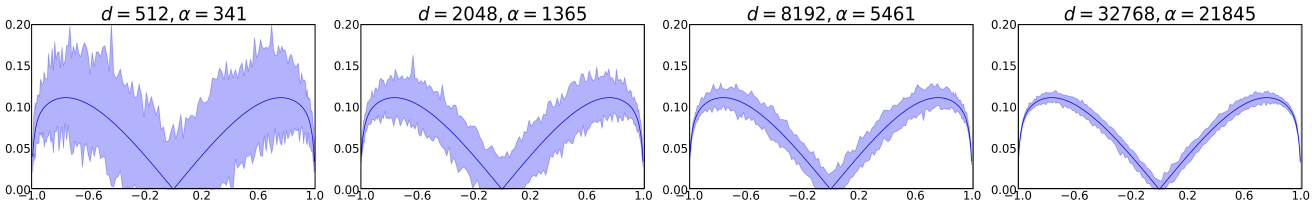


Figure 4. Experimental result on calculating $|\langle T_\alpha(\mathbf{x}), T_\alpha(\mathbf{y})\rangle - \langle \mathbf{x}, \mathbf{y}\rangle|$ and theoretical $\epsilon_{\alpha,\theta}$ with various d and fixed $\alpha = \lfloor \frac{2}{3}d \rfloor$. The x-axis indicates the cosine value between $\mathbf{x}$ and $\mathbf{y}$, and y-axis indicates corresponding range of $|\langle T_\alpha(\mathbf{x}), T_\alpha(\mathbf{y})\rangle - \langle \mathbf{x}, \mathbf{y}\rangle|$. We highlighted the theoretical $\epsilon_{\alpha,\theta}$ value of each angle as a blue solid line.

C.1. Desirable Choice of α

Our first goal is to find the desirable parameter α that minimizes the difference between $\langle \mathbf{x}, \mathbf{y}\rangle$ and $\langle T_\alpha(\mathbf{x}), T_\alpha(\mathbf{y})\rangle$. To this end, one strawman's proposal is to seek a direct minimization of the value $\epsilon_{\alpha,\theta}$ in Proposition 1. Instead of solving such an intricate mimization problem, we take an alternative approach: selecting α that minimizes the deformation of the space of face templates from the transformation T_α. Intuitively, this can be done by maximizing $|\mathcal{Z}_\alpha^d|$, expecting that denser transformation candidates in $\mathbb{S}^{d-1}$ leads a smaller movement of the input face template in terms of cosine similarity. Since $|\mathcal{Z}_\alpha^d| = \binom{d}{\alpha}2^\alpha$, which is a concave function with respect to α, one can easily derive that $|\mathcal{Z}_\alpha^d|$ gets the maximum value when $\alpha = \lfloor \frac{2}{3}d \rfloor$.

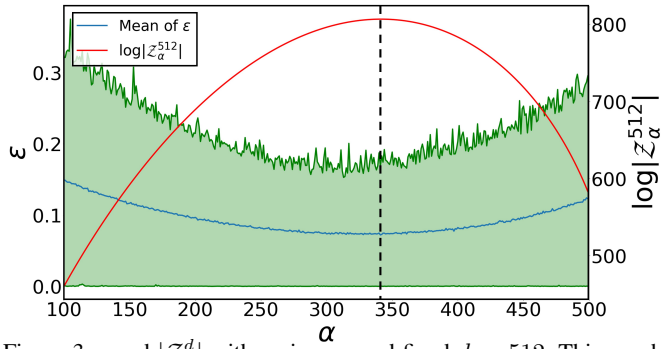


Figure 3. ϵ and $|\mathcal{Z}_\alpha^d|$ with various α and fixed $d = 512$. This graph tells us that ϵ is minimized when $|\mathcal{Z}_\alpha^d|$ is maximized. The dashed line indicates when $\alpha = 341$.

Surprisingly, we observed that such a heuristic approach really pays off. We measured the range and the mean value of $\epsilon = |\langle T_\alpha(\mathbf{x}), T_\alpha(\mathbf{y})\rangle - \langle \mathbf{x}, \mathbf{y}\rangle|$ from randomly sampled 100 pairs of random vectors in $\mathbb{S}^{d-1}$ with various angles and α. Since the dimension d of face templates from recent feature extractors is set to 512, according to our intuition, $\alpha = 341$ should give the smallest ϵ. The visualization of the result is given in Fig. 3. We can figure out despite of the fluctuation in the range of ϵ (green line), the mean value of ϵ (blue line) decreases as the number of codewords (red line) increases. Furthermore, the mean value attains the minimum when $\alpha = 341$, *i.e.*, when $|\mathcal{Z}_\alpha^d|$ is maximized. This justifies our choice of $\alpha = 341$ throughout the parameter selection of IDFace.

C.2. Effect of d and θ on the Error Term.

One may notice that for a fixed θ, the value of $\epsilon_{\alpha,\theta}$ solely depends on the ratio between α and d. On the other hand, as shown in Lemma 1, the error term $o(1)$ added to the $\epsilon_{\alpha,\theta}$ is about the dimension d. Since we already derived that $\alpha = \lfloor \frac{2}{3}d \rfloor$, or $\frac{\alpha}{d} = \frac{2}{3}$, gives the smallest $\epsilon_{\alpha,\theta}$ on average with respect to θ, we now turn our attention to analyze the effect of d and θ on the exact $\epsilon_{\alpha,\theta}$ and the error term $o(1)$ when $\alpha = \lfloor \frac{2}{3}d \rfloor$.

To this end, we conducted the following experiment: for each angle $\theta \in [0, \pi]$, we sample 100 pairs of $(\mathbf{x}, \mathbf{y})$ satisfying $\langle \mathbf{x}, \mathbf{y}\rangle = \cos\theta$ and calculated $|\langle T_\alpha(\mathbf{x}), T_\alpha(\mathbf{y})\rangle - \langle \mathbf{x}, \mathbf{y}\rangle|$ for a fixed $\alpha = \lfloor \frac{2}{3}d \rfloor$. In Figure 4, we provide our experiment result for $d = 512, 2048, 8192$ and 32768 with theoretical $\epsilon_{\alpha,\theta}$ calculated from Proposition 1. This tells us that the estimated $\epsilon_{\alpha,\theta}$ fits well into the empirical result $|\langle T_\alpha(\mathbf{x}), T_\alpha(\mathbf{y})\rangle - \langle \mathbf{x}, \mathbf{y}\rangle|$, without depending on the dimension d. As d becomes larger, $|\langle T_\alpha(\mathbf{x}), T_\alpha(\mathbf{y})\rangle - \langle \mathbf{x}, \mathbf{y}\rangle|$ converges to our theoretical $\epsilon_{\alpha,\theta}$, *i.e.*, $o(1)$ term with respect to d decays to 0, as we predicted in Proposition 1. On the effect of θ, we can figure out that $\epsilon_{\alpha,\theta}$ gets the maximum value when $\cos\theta$ is near ± 0.7. We note that in this case $\epsilon_{\alpha,\theta}$ is 0.111.

D. Additional Experiments and Discussions

We now provide additional experimental results omitted in the main text. This section includes (1) benchmark results of IDFace with other various face recognition models, (2) analysis on the effect of the almost-isometric transformation on intra-class compactness and inter-class discrepancy, (3) comparison with the PCA-based dimensionality reduction technique for further optimization, and (4) omitted discussions about the experimental results.

We note that for all experiments conducted in this section, we utilized the `insightface` library [16] as a baseline and simply modified the loss function for implementing other recognition models. For training each recognition model, we used

four NVIDIA V100 GPUs, following the same training parameter settings, including settings for the optimizer, number of total epochs, and (total) batch size, provided by their implementation of ArcFace[1].

D.1. Evaluation Metrics

We provide the precise definitions of evaluation metrics appeared in this paper. First of all, in the face verification scearios where LFW [17], CFP-FP [60], AgeDB [52], and IJB-C [45] verification benchmark datasets are used for evaluation, we used true accept rate (TAR), false accept rate (FAR), and accuracy as evaluation metrics. For a precise description, we assume that the benchmark dataset $(\mathcal{D})$ consists of pairs of images from the same $(\mathcal{D}_p)$ or different $(\mathcal{D}_n)$ identities. In this setting, for a feature extract F and a threshold parameter τ, we define the TAR and FAR at τ as the ratio of accepted image pairs over $\mathcal{D}_p$ and $\mathcal{D}_n$, respectively, each of which is formally described as follows:

$$\mathsf{TAR}(\tau) = \frac{|\{(I_x, I_y) \in \mathcal{D}_p : s_{\cos}(F(I_x), F(I_y)) > \tau\}|}{|\mathcal{D}_p|}, \quad \mathsf{FAR}(\tau) = \frac{|\{(I_x, I_y) \in \mathcal{D}_n : s_{\cos}(F(I_x), F(I_y)) > \tau\}|}{|\mathcal{D}_n|}.$$

Here, $|\cdot|$ denotes the cardinality of the set and $s_{\cos}(\cdot, \cdot)$ denotes the cosine similarity between two vectors of same length.

In addition, the accuracy at the threshold τ is defined as the ratio of correctly decided image pairs, *i.e.*, over the whole dataset. That is,

$$\mathsf{ACC}(\tau) = \frac{|\{(I_x, I_y) \in \mathcal{D}_p : s_{\cos}(F(I_x), F(I_y)) > \tau\}| + |\{(I_x, I_y) \in \mathcal{D}_n : s_{\cos}(F(I_x), F(I_y)) < \tau\}|}{|\mathcal{D}_p| + |\mathcal{D}_n|}.$$

We note that the accuracy is in fact one of the standard evaluation metrics for the LFW, CFP-FP, and AgeDB datasets. This is the reason why we provided the accuracy on the benchmark results.

On the other hand, for the identification scenario in IJB-C dataset [45], we used a different evaluation metric called true positive identification rate (TPIR) and false positive identification rate (FPIR). For evaluating the accuracy of face identification, we consider two sets of facial images called *galleries*, say $\mathcal{G}_1$ and $\mathcal{G}_2$, whose identities are non-overlap each other. In addition, we also consider a larger *mixed* set $\mathcal{G}_{\text{mixed}}$ that contains all identities in $\mathcal{G}_1$ and $\mathcal{G}_2$. We divide $\mathcal{G}_{\text{mixed}}$ to two subsets, $\mathcal{G}^1_{\text{mixed}}$ and $\mathcal{G}^2_{\text{mixed}}$, with respect to the identity. If identities in $\mathcal{G}_1$ are enrolled to the database, the accuracy evaluation were done by measuring (1) the ratio of the images in $\mathcal{G}$ that are well-identified, *i.e.*, the identity corresponindg to the highest similarity score is identical to the queried one, and (2) the ratio of the images in $\mathcal{G}^2_{\text{mixed}}$ that are recognized as beloning to the enrolled identities. Each ratio corresponds to the TPIR and FPIR, respectively.

With the identification threshold τ for determining whether the queried templates belongs to the enrolled identities or not, we formally describe the TPIR and FPIR as follows:

$$\mathsf{TPIR}(\tau) = \frac{|\{(I, id) \in \mathcal{G}^1_{\text{mixed}} : (I^*, id^*) := \arg\max_{(I', id') \in \mathcal{G}_1} \{s_{\cos}(F(I), F(I'))\}; id = id' \wedge s_{\cos}(F(I), F(I')) > \tau\}|}{|\mathcal{G}_1|},$$

$$\mathsf{FPIR}(\tau) = \frac{|\{I \in \mathcal{G}^2_{\text{mixed}} : \max_{(I', id') \in \mathcal{G}_1} \{s_{\cos}(F(I), F(I'))\} > \tau\}|}{|\mathcal{G}^2_{\text{mixed}}|},$$

assuming that the identities in $\mathcal{G}_1$ are enrolled. We can evaluate the TPIR and FPIR for $\mathcal{G}_2$ in the same manner. Throughout this paper, we reported the average value of TPIR for both cases when $\mathcal{G}_1$ and $\mathcal{G}_2$ are enrolled, respectively.

D.2. Statistics of the Benchmark Datasets

We remark that all the benchmark datasets we used, including LFW, CFP-FP, AgeDB, and IJB-C datasets, are publicly available and already frequently utilizd in the literature of face recognition. For reproducibility, we provide the detailed statistics of each dataset in Table 1.

D.3. Result on Various Face Recognition Models

Our theoretical results on Proposition 1 seem independent of the choice of face recognition model because we did not specify it at that moment. However, we note that the definition of $(\epsilon, \delta, \theta)$-isometry assumes the distribution of the unit feature vectors. Since the distribution of face features is quite far from uniform, to be honest, our theoretical analysis may or may not fit with reality.

[1]For more information, we refer to the following code in the `insightface` library: https://github.com/deepinsight/insightface/blob/master/recognition/arcface_torch/configs/ms1mv3_r100.py

Dataset	LFW	CFP-FP	AgeDB	IJB-C (V)
Imgs/IDs	13,233 / 5,749	7,000 / 500	16,488 / 568	138,836 + 11,779 + 10,040 / 3,531
T/F Pairs	3,000 / 3,000	3,500 / 3,500	3,000 / 3,000	19,557 / 15,638,932

Dataset	IJB-C (ID)
Imgs in G1/G2	5,588 / 6,011
IDs in G1/G2	1,772 / 1,759
Mixed IDs / Imgs	3,531 / 127,152

Table 1. The statistics of various benchmark datasets. We note that IJB-C dataset consists of facial images (138,836), videos (11,779), and non-facial images (10,040). For the IJB-C identification dataset, we provide the statistics of each gallery set.

Dataset	Metric (FAR/FPIR)	Plain	IDFace, Parameters: (α, β)					
			(512, 512)	(341, 341)	(127, 127)	(63, 63)	(341, 127)	(341, 63)
LFW	Accuracy	99.83%	99.87%	99.80%	99.78%	99.67%	99.79%	99.72%
	TAR@FAR	99.73%@0.03%	99.77%@0.03%	99.77%@0.07%	99.63%@0.03%	99.57%@0.13%	99.67%@0.02%	99.57%@0.05%
CFP-FP	Accuracy	99.04%	98.57%	98.83%	98.76%	97.66%	98.79%	98.46%
	TAR@FAR	98.40%@0.14%	97.83%@0.63%	98.37%@0.54%	97.89%@0.26%	96.83%@1.51%	98.16%@0.43%	97.77%@0.73%
AGE-DB	Accuracy	98.38%	98.02%	98.35%	98.02%	97.07%	97.98%	97.91%
	TAR@FAR	96.97%@0.20%	96.47%@0.37%	97.13%@0.43%	96.60%@0.57%	95.83%@1.70%	96.53%@0.37%	96.52%@0.67%
IJB-C(V)	TAR (1e-3)	98.02%	97.27%	97.71%	97.34%	95.95%	97.50%	97.12%
IJB-C(ID)	TPIR(1e-2)	95.35%	94.23%	95.04%	94.00%	91.16%	94.59%	93.71%
	TPIR(1e-3)	88.96%	87.80%	89.18%	87.58%	83.85%	87.12%	87.14%
	TPIR(1e-4)	64.71%	70.69%	66.27%	54.71%	52.91%	61.34%	58.96%

(a) ArcFace

Dataset	Metric (FAR/FPIR)	Plain	IDFace, Parameters: (α, β)					
			(512, 512)	(341, 341)	(127, 127)	(63, 63)	(341, 127)	(341, 63)
LFW	Accuracy	99.80%	99.63%	99.75%	99.65%	99.55%	99.63%	99.68%
	TAR@FAR	99.67%@0.03%	99.43%@0.07%	99.63%@0.10%	99.40%@0.10%	99.30%@0.20%	99.43%@0.02%	99.45%@0.05%
CFP-FP	Accuracy	98.91%	98.27%	98.43%	98.37%	97.21%	98.53%	98.21%
	TAR@FAR	98.26%@0.26%	97.34%@0.80%	97.69%@0.60%	97.26%@0.51%	95.31%@0.89%	97.53%@0.41%	97.29%@0.79%
AGE-DB	Accuracy	98.15%	97.77%	98.15%	97.78%	96.37%	97.92%	97.53%
	TAR@FAR	97.03%@0.50%	95.90%@0.27%	97.17%@0.70%	96.43%@0.80%	95.20%@2.07%	96.25%@0.37%	96.12% @0.90%
IJB-C(V)	TAR (1e-3)	97.04%	96.14%	96.66%	96.27%	94.53%	96.50%	95.91%
IJB-C(ID)	TPIR(1e-2)	94.00%	91.94%	93.34%	92.19%	89.01%	92.95%	91.95%
	TPIR(1e-3)	85.63%	85.00%	86.73%	85.32%	81.32%	84.24%	84.49%
	TPIR(1e-4)	58.34%	57.25%	55.13%	51.85%	48.18%	55.63%	64.60%

(b) MagFace

Dataset	Metric FAR(FPIR)	Plain	IDFace, Parameters: (α, β)					
			(512, 512)	(341, 341)	(127, 127)	(63, 63)	(341, 127)	(341, 63)
LFW	Accuracy	99.78%	99.75%	99.78%	99.78%	99.72%	99.81%	99.79%
	TAR@FAR	99.67%@0.00%	99.67%@0.10%	99.73%@0.07%	99.67%@0.03%	99.47%@0.03%	99.70%@0.00%	99.70%@0.03%
CFP-FP	Accuracy	99.06%	98.67%	99.07%	98.70%	97.83%	98.96%	98.63%
	TAR@FAR	98.49%@0.29%	97.89%@0.46%	98.34%@0.14%	98.26%@0.71%	96.29%@0.63%	98.19%@0.23%	97.87%@0.54%
AGE-DB	Accuracy	98.55%	98.12%	98.45%	98.18%	97.32%	98.34%	97.90%
	TAR@FAR	97.80%@0.53%	96.80%@0.40%	97.10%@0.20%	96.97%@0.43%	95.87%@0.43%	97.35%@0.55%	96.65%@0.68%
IJB-C(V)	TAR (1e-3)	97.98%	97.54%	97.83%	97.55%	96.76%	97.68%	97.41%
IJB-C(ID)	TPIR(1e-2)	95.06%	94.08%	94.73%	93.92%	91.16%	94.35%	93.75%
	TPIR(1e-3)	88.40%	85.83%	86.55%	87.42%	86.53%	87.51%	86.06%
	TPIR(1e-4)	52.04%	50.13%	48.43%	53.79%	47.55%	50.75%	55.13%

(c) SphereFace2

Dataset	Metric FAR(FPIR)	Plain	IDFace, Parameters: (α, β)					
			(512, 512)	(341, 341)	(127, 127)	(63, 63)	(341, 127)	(341, 63)
LFW	Accuracy	99.82%	99.80%	99.82%	99.83%	99.65%	99.82%	99.80%
	TAR@FAR	99.67%@0.00%	99.63%@0.00%	99.67%@0.00%	99.67%@0.00%	99.67%@0.17%	99.70%@0.03%	99.67%@0.00%
CFP-FP	Accuracy	99.09%	98.67%	99.01%	98.81%	98.20%	99.03%	98.79%
	TAR@FAR	98.49%@0.17%	97.86%@0.34%	98.26%@0.17%	98.31%@0.66%	97.46%@1.06%	98.41%@0.29%	97.99%@0.36%
AGE-DB	Accuracy	98.45%	98.00%	98.23%	98.25%	97.53%	98.26%	98.10%
	TAR@FAR	97.33%@0.23%	96.90%@0.70%	96.93%@0.17%	97.30%@0.80%	95.83%@0.77%	96.98%@0.32%	97.12%@0.85%
IJB-C(V)	TAR (1e-3)	97.93%	97.31%	97.68%	97.48%	96.36%	97.56%	97.20%
IJB-C(ID)	TPIR(1e-2)	94.99%	93.83%	94.51%	93.76%	91.17%	94.42%	93.60%
	TPIR(1e-3)	89.56%	87.83%	88.45%	87.45%	82.90%	88.54%	87.54%
	TPIR(1e-4)	60.18%	60.45%	59.88%	62.10%	45.43%	62.55%	56.36%

(d) ElasticFace

Figure 5. Various Face Recognition Benchmark results on non-protected template extractor (Plain) and IDFace with various (α, β).

For this reason, we conducted extensive experiments to evaluate the effectiveness of IDFace on other recently proposed face recognition models. We select ElasticFace [5], MagFace [48], SphereFace2 [66], AdaFace [30], and AdaFace-KPRPE [31]. We implemented the first three FR models by following their official source codes ([47], [4], and [40] for

Dataset	Metric FAR(FPIR)	Plain	IDFace, Parameters: (α, β)					
			(512, 512)	(341, 341)	(127, 127)	(63, 63)	(341, 127)	(341, 63)
LFW	Accuracy	99.82%	99.77%	99.78%	99.83%	99.73%	99.80%	99.82%
	TAR@FAR	99.67%@0.03%	99.57%@0.03%	99.63%@0.07%	99.67%@0.00%	99.57%@0.10%	99.65%@0.05%	99.68%@0.03%
CFP-FP	Accuracy	99.24%	98.96%	99.24%	99.09%	98.20%	99.19%	98.99%
	TAR@FAR	98.66%@0.17%	98.14%@0.23%	98.71%@0.23%	98.51%@0.34%	96.94%@0.54%	98.53%@0.16%	98.24%@0.26%
AGE-DB	Accuracy	98.00%	97.67%	97.97%	97.55%	96.48%	97.80%	97.23%
	TAR@FAR	96.73%@0.73%	96.37%@1.03%	96.50%@0.57%	96.23%@1.13%	94.17%@1.20%	96.65%@1.05%	95.62%@1.15%
IJB-C(V)	TAR (1e-3)	98.39%	97.93%	98.27%	97.90%	96.94%	98.14%	97.78%
IJB-C(ID)	TPIR(1e-2)	96.42%	95.15%	95.95%	95.32%	92.87%	95.72%	95.19%
	TPIR(1e-3)	85.81%	85.83%	85.11%	85.40%	86.21%	86.59%	85.96%
	TPIR(1e-4)	65.60%	63.88%	68.38%	62.38%	61.39%	63.45%	64.66%

(a) AdaFace-IR101

Dataset	Metric FAR(FPIR)	Plain	IDFace, Parameters: (α, β)					
			(512, 512)	(341, 341)	(127, 127)	(63, 63)	(341, 127)	(341, 63)
LFW	Accuracy	99.82%	99.70%	99.82%	99.78%	99.70%	99.81%	99.78%
	TAR@FAR	99.63%@0.00%	99.53%@0.13%	99.63%@0.00%	99.60%@0.03%	99.53%@0.13%	99.63%@0.02%	99.60%@0.03%
CFP-FP	Accuracy	99.30%	98.77%	99.11%	99.17%	98.19%	99.11%	98.78%
	TAR@FAR	98.66%@0.06%	97.91%@0.37%	98.51%@0.29%	98.51%@0.17%	97.00%@0.63%	98.37%@0.16%	98.04%@0.49%
AGE-DB	Accuracy	98.10%	97.30%	97.67%	97.65%	96.23%	97.55%	97.32%
	TAR@FAR	97.10%@0.90%	95.67%@1.07%	96.77%@1.43%	96.17%@0.87%	94.97%@2.50%	96.22%@1.12%	95.72%@1.08%
IJB-C(V)	TAR (1e-3)	98.40%	97.92%	98.20%	97.98%	96.67%	98.07%	97.77%
IJB-C(ID)	TPIR(1e-2)	96.46%	95.27%	96.00%	95.35%	92.10%	95.74%	94.80%
	TPIR(1e-3)	90.52%	90.55%	89.79%	87.45%	85.06%	89.54%	86.84%
	TPIR(1e-4)	60.87%	57.59%	60.94%	62.14%	64.17%	65.31%	61.61%

(b) AdaFace-KPRPE

Figure 6. Various Face Recognition Benchmark results on non-protected template extractor (Plain) and IDFace with various (α, β).

ElasticFace, MagFace, and SphereFace2, respectively). Here, we used ResNet100 as a backbone and trained each model with the MS1MV3 dataset. In addition, for AdaFace and AdaFace-KPRPE, we exploited the pre-trained models provided in the CVLFace library [29] that is maintained by the authors of AdaFace. For each model, we used `IR101` and `ViT-KPRPE` architectures trained by the WebFace12M dataset [70], respectively. The benchmark results are given in Figure 5, 6, which indicates that our IDFace can harmonize well with various recognition systems without significant accuracy degradation.

In particular, to visualize the trade-off relationship between TAR and FAR, we also evaluated the receiver operating characteristic (ROC) and detection error trade-off (DET) curves for IJB-C verification and identification benchmarks, respectively. We tested the IDFace with the same feature extractors in Figure 5, 6 on the parameter settings in $\alpha = 341$ and $\beta = 341, 127, 63$. The results are given in Figure 7, 8. For ROC curves in the verification task, one may observe that a slight accuracy degradation when IDFace is applied. Nevertheless, in terms of area under the curve (AUC) score, the amount of accuracy drop seems insignificant. On the other hand, in the identification task which is our main interest, we can observe that the differences between DET curves from IDFace and the plain model seems negligible. These results indicate that our IDFace can be applied to various face recognition models and various levels of FAR, without significant accuracy degradation.

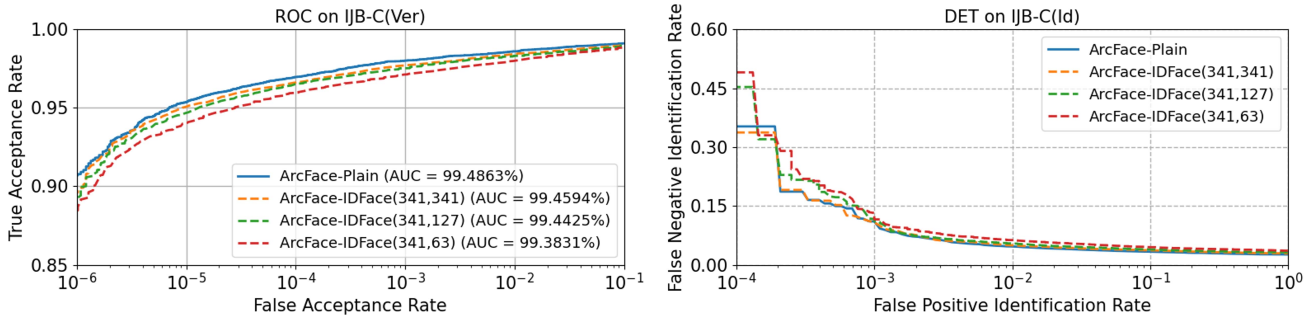
D.4. Effect of our Transformation on Intra and Inter Class Variations

We also analyzed the effect of our transformation in terms of intra-class compactness and inter-class discrepancy on various transformation parameters. To this end, we measured the distribution of the cosine distance between positive and negative pairs on the LFW dataset, respectively. We note that when $\beta < \alpha$, the range of the inner product value between transformed vectors becomes narrower. Concretely, when we choose $(\alpha, \beta) = (341, 63)$, then the range of all possible cosine values is $\left[-\sqrt{\frac{63}{341}}, \sqrt{\frac{63}{341}} \approx 0.4298\right]$.

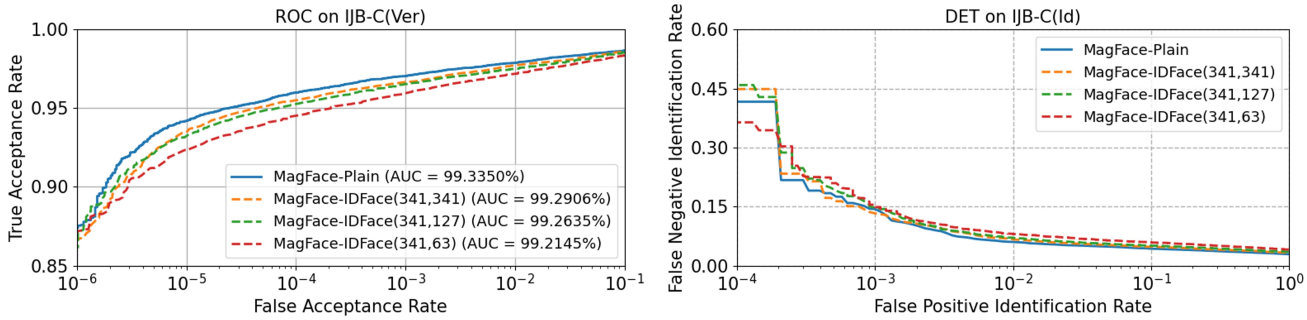
To compensate for such a range mismatch, we multiply an appropriate constant to fit the range of the cosine value into $[-1, 1]$. We note that such re-scaling itself does not affect the accuracy of the recognition system. The result of each distribution after re-scaling is provided in Figure 9. We can figure out that the distribution of negative pairs is more widespread than the result from Plain due to the effect of re-scaling. However, the distributions of cosine values from positive pairs and negative pairs, respectively, are still well distinguished. This result explains the reason for the small accuracy degradation of IDFace even when $\beta < \alpha$.

D.5. Comparison with PCA-based Dimensionality Reduction

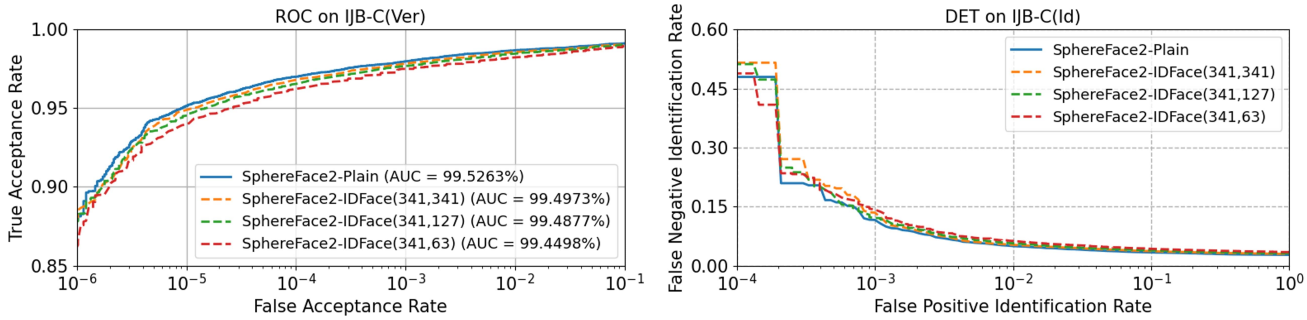
Recall that lots of previous HE-based face template protections [1, 3, 11] employed dimensionality reduction techniques for a good trade-off between efficiency and accuracy. However, as reported in their previous works, the amount of accuracy degra-



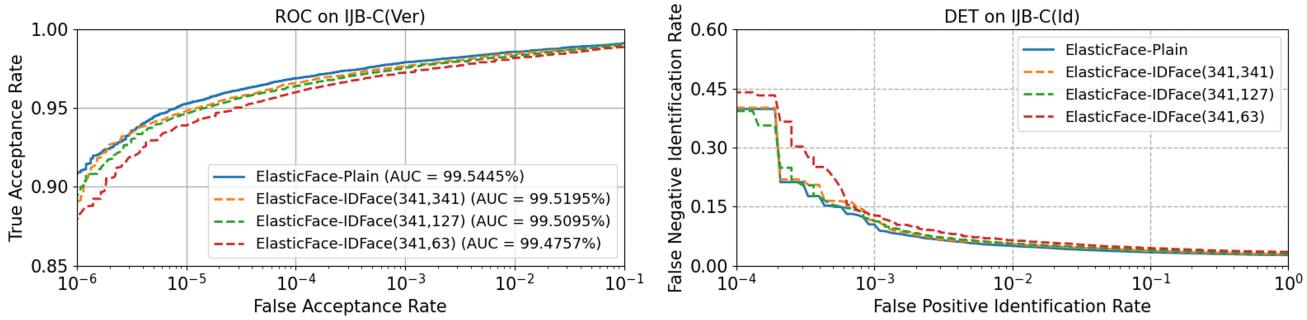
(a) ArcFace



(b) MagFace



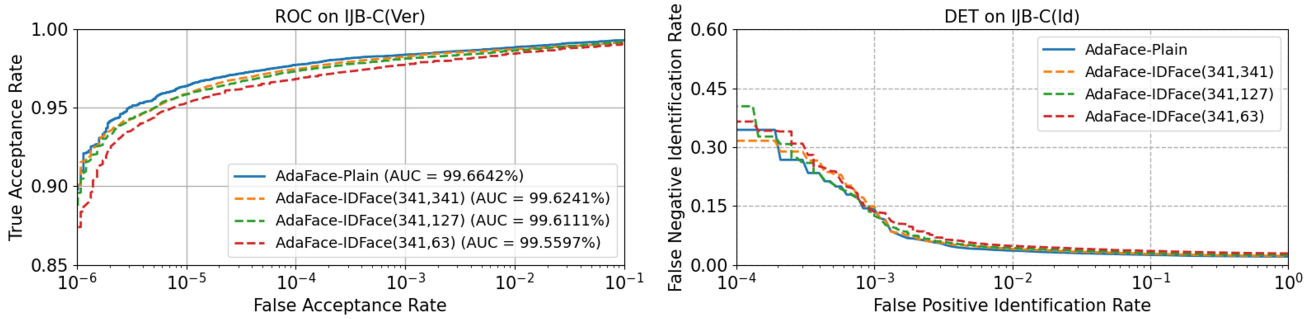
(c) SphereFace2



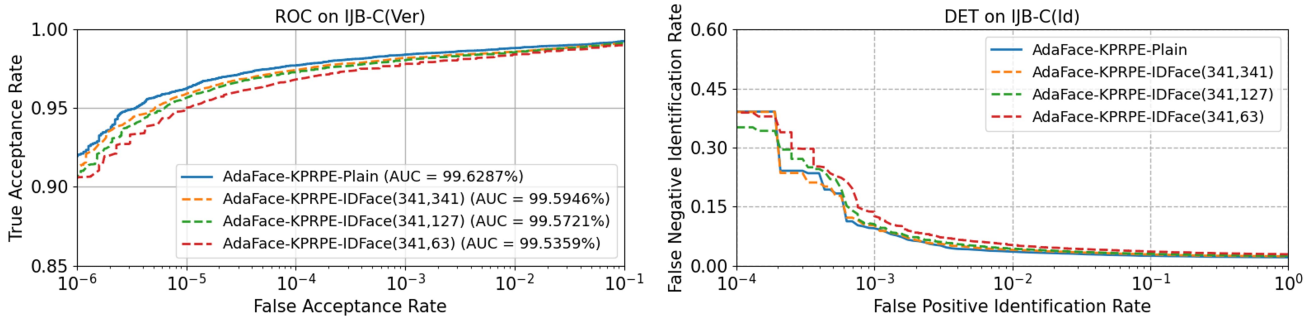
(d) ElasticFace

Figure 7. IJB-C verification/identification benchmark results for non-protected feature extractor (Plain) and IDFace with various (α, β). We also reported AUC-ROC curves and DET curves for each task, respectively.

dation from PCA [1, 3] or neural network-based approach [11] is at most 13.5%, which is quite large compared to that from IDFace. Of course, the amount of degradation depends on the experimental setting, such as the accuracy of an unprotected face recognition model or the dimension of face templates before dimensionality reduction. Hence, for a fair comparison

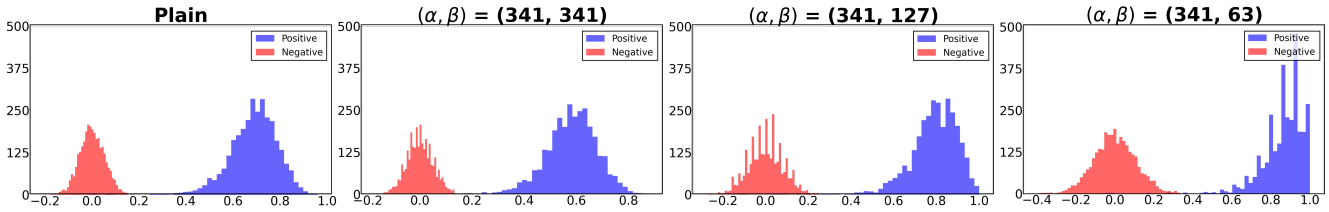


(a) AdaFace-IR101

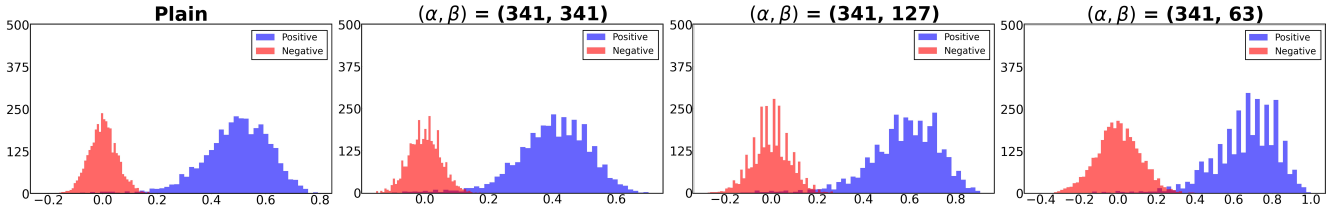


(b) AdaFace-KPRPE

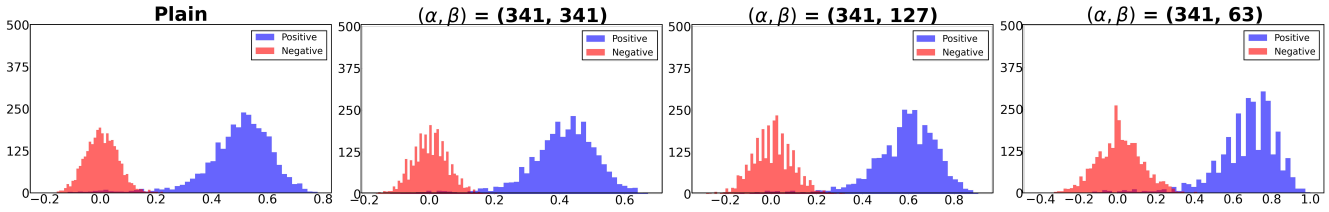
Figure 8. IJB-C verification/identification benchmark results for non-protected feature extractor (Plain) and IDFace with various (α, β). We also reported AUC-ROC curves and DET curves for each task, respectively.



(a) LFW



(b) CFP-FP



(c) AgeDB

Figure 9. The distribution of the rescaled cosine value from each positive, negative pairs on LFW, CFP-FP, AgeDB benchmark datasets. We applied our proposed transform with parameter α, β, by varying $\beta = 63, 127, 341$ while α is fixed to 341. We used ArcFace as a feature extractor.

between IDFace and previous works in terms of accuracy degradation, we provide the result of accuracy degradation when PCA is applied to the templates extracted by the recognition model in our experimental setting.

For training the PCA matrix, we exploited the same dataset (MS1MV3) that was used for training ArcFace. We selected the output dimensions as 32, 64, and 128 for comparison with the previous methods in Table 1 in main text. After training the PCA matrix, we evaluated the accuracy by running benchmarks as if the PCA matrix were added to the final layer of the feature extractor. The result is given in Table 2, which tells us that the accuracy degradation from PCA is around 0.17%–4.67%, 0.67%–11.30%, and 0.52%–11.54% for 128D, 64D and 32D, respectively. We note that even in the fastest parameter setting in IDFace ($\beta = 63$), the accuracy loss from IDFace is smaller than that from PCA with almost all benchmarks we tested. This result indicates that (1) the proposed IDFace is even better than previous methods with PCA of 128D in terms of not only efficiency but also accuracy, and (2) applying PCA to our IDFace would show a larger accuracy degradation than this result.

Dataset	Metric (FAR)	Plain	128D	64D	32D
LFW	Accuracy	99.83%	99.66%	99.16%	99.31%
CFP-FP		99.04%	98.02%	96.15%	96.11%
AgeDB		98.38%	97.46%	95.90%	95.71%
IJB-C(V)	TAR (1e-3)	98.01%	96.43%	93.29%	93.27%
IJB-C(ID)	TPIR (1e-0)	97.38%	95.77%	91.75%	91.72%
	TPIR (1e-1)	96.68%	94.83%	89.32%	89.27%
	TPIR (1e-2)	95.35%	93.40%	85.61%	85.48%
	TPIR (1e-3)	88.96%	86.45%	78.66%	79.23%
	TPIR (1e-4)	64.71%	60.04%	53.41%	53.17%

Table 2. Benchmark results of ArcFace on evaluation datasets with PCA on various dimensions.

D.6. Discussion on the Accuracy Drop

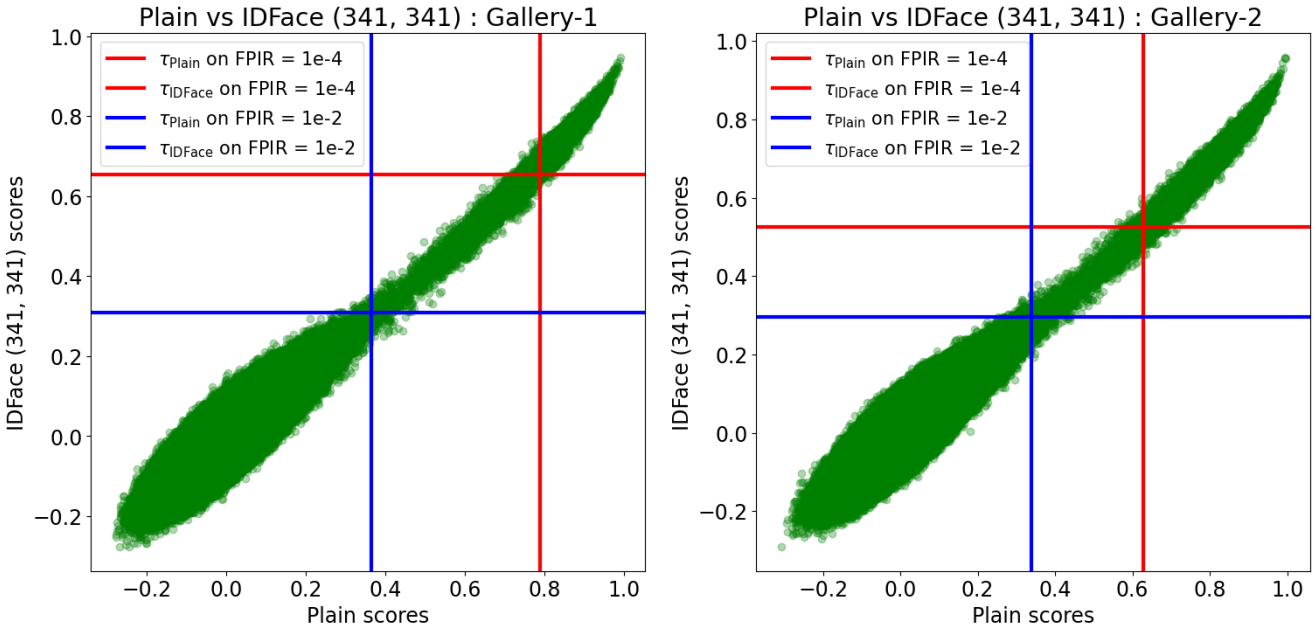
Through experiments, we showed that the tendency of accuracy degradation with respect to the transformation parameters, giving intuition that the reason for accuracy degradation is strongly tied to the extent how the transformation preserves the distance relationship. However, we found that such an interpretation is still insufficient for fully explaining the accuracy degradation of IDFace. For example, we can observe an odd phenomenon that was commonly observed regardless of the choice of the feature extractor: the accuracy increases after employing the IDFace. To clarify these aspects, we further investigate how IDFace, especially our almost isometric transformation, yields accuracy degradation.

On the Distance Preservation Property With Respect to Inputs. According to our analyses, the change in inner product value from ours is at most 0.111 under the suggested parameter. However, the upper bound 0.111 is too high to explain the small accuracy drop of IDFace, so more analysis for our transformation is necessary.

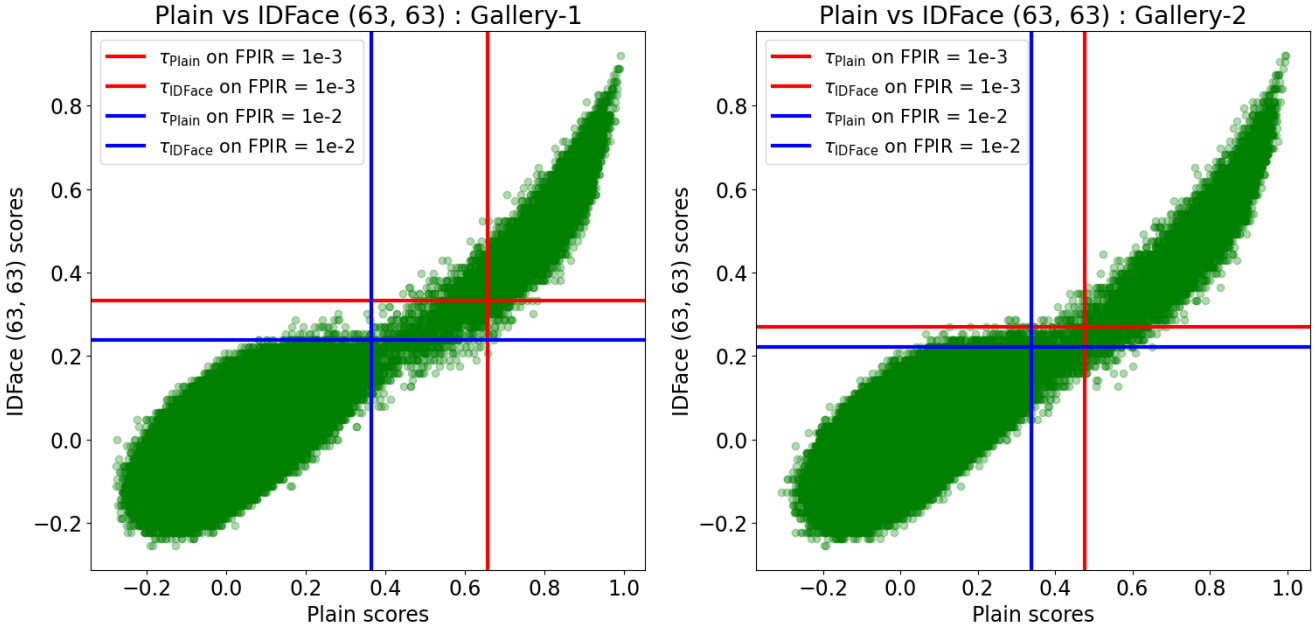
In fact, the bound 0.111 is about the worst case, and the change of the inner product value is also affected by that value before transformation. As we can observe in Figure 4, the difference in the inner product value becomes the largest when the inner product before the transformation is near 0.7 and becomes narrower when the inner product value of inputs tends to 0. We note that the best threshold for each recognition model we evaluated is near 0.15–0.25 for all datasets. This implies that the difference in inner product value near the threshold is relatively small (less than 0.05) compared to the upper bound we derived. We can expect that the inner product values nearby the threshold are relatively less changed than the upper bound, where pairs of faces whose inner product value is nearby here are subject to changing the decision of the model.

It is also worthwhile to mention that the distribution of feature vectors is quite different from uniform, which is in fact a gap between theory and reality. More precisely, in the definition of $(\epsilon, \delta, \theta)$-isometry, we calculated ϵ by considering a uniformly sampled unit vector x and another unit vector y uniformly sampled from vectors whose angles from x are θ. Along with the ease of proof, we intended to construct a data-agnostic distance-preserving transformation via this definition; we note that if some part of the BTP relies on the specific distribution derived from the input dataset, then this BTP cannot be deployed in an open-set face recognition scenario. In fact, our experimental results show that (1) the accuracy degradation carried by IDFace is admissible in various face recognition models, and (2) the almost-isometric transformation would not harm both intra-class compactness and inter-class discrepancy. Nevertheless, from a theoretical perspective, there is still a gap that needs to be further investigated. We leave this as an interesting future work.

Investigating an Odd Phenomenon: Observing Changes on the Decision of the FR Model. Although the above-mentioned analysis gives an insight about the effect of the transformation nearby the threshold, still this does not give a direct explanation on the accuracy drop and is insufficient to explain an unexpected and rather odd phenomenon across Tab. 5,6, and Tab. 3 in the main text: IDFace achieved slightly better accuracy than the plain model for some settings, and the amount of accuracy change becomes larger for low FPIR settings, e.g., FPIR-1e-3 or 1e-4. To complement this, we investigated the occurrence of the "bad" events that would change the decision of the FR model after transformation. More precisely, we can consider two cases, (1) the pairs of faces from the same identity are no longer recognized as the same person, or (2) those pairs from



(a) Plain vs. IDFace with $(\alpha, \beta) = (341, 341)$.



(b) Plain vs. IDFace with $(\alpha, \beta) = (63, 63)$.

Figure 10. Score distribution in the IJB-C identification task on AdaFace-IResNet101 [30] with non-protected feature extractor (Plain) and IDFace. Each point represents an image pair, with scores from Plain (x-axis) and IDFace. Vertical and horizontal lines indicate the thresholds for each setting.

different identities become recognized as the same person.

To this end, we tracked the cosine similarity scores from each pair of faces before and after transformation. We considered all the pairs of facial images that appeared during IJB-C identification benchmark procedure for each gallery (G1 and G2). The total numbers of pairs made from each gallery are 34,718,796 and 34,464,087, respectively. We used the AdaFace-IResNet101 [30] as a feature extractor and tested IDFace with parameters $(\alpha, \beta) = (341, 341)$ and $(63, 63)$, where we can

	FPIR	Settings	TPIR(%)	Thres.
G1	1e-2	Plain	97.05	0.3649
		IDFace	96.69	0.3079
	1e-4	Plain	51.68	0.7888
		IDFace	58.13	0.6539
G2	1e-2	Plain	95.78	0.3389
		IDFace	95.22	0.2962
	1e-4	Plain	79.52	0.6282
		IDFace	78.63	0.5249
Avg.	1e-2	Plain	96.41	N/A
		IDFace	95.95	
	1e-4	Plain	65.60	
		IDFace	68.38	

(a) Plain vs. IDFace, $(\alpha, \beta) = (341, 341)$.

	FPIR	Settings	TPIR(%)	Thres.
G1	1e-2	Plain	97.05	0.3649
		IDFace	93.80	0.2381
	1e-3	Plain	80.00	0.6578
		IDFace	84.56	0.3333
G2	1e-2	Plain	95.78	0.3389
		IDFace	91.95	0.2222
	1e-3	Plain	91.63	0.4756
		IDFace	87.86	0.2698
Avg.	1e-2	Plain	96.41	N/A
		IDFace	92.87	
	1e-3	Plain	85.81	
		IDFace	86.21	

(b) Plain vs. IDFace, $(\alpha, \beta) = (63, 63)$.

Table 3. Evaluation results of IJB-C identification benchmark with respect to galleries. We reported results from the same setting as in Fig. 10. G1, G2, and Avg. indicate gallery 1, gallery 2, and average value of TPIR from each gallery, respectively.

observe that both the increase and decrease of accuracy at the same parameter but different levels of FPIRs.

We visualized the data through the scatter plot about the scores of pairs after transformation (y-axis) over those before transformation (x-axis), and the result is provided in Fig. 10. We also depict the threshold from the benchmark before/after transformation by drawing horizontal and vertical solid lines. From this figure, we can observe that the points in the 2nd and 4th quadrants correspond to the case that the FR model gives a different decision because of the transformation. More precisely, points in the 2nd quadrant were recognized as the different person but not after transformation, and vice versa. Since FPIR is fixed for each threshold, we can ensure that points in the 2nd quadrant contribute to increasing the accuracy after transformation, and vice versa.

From these figures, first we can observe that at the left hand side (Gallery 1) of Fig. 10a, we can observe that more samples lie in the 2nd quadrant than the 4th quadrant when FPIR=1e-4, whereas these quantities are almost the same in FPIR=1e-2. On the other hand, on the right hand side (Gallery 2), the number of samples in the 2nd and 4th quadrants is similar when FPIR=1e-4, whereas there are more samples in 4th quadrant when FPIR=1e-2. We figured out that this corresponds to the benchmark result provided in Tab. 3. We can observe that in gallery 1, there is a huge increase on TPIR (6.45%) after transformation when FPIR=1e-4, whereas there is a slight degradation on TPIR (0.36%) when FPIR=1e-2. Both results were expected from Fig. 10a. We can interpret the result in $(\alpha, \beta) = (63, 63)$ in a similar way.

Along with the above-mentioned analysis based on "bad" events, we can also observe that larger variations of the accuracy appear when FPIR is low. We guess that this coincides with our analysis on the distance preservation property of the proposed transformation. More precisely, the acceptance threshold in low FPIR settings becomes tighter, *i.e.*, becomes larger in our setting compared to relatively high FPIR settings, and as we observed in Figure 4, IDFace would affect the distance relationship more at this region. Concretely, in the IJB-C benchmark for identification, the threshold for FRIR=1e-4 is set to 0.7888, whereas that for FPIR=1e-2 is set to 0.3649 in gallery 1 without transformation. Thus, we can expect that the accuracy more fluctuates in a stochastic way as the threshold becomes tighter, and this coincides with our analysis results.

We remark that although our analysis provides details about how the proposed transformation affects the accuracy of the FR model, this does not tell us why such a phenomenon *should* occur. At this moment, we guess that our results are merely (un)fortunate coincidences on "bad" events. We leave detailed investigations about the phenomenon, e.g., clarifying what conditions make "bad" events happen, as interesting future work.

E. Viewing Almost-Isometric Transformation as a Ternary Quantization Method

As mentioned in Section 4, one may regard our almost-isometric transformation as a ternary quantization tailored for real-valued unit vectors. In fact, the quantization methods themselves are frequently appearing in several realms of deep learning literature, especially for reducing the computational cost of doing expensive operations [19, 37, 41, 69], *e.g.*, matrix multiplication. For this reason, employing quantization methods for improving efficiency would seem to be rather natural. Nevertheless, we found that many previous HE-based BTP constructions [1, 3, 10, 11, 18, 42, 46, 57] regarded quantization methods as a way of fitting the real-valued templates to discrete spaces where the cryptographic tools are defined, *e.g.*, finite fields. In addition, analysis of quantization methods in terms of distance preservation has not been considered in their works. To our knowledge, ours is the first attempt to exploit the ternary quantization method for designing an efficient HE-based BTP, along

with a theoretical guarantee on the distance preservation.

Of course, the proposed almost-isometric transformation may be replaced to other ternary quantization methods. However, when considering our application scenario, we found that the quantization method would satisfy some desirable properties tailored for IDFace. First, it would be beneficial if the quantization method is independent of the distribution of the input data. If not the quantization method does depend on the distribution of the face templates, which is determined by the choice of feature extractor, then the resulting BTP scheme becomes no longer applicable in a plug-and-play manner; additional processes such as training would be required for the quantization method. In fact, several quantization methods, *e.g.*, product quantization [22], entropy-based quantization [33], or quantization methods for the weight of neural networks [19, 37, 41, 69], exploit the distribution of input data or require additional training; so they do not satisfy this property.

In addition, to ensure the correct computation of the cosine similarity, we recommend that the nonzero elements of each quantized vector would not depend on the input vector. Recall that when we compute the cosine similarity between two quantized vectors $T_\alpha(\mathbf{x})$ and $T_\beta(\mathbf{y})$, we omitted the normalization of each vector. In our almost-isometric transformation, this does not affect the correctness of the cosine similarity value because the ℓ_2 norm of $T_\alpha(\mathbf{x})$ and $T_\beta(\mathbf{y})$ are always $\sqrt{\alpha}$ and $\sqrt{\beta}$, respectively, regardless of the input vectors $\mathbf{x}$ and $\mathbf{y}$. That is, the cosine similarity between $T_\alpha(\mathbf{x})$ and $T_\beta(\mathbf{y})$ can be viewed as the inner product $\langle T_\alpha(\mathbf{x}), T_\beta(\mathbf{y})\rangle$ with a constant factor $\frac{1}{\sqrt{\alpha\beta}}$. Of course, one may mitigate this issue by post-processing the output of the quantization method or slightly tweak the proposed IDFace. Nevertheless, ternary quantization methods without this property, *e.g.*, setting intervals for assigning the corresponding value for each component such as equal-width or equal-probability quantization methods [9, 39], would not be employed for IDFace *as is*.

By reflecting these properties, few quantization methods remain. Especially, these methods can be classified via the number of nonzero elements that appeared in the quantized vector. At this moment, we now introduce our rationale to select our almost-isometric transformation. Recall that to minimize the change of cosine similarity carried by the quantization, as seen in Section C, it is beneficial to make the number of possible quantization results as large as possible. Since the proposed transformation fully utilizes the space of $\mathcal{Z}_\alpha$, along with a mathematical proof that it is indeed almost-isometric regardless of the input data, we selected it for our IDFace. We leave investigations and analyses on other quantization methods satisfying those two requirements as future work.

F. Omitted Algorithms and Full Description of IDFace

In this section, we provide full descriptions of omitted algorithms and the proposed IDFace for the sake of completeness.

F.1. Omitted Algorithms

We first provide the full algorithms of the almost-isometric transformation (T_α in Section 4.1) and the space-efficient encoding (Encode and Decode in Section 4.2), each of which is described in Algorithm 1, 2 and 3, respectively. Recall that $\mathcal{Z}_\alpha^d$ is a set of ternary vectors of length d with α nonzero components whose components are one of $\{0, \pm 1\}$.

Algorithm 1 Transformation T_α

Functionality: $T_\alpha : \mathbb{S}^{d-1} \to \mathcal{Z}_\alpha^d$
Require: $\mathbf{x} = (x_1, \ldots, x_d) \in \mathbb{S}^{d-1}$ and $\alpha \in [d]$,
1: Find a set $J \subset [d]$ of α indices such that for any $j \in J$ and $k \notin J$, $|x_j| \geq |x_k|$.
2: For all $j \in [d]$, set $z_j = \begin{cases} \frac{x_j}{|x_j|} & \text{if } j \in J \\ 0 & \text{if } j \notin J \end{cases}$
3: **return** $\mathbf{z} = (z_1, \ldots, z_d) \in \mathcal{Z}_\alpha^d$

Algorithm 2 Encode

Require: $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathcal{Z}_\alpha^d$ and $p \in \mathbb{N}$ such that $p \geq \alpha$
1: For $i \in [m]$, compute $\mathbf{x}_i^+ = (|\mathbf{x}_i| + \mathbf{x}_i)/2$ and $\mathbf{x}_i^- = (|\mathbf{x}_i| - \mathbf{x}_i)/2$
2: For $* \in \{+, -\}$, compute $\mathbf{x}^* = \sum_{i=1}^{m=1} p^{i-1} \cdot \mathbf{x}_i^*$
3: **return** $(\mathbf{x}^+, \mathbf{x}^-)$.

In addition, we provide the full algorithms of our improved database encryption scheme ($\mathsf{IDFace.ENC_{DB}}$, $\mathsf{IDFace.IP_{DB}}$) in Algorithm 4 and 5, respectively, which were introduced in Section 4.3.

Algorithm 3 Decode

Require: $(\mathbf{x}^+, \mathbf{x}^-) \in \mathbb{N}^d \times \mathbb{N}^d$ and $p \in \mathbb{N}$

For i from 1 to m do:
  For $* \in \{+, -\}$, compute $\mathbf{s}_i^* \leftarrow \mathbf{x}^* \pmod p$
  Compute $\mathbf{s}_i \leftarrow \mathbf{s}_i^+ - \mathbf{s}_i^-$
  For $* \in \{+, -\}$, update $\mathbf{x}^* \leftarrow \frac{\mathbf{x}^* - \mathbf{s}_i^*}{p}$
return $(\mathbf{s}_1, \ldots, \mathbf{s}_m)$

Algorithm 4 $\mathsf{IDFace.ENC}_{\mathtt{DB}}$

Require: $\mathbf{X} \in \mathbb{R}^{mN \times d}$, $\alpha \in [d]$, and pk

Parse $\mathbf{X}$ as $[\mathbf{x}_1, \ldots, \mathbf{x}_{mN}]^T$
For $i \in [mN]$, compute $\mathbf{z}_i = T_\alpha(\mathbf{x}_i)$
For $i \in [N]$, compute $(\mathbf{x}_i^+, \mathbf{x}_i^-) \leftarrow \mathsf{Encode}(\mathbf{z}_{(i-1)m+1} \ldots, \mathbf{z}_{im})$
For $* \in \{+, -\}$, set $\mathbf{X}^* = [\mathbf{x}_1^*, \ldots, \mathbf{x}_N^*]^T$.
For $* \in \{+, -\}$, set $\mathcal{C}^* \leftarrow \mathsf{ENC}_{\mathtt{DB}}(\mathbf{X}^*, \mathsf{pk})$
return $\mathcal{C} = (\mathcal{C}^+, \mathcal{C}^-)$

Algorithm 5 $\mathsf{IDFace.IP}_{\mathtt{DB}}$

Require: $\mathbf{y} \in \mathbb{R}^d$, a set $\mathcal{C}$, and $\beta \in [d]$

$\mathbf{z} \in \{-1, 0, 1\}^d \leftarrow T_\beta(\mathbf{y})$
Compute $\mathbf{z}^+ = (|\mathbf{z}| + \mathbf{z})/2$ and $\mathbf{z}^- = (|\mathbf{z}| - \mathbf{z})/2$.
Parse $(\mathcal{C}^+, \mathcal{C}^-)$ as $\mathcal{C}$ and compute
  $\mathbf{ct}^+ = \mathsf{IP}_{\mathtt{DB}}(\mathbf{z}^+, \mathcal{C}^+) \oplus \mathsf{IP}_{\mathtt{DB}}(\mathbf{z}^-, \mathcal{C}^-)$,
  $\mathbf{ct}^- = \mathsf{IP}_{\mathtt{DB}}(\mathbf{z}^+, \mathcal{C}^-) \oplus \mathsf{IP}_{\mathtt{DB}}(\mathbf{z}^-, \mathcal{C}^+)$
return $(\mathbf{ct}^+, \mathbf{ct}^-)$

F.2. Full Description of IDFace

We now provide the full description of IDFace. Recall that IDFace is an instantiation of the biometric identification scheme from the proposed database encryption scheme $(\mathsf{IDFace.ENC}_{\mathtt{DB}}, \mathsf{IDFace.IP}_{\mathtt{DB}})$. Although its generic construction was already given in Section 4.3, we give a detailed description for the sake of completeness. Here, DEC denotes the decryption algorithm of the underlying AHE, and $(\mathsf{pk}, \mathsf{sk})$ is a pair of the public key and secret key of it. In addition, we consider the database DB as a list, which is initialized as an emptyset. The description of IDFace is given in Fig. 11.

G. Secure Two-Party Computation-based Variant of IDFace

Recall that the proposed transformation is independent of the HE, so that it can be utilized to speed up secure face identification protocols from other cryptographic tools. As an example of this, we provide another face identification protocol called 2PCFace from a two-party computation-based approach.

The basic idea of designing 2PCFace is as follows: For two binary vectors $\mathbf{x}, \mathbf{y} \in \{0, 1\}^d$, their inner product can be computed as $\langle \mathbf{x}, \mathbf{y} \rangle = \mathcal{HW}(\mathbf{x} \wedge \mathbf{y})$, where $\mathcal{HW}$ denotes the Hamming weight of the given binary vector, and $\wedge$ is a component-wise AND operation. If we consider an additive share $(\mathbf{x}_1, \mathbf{x}_2)$ of x, which means $\mathbf{x}_1 \oplus \mathbf{x}_2 = \mathbf{x}$ for component-wise XOR operator $\oplus$, we have that $\langle \mathbf{x}, \mathbf{y} \rangle = \mathcal{HW}((\mathbf{x}_1 \wedge \mathbf{y}) \oplus (\mathbf{x}_2 \wedge \mathbf{y}))$. From this, we can construct an identification protocol with two servers, $\mathcal{S}_1$ and $\mathcal{S}_2$, where each server stores the enrolled templates after transformation in the form of an additive share. After receiving the transformed template $\mathbf{y}$, the matching score between the enrolled templates can be obtained by first calculating $\mathbf{x}_1 \wedge \mathbf{y}$ and $\mathbf{x}_2 \wedge \mathbf{y}$ locally for each server, sharing the result between servers, and finally calculating $\mathcal{HW}((\mathbf{x}_1 \wedge \mathbf{y}) \oplus (\mathbf{x}_2 \wedge \mathbf{y}))$ using the shared value. Although the output of almost-isometric transformation is a ternary vector comprised of $\{-1, 0, 1\}$, we can decompose it into the subtraction of two binary vectors by the sign of each component, as we did in the space-efficient encoding.

For a precise description, we introduce a helper algorithm called GenShare, which will be a subroutine of 2PCFace. GenShare takes a template $\mathbf{x} \in \mathbb{R}^{mN \times d}$ and a transformation parameter α as inputs, returning additive shares of the positive components and negative components of $T_\alpha(\mathbf{x})$, respectively. The formal description of Share is provided in Algorithm 6. Using GenShare with the above idea to compute matching scores, we describe the full protocol of 2PCFace in Figure 12.

Since component-wise AND operations can be seen as look-up operations, we can utilize the same efficiency-accuracy trade-off trick in 2PCFace by employing different transformation parameters (α, β) in Enroll and Identify, respectively. More precisely, during identification, we can treat $\mathbf{z}_{i,j}^\dagger \wedge \mathbf{y}^*$ for $\dagger, * \in \{+, -\}$ as look-up components of $\mathbf{z}_{i,j}^\dagger$ on the positions where corresponding components in $\mathbf{y}^*$ are nonzero. Thus, if we denote $\mathbf{z}_{i,j}^\dagger[\mathbf{y}^*]$ as a subvector of $\mathbf{z}_{i,j}^\dagger$ by collecting the positions where corresponding $\mathbf{y}^*$'s components are nonzero, then we can optimize the Identify in terms of both communication cost and computational cost by replacing each $\mathbf{p}_{i,j}^{(1)}, \mathbf{p}_{i,j}^{(2)}, \mathbf{m}_{i,j}^{(1)}, \mathbf{m}_{i,j}^{(2)}$ to $\mathbf{z}_{i,j}^+[\mathbf{y}^+], \mathbf{z}_{i,j}^-[\mathbf{y}^-], \mathbf{z}_{i,j}^+[\mathbf{y}^-], \mathbf{z}_{i,j}^-[\mathbf{y}^+]$, respectively. Note that the remaining operations are still well-defined because each corresponding binary vector computed by each server would

Enroll

Parameter: $\alpha \in [d]$ and a bound $p \geq \alpha$.

$\mathcal{S}_{local}$'s input: $\mathbf{X} \in \mathbb{R}^{mN \times d}$, $\mathtt{ID}$, $\mathtt{DB}$, and pk

---

$\boxed{\mathcal{S}_{local}}$: $\mathcal{C} \leftarrow \mathsf{IDFace.ENC}_{\mathtt{DB}}(X, \alpha, \mathsf{pk})$

$\boxed{\mathcal{S}_{local}}$: Add $(\mathtt{ID}, \mathcal{C})$ on the bottom row of $\mathtt{DB}$

Identify

Parameter: $\beta \in [d]$, a threshold τ, and a bound p.

$\mathcal{S}_{local}$'s input: $\mathbf{y} \in \mathbb{R}^d$ and $\mathtt{DB}$ of size D

$\mathcal{S}_{key}$'s input: secret key sk

$\boxed{\mathcal{S}_{local}}$: For $(\mathtt{ID}_i, \mathcal{C}_i) \in \mathtt{DB}$, compute $(\mathbf{ct}_i^+, \mathbf{ct}_i^-) \leftarrow \mathsf{IDFace.IP}_{\mathtt{DB}}(\mathbf{y}, \mathcal{C}_i, \beta)$

$\boxed{\mathcal{S}_{local} \rightarrow \mathcal{S}_{key}}$: Send $(\mathbf{ct}_i^+, \mathbf{ct}_i^-)_{i=1}^D$.

$\boxed{\mathcal{S}_{key}}$: For $i \in [D]$ and $* \in \{+, -\}$ do:
$\mathbf{pt}_i^* \leftarrow \mathsf{DEC}_{\mathsf{sk}}(\mathbf{ct}_i^*)_{i=1}^D$

$\boxed{\mathcal{S}_{local}}$: For $i \in [D]$ do:
For $* \in \{+, -\}$, parse $\mathbf{pt}_i^*$ as $(s_{ij}^*)_{j=1}^N$.
For $j \in [N]$, compute $(s_{ijk})_{k=1}^m \leftarrow \mathsf{Decode}(s_{ij}^+, s_{ij}^-, p)$.
If $\max_{i,j,k}\{s_{ijk}\} > \tau$:
Set $idx \leftarrow (i, mj + k - 1)$.
Otherwise, Set $idx \leftarrow \perp$.

$\boxed{\mathcal{S}_{key} \rightarrow \mathcal{S}_{local}}$: Send idx.

$\boxed{\mathcal{S}_{local}}$: If $idx \neq \perp$:
Parse $idx \rightarrow (r, s)$ and **return** $\mathtt{ID}_r[s]$.
Otherwise, **return** "reject".

Figure 11. Full description of IDFace

be the same length.

Compared to HE-based approaches, including IDFace, 2PCFace has some advantages in computational efficiency and the underlying security assumption. For the former, we note that each server in 2PCFace stores bit strings of the same length as the transformed template, thus preventing the storage overhead from the encryption. In addition, the unit operation conducted by each server becomes bit-wise operations, which are considerably cheaper than homomorphic addition and (scalar) multiplication. For the latter, 2PCFace does not require an assumption for securing the secret key, which is an essential requirement in HE-based approaches. In fact, the attacker cannot distinguish between the shares of enrolled templates and a random string of equal length unless the attacker steals the whole databases of both servers. Moreover, under the same

Algorithm 6 GenShare

Require: A template $\mathbf{x} \in \mathbb{R}^d$, and a parameter α
Ensure: $\mathbf{z}_1^+, \mathbf{z}_1^-, \mathbf{z}_2^+, \mathbf{z}_2^- \in \{0,1\}^d$ such that $(\mathbf{z}_1^+ \oplus \mathbf{z}_2^+) - (\mathbf{z}_1^- \oplus \mathbf{z}_2^-) = T_\alpha(\mathbf{x})$
1: Compute $\mathbf{z} \leftarrow T_\alpha(\mathbf{x})$ and set $\mathbf{z}^+ \leftarrow (|\mathbf{z}| + \mathbf{z})/2$ and $\mathbf{z}^- \leftarrow (|\mathbf{z}| - \mathbf{z})/2$.
2: Sample $\mathbf{z}_1^+, \mathbf{z}_1^- \xleftarrow{\$} \{0,1\}^d$ and set $\mathbf{z}_2^+ \leftarrow \mathbf{z}_1^+ \oplus \mathbf{z}^+$ and $\mathbf{z}_2^- \leftarrow \mathbf{z}_2^+ \oplus \mathbf{z}^-$.
3: **return** $(\mathbf{z}_1^+, \mathbf{z}_2^+, \mathbf{z}_1^-, \mathbf{z}_2^-)$.

attacker, one can easily check that 2PCFace satisfies all standard security requirements for BTP: *irreversibility*, *revocability*, and *unlinkability*. We also remark that the amount of accuracy degradation in 2PCFace is identical to that of IDFace, under the same transformation parameters (α, β).

However, compared to IDFace, 2PCFace has some downsides on (1) its communication cost and (2) extension to the scenario with mulitple devices and servers, which is presented in Appendix H.

Comparison on the Communication Cost. In IDFace, the ciphertext of whole inner product values can be computed locally, but this is impossible in 2PCFace; In this case, each server calculates only a partial part of inner product values and communicates with each other to obtain the full inner product value. That is, for one execution of Identify, the server $\mathcal{S}_2$ in 2PCFace should send the bit string of the same length as the entire database in one server, whereas IDFace suffices to send the encryption/decryption of the inner product value corresponding to each stored identity between servers $\mathcal{S}_{local}$ and $\mathcal{S}_{key}$. Although the cost of the former can be reduced by sending subvectors rather than the entire one, we figure out that the communication cost of IDFace is cheaper than 2PCFace under the same transformation parameters.

For a concrete comparison, we provide the formula for the measuring communication cost of each protocol as follows: For simplicity, we assume that the transformation parameters are (α, β), the dimension of the template is d, and the number of enrolled identities is D.

Communication Cost of IDFace**.** We can figure out that the communication occurs when (1) $\mathcal{S}_{local}$ sends $(\mathbf{ct}_i^+, \mathbf{ct}_i^-)_{i=1}^{\hat{D}}$ to $\mathcal{S}_{key}$ and (2) $\mathcal{S}_{key}$ sends idx to $\mathcal{S}_{local}$, where $\hat{D} = \lceil \frac{D}{N_s \cdot N_{enc}} \rceil$ for the number of slots N_s in underlying AHE and the number of templates N_{enc} encoded at once. More precisely, $2\hat{D}$ ciphertexts and 1 string representing the index are communicated. Since $\lceil \log D \rceil$ bits suffice for representing all D identities, the total communication cost is $2\hat{D} \cdot S_{ct} + \log D$, where S_{ct} stands for the size of ciphertext.

Communication Cost of 2PCFace**.** We can figure out that the communication occurs when (1) $\mathcal{S}_1$ sends $(\mathbf{y}^+, \mathbf{y}^-)$ to $\mathcal{S}_2$ and (2) $\mathcal{S}_2$ sends subvectors $(\mathbf{z}_{i,2}^+[\mathbf{y}^+], \mathbf{z}_{i,2}^-[\mathbf{y}^-], \mathbf{z}_{i,2}^+[\mathbf{y}^-], \mathbf{z}_{i,2}^-[\mathbf{y}^+])_{i=1}^D$ to $\mathcal{S}_1$. The former takes $2d$ bits of communication. For the latter, we can observe that since the number of nonzero entries in $\mathbf{y}$ is β, the concatenation of these 4 vectors in the index i has length 2β. Hence, the latter requires $2N\beta$ bits of communication, and the total cost is $2D\beta + 2d$ bits.

By following these formulae, we compute the exact communication cost of IDFace instantiated by CKKS and Paillier Cryptosystem (PC), and 2PCFace. For simplicity, we denote $\mathsf{IDFace}_{\mathsf{CKKS}}$ and $\mathsf{IDFace}_{\mathsf{PC}}$ for the instantiation of IDFace using CKKS and PC, respectively. The number of enrolled identities D is set to 1 M. The corresponding N_s, N_{enc} and S_{ct} in the parameter settings of CKKS and PC used in IDFace[2], along with the comparison result on the communication cost, are provided in Table 4. According to the table, the communication cost of $\mathsf{IDFace}_{\mathsf{CKKS}}$ ($\mathsf{IDFace}_{\mathsf{PC}}$, resp.) is about 1.84X $\sim$ 6.28X (5.13X $\sim$ 18.44X, resp.) cheaper than 2PCFace. Thus, for circumstances where the bandwidth of the communication channel is limited (*e.g.,* 100MB/s or 1GB/s), deploying IDFace would be more favorable than 2PCFace.

Protocols	N_s	N_{enc}	S_{ct}
$\mathsf{IDFace}_{\mathsf{CKKS}}$	4096	(8, 7, 5)	132KB
$\mathsf{IDFace}_{\mathsf{PC}}$	1	(341, 292, 227)	0.5KB

(α, β)	$\mathsf{IDFace}_{\mathsf{CKKS}}$	$\mathsf{IDFace}_{\mathsf{PC}}$	2PCFace
(341, 341)	12.94MB	4.41MB	81.30MB
(341, 127)	9.24MB	3.43MB	30.28MB
(341, 63)	8.18MB	2.93MB	15.02MB

Table 4. N_s, N_{enc}, and S_{ct} for $\mathsf{IDFace}_{\mathsf{CKKS}}$ and $\mathsf{IDFace}_{\mathsf{PC}}$ (Left), and comparison of communication cost required for one identification in each protocol (Right). In the column of N_{enc}, the tuple indicates N_{enc} values for $\beta = 63, 127, 341$, respectively. The number of identities is set to 1M.

[2]The detailed parameter settings of each scheme were already provided in Section 5.

Enroll

Parameter: $\alpha \in [d]$

$\mathcal{S}_1$'s input: $\mathbf{x} \in \mathbb{R}^d$, id, $\mathtt{DB}_1$

$\mathcal{S}_2$'s input: $\mathtt{DB}_2$

$\boxed{\mathcal{S}_1}$: $(\mathbf{z}_1^+, \mathbf{z}_2^+, \mathbf{z}_1^-, \mathbf{z}_2^-) \leftarrow \mathsf{GenShare}(\mathbf{x}, \alpha)$.

$\boxed{\mathcal{S}_1 \rightarrow \mathcal{S}_2}$: Send $(\mathbf{z}_2^+, \mathbf{z}_2^-)$

$\boxed{\mathcal{S}_1, \mathcal{S}_2}$: Add $(id, \mathbf{z}_i^+, \mathbf{z}_i^-)$ on the bottom row of $\mathtt{DB}_i$ for $i \in \{1, 2\}$

Identify

Parameter: $\beta \in [d]$, threshold τ.

$\mathcal{S}_1$'s input: $\mathbf{y} \in \mathbb{R}^d$ and $\mathtt{DB}_1$ of size D

$\mathcal{S}_2$'s input: $\mathtt{DB}_2$ of size D

$\boxed{\mathcal{S}_1}$: Set $\tilde{\mathbf{y}} \leftarrow T_\beta(\mathbf{y})$, $\mathbf{y}^+ \leftarrow (|\tilde{\mathbf{y}}| + \tilde{\mathbf{y}})/2$ and $\mathbf{y}^- \leftarrow (|\tilde{\mathbf{y}}| - \tilde{\mathbf{y}})/2$.

$\boxed{\mathcal{S}_1 \rightarrow \mathcal{S}_2}$: Send $(\mathbf{y}^+, \mathbf{y}^-)$

$\boxed{\mathcal{S}_1, \mathcal{S}_2}$: For $(id_i, \mathbf{z}_{i,j}^+, \mathbf{z}_{i,j}^-) \in \mathtt{DB}_j$ do ($j \in \{1, 2\}$):
$\mathbf{p}_{i,j}^{(1)} \leftarrow \mathbf{z}_{i,j}^+ \wedge \mathbf{y}^+$, $\mathbf{p}_{i,j}^{(2)} \leftarrow \mathbf{z}_{i,j}^- \wedge \mathbf{y}^-$,
$\mathbf{m}_{i,j}^{(1)} \leftarrow \mathbf{z}_{i,j}^+ \wedge \mathbf{y}^-$, and $\mathbf{m}_{i,j}^{(2)} \leftarrow \mathbf{z}_{i,j}^- \wedge \mathbf{y}^+$

$\boxed{\mathcal{S}_2 \rightarrow \mathsf{S}_1}$: Send $(\mathbf{p}_{i,2}^{(1)}, \mathbf{p}_{i,2}^{(2)}, \mathbf{m}_{i,2}^{(1)}, \mathbf{m}_{i,2}^{(2)})_{i=1}^D$

$\boxed{\mathcal{S}_1}$: For $i \in [D]$ do:
Set $s_i \leftarrow (\mathcal{HW}(\mathbf{p}_{i,1}^{(1)} \oplus \mathbf{p}_{i,2}^{(1)}) + \mathcal{HW}(\mathbf{p}_{i,1}^{(2)} \oplus \mathbf{p}_{i,2}^{(2)}))$
$-(\mathcal{HW}(\mathbf{m}_{i,1}^{(1)} \oplus \mathbf{m}_{i,2}^{(1)}) + \mathcal{HW}(\mathbf{m}_{i,1}^{(2)} \oplus \mathbf{m}_{i,2}^{(2)}))$
If $\max_i\{s_i\} > \tau$:
return the corresponding id_i.
Otherwise, **return** $\perp$

Figure 12. Full description of 2PCFace.

H. Application of IDFace for Scenarios with Multiple Devices and Servers

Recall that both IDFace and 2PCFace regard the scenario in which there is only a single device for recognition. We now consider an extension for this: The large-scale identification scenario where multiple recognition devices are involved. Although it is possible for a single server to deal with multiple devices, its computational burden becomes extremely severe if we consider lots of devices, *e.g.* more than a hundred, or 1K. As shown in some real-world applications [15, 61], the computational burden for the server in this circumstance can be alleviated by employing multiple servers to distribute the Identify requests

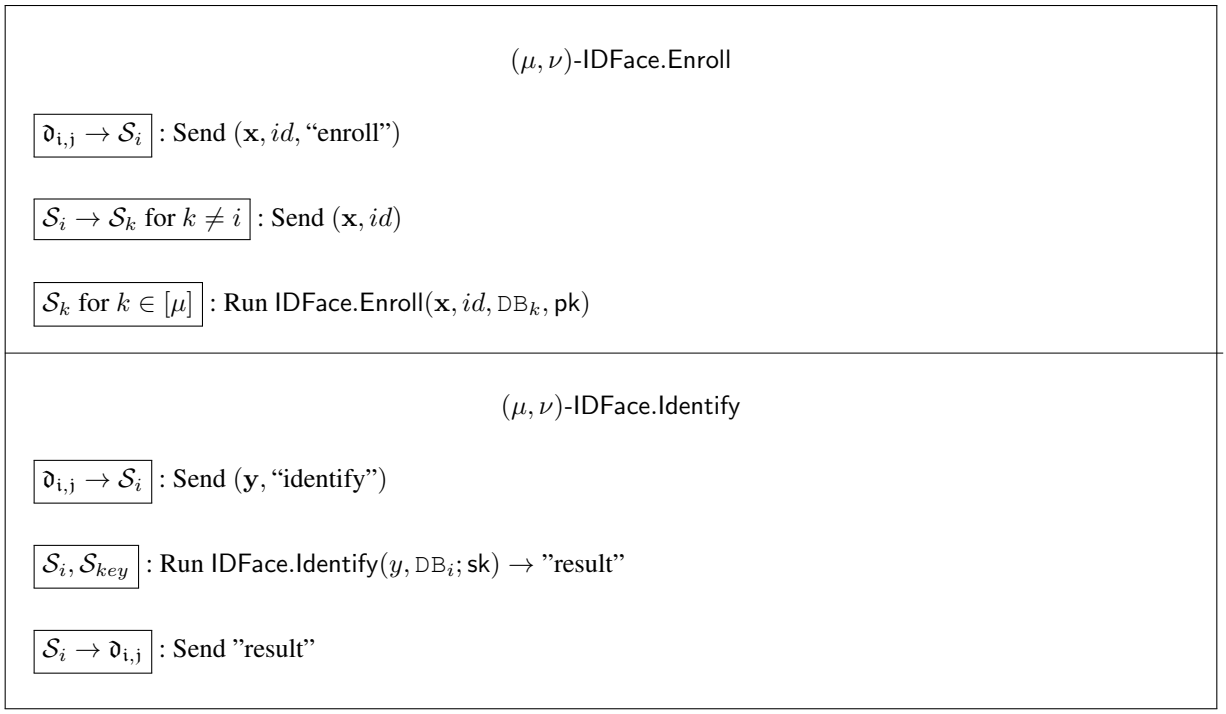


Figure 13. Description of (μ, ν)-IDFace

from the whole device. We assume that these multiple servers are well-synchronized, holding the same enrolled templates in the form of encrypted ones. Note that the stored data on each server is not necessarily identical. In fact, it is desirable to ensure that they are (computationally) indistinguishable from the random string for the sake of unlinkability. In this paragraph, we will show that IDFace can be naturally extended with a negligible overhead on communication cost, whereas the extension of 2PCFace requires relatively larger communication cost overhead proportional to the number of servers.

For a precise comparison, let us consider the case where there are μ servers, denoted by $\mathcal{S}_1, \ldots, \mathcal{S}_\mu$, that store enrolled templates and assume that each server handles the requests from ν devices, denoted by $\mathfrak{d}_{i,1}, \ldots, \mathfrak{d}_{i,\nu}$ for the server $\mathcal{S}_i$. We denote $\mathtt{DB}_i$ as the database of each server, which is initialized to an empty list. In particular, for extending IDFace, we assume that there is only one key server S_{key} in order to minimize the threat of the leakage of the secret key. We will call the extension of IDFace (2PCFace, resp.) as (μ, ν)-IDFace ((μ, ν)-MPCFace, resp.). For ease of description, we assume that the device $\mathfrak{d}_{i,j}$ initially provides the input $(\mathbf{x}, id,$ "enroll") or $(\mathbf{y},$ "identify") to the associated server $\mathcal{S}_i$ for Enroll and Identify, respectively. In addition, we omit the threshold parameter τ for Identify and transformation parameters α, β, which are used for Enroll and Identify, respectively.

Description of (μ, ν)-IDFace. We denote $(\mathsf{pk}, \mathsf{sk})$ as the public key and secret key pair of the underlying AHE scheme. In addition, we will utilize IDFace.Enroll and IDFace.Identify as subroutines. Using them, we formally describe (μ, ν)-IDFace as Figure 13:

Description of (μ, ν)-MPCFace. For designing the extension, a naive approach is to divide each sever $\mathcal{S}_i$ into sub-servers $(\mathsf{S}_{i1}, \mathcal{S}_{i2})$ and follow the construction of (μ, ν)-IDFace, using 2PCFace as a subroutine. We will not consider this approach because, in this case, the adversary can recover the whole enrolled template by compromising only two sub-servers $\mathcal{S}_{i1}, \mathcal{S}_{i2}$. Rather, we extend 2PCFace by the computation with μ-parties by the following observation: For an additive share $(\mathbf{x}_1, \ldots, \mathbf{x}_\mu)$ of $\mathbf{x}$, *i.e.*, $\oplus_{i=1}^{\mu} \mathbf{x}_i = \mathbf{x}$, we have that $\langle \mathbf{x}, \mathbf{y} \rangle = \mathcal{HW}(\oplus_{i=1}^{\mu}(\mathbf{x}_i \wedge \mathbf{y}))$. From this, we extend the phase of broadcasting $\mathbf{x}_i \wedge \mathbf{y}$ in 2PCFace.Identify to μ parties.

We will denote GenShare$(\mathbf{x}; \mu)$ as an extension of the original GenShare algorithm to make an additive share $(\mathbf{z}_1^\dagger, \ldots, \mathbf{z}_\mu^\dagger)_{\dagger \in \{+,-\}}$ of transformed $\mathbf{x}$ for μ parties. In addition, we denote DBAND as an algorithm that takes a database $\mathtt{DB} = (id_i, \mathbf{z}_i^+, \mathbf{z}_i^-)_{i=1}^D$ and two binary strings $\mathbf{y}^+, \mathbf{y}^- \in \{0,1\}^d$ as inputs and returning tuples of sub-vectors of each $\mathbf{z}_i^+$ and $\mathbf{z}_i^-$: $(\mathbf{z}_i^+[\mathbf{y}^+], \mathbf{z}_i^-[\mathbf{y}^-], \mathbf{z}_i^+[\mathbf{y}^-], \mathbf{z}_i^-[\mathbf{y}^+])_{i=1}^D$. Using them as subroutines, we formally describe (μ, ν)-MPCFace as Figure 14:

(μ, ν)-MPCFace.Enroll

$\boxed{\mathfrak{d}_{\mathrm{i,j}} \to \mathcal{S}_i}$: Send $(\mathbf{x}, id,$ "enroll").

$\boxed{\mathcal{S}_i}$: Run $\mathsf{GenShare}(\mathbf{x}; \mu) \to (\mathbf{z}_1^\dagger, \ldots, \mathbf{z}_\mu^\dagger)_{\dagger \in \{+,-\}}$

$\boxed{\mathcal{S}_i \to \mathcal{S}_k \text{ for } k \neq i}$: Send $(id, \mathbf{z}_k^+, \mathbf{z}_k^-)$.

$\boxed{\mathcal{S}_k \text{ for } k \in [\mu]}$: Add $(id, \mathbf{z}_k^+, \mathbf{z}_k^-)$ to the last row of $\mathtt{DB}_k$.

(μ, ν)-MPCFace.Identify

$\boxed{\mathfrak{d}_{\mathrm{i,j}} \to \mathcal{S}_i}$: Send $(\mathbf{y},$ "identify").

$\boxed{\mathcal{S}_i \to \mathcal{S}_k \text{ for } k \neq i}$: Send $\mathbf{y}^+, \mathbf{y}^-$.

$\boxed{\mathcal{S}_k \text{ for } k \in [\mu]}$: Run $\mathsf{DBAND}(\mathtt{DB}_k, \mathbf{y}^+, \mathbf{y}^-) \to (\mathbf{p}_{k,l}^{(1)}, \mathbf{p}_{k,l}^{(2)}, \mathbf{m}_{k,l}^{(1)}, \mathbf{m}_{k,l}^{(2)})_{l=1}^D$.

$\boxed{\mathcal{S}_k \to \mathcal{S}_i \text{ for } k \neq i}$: Sends $(\mathbf{p}_{k,l}^{(1)}, \mathbf{p}_{k,l}^{(2)}, \mathbf{m}_{k,l}^{(1)}, \mathbf{m}_{k,l}^{(2)})_{l=1}^D$.

$\boxed{\mathcal{S}_i}$: Compute for $l \in [D]$,
$s_l \leftarrow (\mathcal{HW}(\oplus_{k=1}^\mu \mathbf{p}_{k,l}^{(1)}) + \mathcal{HW}(\oplus_{k=1}^\mu \mathbf{p}_{k,l}^{(2)})) - (\mathcal{HW}(\oplus_{k=1}^\mu \mathbf{m}_{k,l}^{(1)}) + \mathcal{HW}(\oplus_{k=1}^\mu \mathbf{m}_{k,l}^{(2)}))$.

$\boxed{\mathcal{S}_i \to \mathfrak{d}_{i,j}}$: If $\max_{l \in [D]}\{s_l\} > \tau$:
Send corresponding id_i to $\mathfrak{d}_{i,j}$,
Otherwise: Send $\perp$ to $\mathfrak{d}_{i,j}$.

Figure 14. Description of (μ, ν)-MPCFace.

Communication Overheads From Extensions. For each construction, the advantage of (μ, ν)-IDFace over (μ, ν)-MPCFace is that the former can run IDFace.Identify "locally". That is, the remaining servers do nothing during the protocol. For this reason, additional communication on the former occurs only for broadcasting $(\mathbf{x}, id)$ during Enroll. Concretely, with transformation before sending, the communication cost for this is $(\mu - 1)(2d + |id|)$, where $|id|$ is the bit length of id. We note that $|id| = \lceil \log_2 D \rceil$ suffices to distinguish all enrolled identities, so that the asymptotic communication overhead with respect to μ and D is $O(\mu \log D)$. For our experimental setting where $d = 512$ and the database of size $D = 1$M, $|id| = 20 > \log D$ suffices, so $(2d + |id|) \approx 0.26$KB.

But in the latter, communication occurs between servers in both Enroll and Identify: In Enroll, the additive share of input $\mathbf{x}$ with id is sent to other servers, and in Identify the output of DBAND is merged to a single server. Although the size of the former is identical to that from (μ, ν)-IDFace, the latter is quite large. According to the communication cost formula of 2PCFace, we have that the latter carries $(\mu - 1)(2D\beta + 2d)$, or asymptotically $O(\mu D)$, communication overhead. As we can infer from the result in Table 4, for the dataset with 1M enrolled identities, this overhead becomes extremely large (from 285MB to 1544MB) even when $\mu = 21$. The communication cost overhead with respect to various parameters and the

μ	Parameters (α, β)	Enroll			Identify		
		2PCFace	MPCFace	Overhead	2PCFace	MPCFace	Overhead
3	(341,341)	0.13KB	0.26KB	+0.13KB	81.30MB	162.60MB	+81.30MB
	(341,127)				30.28MB	60.56MB	+30.28MB
	(341,63)				15.02MB	30.04MB	+15.02MB
6	(341,341)	0.13KB	0.64KB	+0.51KB	81.30MB	406.50MB	+325.20MB
	(341,127)				30.28MB	151.39MB	+121.11MB
	(341,63)				15.02MB	75.10MB	+60.08MB
11	(341,341)	0.13KB	1.28KB	+1.15KB	81.30MB	813.01MB	+731.71MB
	(341,127)				30.28MB	302.79MB	+272.51MB
	(341,63)				15.02MB	150.20MB	+135.18MB
21	(341,341)	0.13KB	2.55KB	+2.42KB	81.30MB	1626.02MB	+1544.72MB
	(341,127)				30.28MB	605.59MB	+575.31MB
	(341,63)				15.02MB	300.41MB	+285.39MB

Table 5. Communication cost of (μ, ν)-MPCFace with various number of servers and parameters, along with the overhead compared to 2PCFace under the same parameters. The number of identities is set to 1M. We note that the communication cost overhead of (μ, ν)-MPCFace in Enroll is identical to (μ, ν)-IDFace

number of servers μ are summarized in Table 5.

I. Extending IDFace to Other Types of Biometrics

Although we focused on protecting face templates in IDFace, we note that the core contribution of our work is to accelerate the computation of inner products in an encrypted domain while preserving the inner product value as much as possible. That is, our work can be adapted to other biometric authentication systems using different types of biometrics where cosine similarity is used for measuring the matching score, such as fingerprint recognition or speaker recognition. In this section, we investigate the extensibility of IDFace to these tasks.

I.1. Extension 1: Speaker Recognition

We conducted experiments on evaluating the accuracy of three publicly available speaker recognition models with IDFace, including RawNet3 [67], MR-RawNet [2], and ReDimNet [68]. The source codes and pre-trained parameters of all these models are disclosed by the authors. In particular, for ReDimNet, we utilized the (`b6, finetuned`) version in their paper, which showed the best performance among the versions disclosed by the authors. We used the cleaned version of VoxCeleb1 test dataset [55], which is widely used for evaluation in speaker recognition tasks. We used full utterances of each audio sample in the benchmark dataset for feature extraction. For the evaluation metrics, we used Error Equal Rate (EER) and minimum Detection Cost Function (MinDCF). We note that the latter is used for the annual speaker recognition challenge held by NIST [56]. According to the official evaluation plan, we selected the parameter setting `id:1` in the audio track, which is $P_{target} = 0.01, C_{FalseAlarm} = C_{Miss} = 1$. EER and MinDCF can be represented in terms of FAR and FRR, where $\mathsf{FRR}(\tau) = 1 - \mathsf{TAR}(\tau)$, as follows:

$$\mathsf{EER} = \frac{\mathsf{FRR}(\hat{\tau}) + \mathsf{FAR}(\hat{\tau})}{2}, \text{ where } \hat{\tau} = \arg\min_{\tau} |\mathsf{FAR}(\tau) - \mathsf{FRR}(\tau)| .$$

$$\mathsf{MinDCF} = \min_{\tau} \left(C_{\mathsf{miss}} \cdot \mathsf{FRR}(\tau) \cdot (1 - P_{\text{target}}) + C_{\mathsf{FalseAlarm}} \cdot \mathsf{FAR}(\tau) \cdot P_{\text{target}} \right) .$$

As can be inferred from the definitions, smaller EER and MinDCF imply means the model has more discrimination power, *i.e.*, having a better performance.

For parameter selection of IDFace, we followed the same strategy as our analysis in face recognition task. We checked that the RawNet3 and MR-RawNet models output 256-dimensional templates, whereas 192-dimensinoal for ReDimNet. By following our analysis, we select $\alpha = \left\lfloor \frac{2}{3} d \right\rfloor$, which is 170 and 127 for the former two feature extractors and and the latter one, respectively. In addition, we also select β in accordance with α and our accuracy-efficiency trade-off trick: $\beta = 170, 127, 63$ for the former, and $\beta = 127, 63, 31$ for the latter.

The evaluation results can be found in Tab. 6. From this table, we can observe that the accuracy drops in EER for all cases are less than 1%, so does in MinDCF except for the case of $\beta = 31$ on ReDimNet. This demonstrates that our IDFace can be employed for protecting voice templates in large-scale identification tasks without significant accuracy degradation, allowing faster identification even than the case of protecting face templates.

Model	Metric	Plain	IDFace, Parameters: ($\alpha = 170$)		
			$\beta = 170$	$\beta = 127$	$\beta = 63$
RawNet3	EER(%)	0.83	1.19	1.22	1.30
	MinDCF	0.105	0.151	0.135	0.158
MR-RawNet	EER(%)	1.38	1.56	1.59	1.89
	MinDCF	0.131	0.159	0.175	0.183

Model	Metric	Plain	IDFace, Parameters: ($\alpha = 127$)		
			$\beta = 127$	$\beta = 63$	$\beta = 31$
ReDimNet	EER(%)	0.40	0.69	0.75	1.15
	MinDCF	0.033	0.089	0.107	0.146

Table 6. Various speaker verification benchmark results of non-protected template extractor (Plain) and IDFace for various (α, β).

I.2. Extension 2: Fingerprint Recognition

We also conducted experiments on evaluating the accuracy of fingerprint recognition models, DeepPrint [10], for IDFace using FVC2004 dataset [44]. The pre-trained model is publicly available in github by the authors of [59]. The evaluation result can be found in Tab 7. Although the baseline performance (Plain) is significantly lower compared to face and speaker recognition models, it is noteworthy that the accuracy drops in EER are still less than 1%. Note that there are no publicly available large-scale datasets on fingerprint recognition, unlike face recognition or speaker recognition. Therefore, our pre-trained model is trained by synthetic fingerprints data generated by SFinGE [6].

Model	Metric	Plain	$(\alpha, \beta) = (341, 63)$	$(\alpha, \beta) = (341, 127)$	$(\alpha, \beta) = (341, 341)$
DeepPrint [10]	EER(%)	5.295	5.919	5.621	5.617

Table 7. Performance comparison between non-protected fingerprint template extractor (Plain) and IDFace for various β using DeepPrint.

I.3. Discussion on Accuracy Drop

We briefly share our discussions about the tendency of accuracy drops appeared in each task. First of all, in contrast of face recognition and fingerprint recognition tasks, we can observe that the speaker recognition models used in our analysis produces smaller dimension of templates than 512. Since the error term in $\epsilon_{\alpha,\theta}$ becomes larger as d becomes smaller according to Proposition 1, one may expect that the accuracy drop in speaker recognition tasks would be larger than that from the other tasks. However, we observed that the accuracy drop in speaker recognition models is comparable with that of both face and fingerprint recognition tasks. We suspect that this is because the amount of the change of cosine similarity value from our transformation is solely determined by the ratio between the number of nonzeros (α) and the dimension d. Here, if we assume that the error term *randomly* contributes to the inner product value in terms of whether the decision of the recognition model is changed or not due to the transformation, then for a large number of sample pairs, we can expect that the effect of our transformation on the accuracy would be determined by the average error $\epsilon_{\alpha,\theta}$ in a macroscopic viewpoint. We leave a more thorough analysis on this phenomenon as future work.

We also address another non-trivial phenomenon that appeared in Tab. 6: the amount of accuracy drop in MinDCF is slightly larger than that of EER. To this end, we compared the thresholds corresponding to EER and MinDCF for each speaker recognition model, including RawNet3, MR-RawNet, and ReDimNet, obtaining (0.3376, 0.3383, 0.3519) and (0.4720, 0.4944, 0.4339) for each model in EER and MinDCF, respectively. Since the variation of the cosine similarity at the latter is larger than the former according to Fig. 4, we can conclude that this phenomenon does in fact coincide with our theoretical analysis.

J. Additional Figures

In this section, we provide additional figures that were omitted in the main text due to space constraints. In Figure 15, we visualize how the proposed space-efficient encoding technique works, as introduced in Section 4.2. The left hand side shows the procedure Encode to encode m transformed templates $x_1, \ldots, x_m$ into two vectors x^+ and x^-. The right side shows how we can compute the inner product value between each m template and query with Encode at once, along with the decoding procedure Decode to retrieve the inner product results. With additive homomorphic encryption, we can process the above computation with the encoded vector $x^\dagger$ encrypted for $\dagger \in \{+, -\}$, as described in $\mathsf{IDFace.IP}_{\mathsf{DB}}$. For more detailed information, we recommend the reader refer to Section 4.2.

We also provide the visualization of the application scenario of IDFace. Recall that IDFace enables securing a face identification system through encrypting the template database, without accompanying significant accuracy or efficiency degradation compared to the identification system without protection. For this reason, we expect that our IDFace can be used for any face identification system where the user physically presents his/her biometrics into the system, such as face identification systems for airports or entrances of the building. We remark that there already have been several real-world use cases, such as

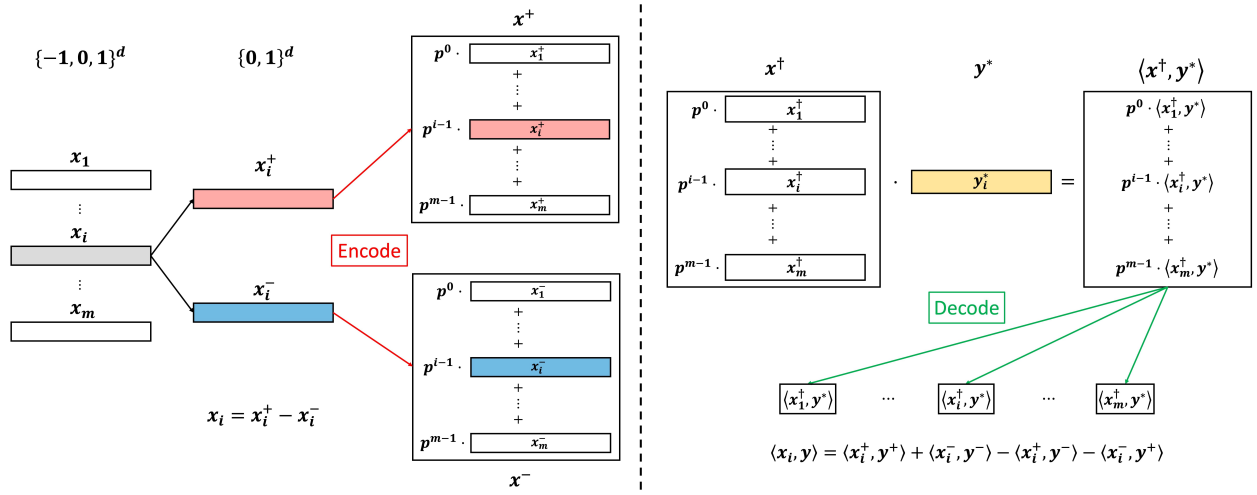


Figure 15. The overview of space efficient encoding technique. More details are provided in Section 4.2.

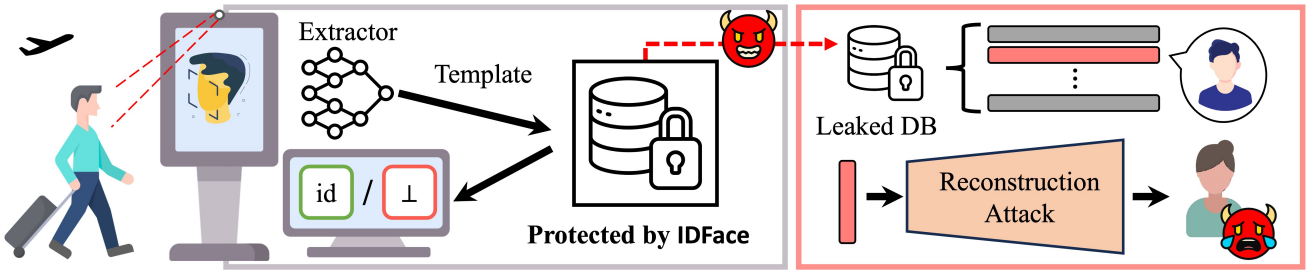


Figure 16. Target application scenario of IDFace.

several airports in Europe participating in Star Alliance Biometrics[3] and securing building entrances in several facilities, e.g., public schools in the United States[4,5] and casinos[6,7]. For more details about the application scenario and the threat model, we recommend the reader refer to Section 2.1 (biometric identification system) and Section 3.2 (biometric identification system with database encryption). In addition, the threat model we considered in IDFace is also described in Section 5.3.

References

[1] Pia Bauspieß, Jonas Olafsson, Jascha Kolberg, Pawel Drozdowski, Christian Rathgeb, and Christoph Busch. Improved homomorphically encrypted biometric identification using coefficient packing. In *2022 International Workshop on Biometrics and Forensics (IWBF)*, pages 1–6, 10.1109/IWBF55382.2022.9794523, 2022. IEEE. 13, 14, 18

[2] Seung bin Kim, Chan yeong Lim, Jungwoo Heo, Ju ho Kim, Hyun seo Shin, Kyo-Won Koo, and Ha-Jin Yu. Mr-rawnet: Speaker verification system with multiple temporal resolutions for variable duration utterances using raw waveforms. In *Interspeech 2024*, pages 2125–2129, 2024. 26

[3] Vishnu Naresh Boddeti. Secure face matching using fully homomorphic encryption. In *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–10. IEEE, 2018. 13, 14, 18

[4] Fadi Boutros. Elasticface. `https://github.com/fdbtrs/ElasticFace`. accessed: 2024-05-16. 12

[5] Fadi Boutros, Naser Damer, Florian Kirchbuchner, and Arjan Kuijper. Elasticface: Elastic margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1578–1587, 2022. 12

[6] Raffaele Cappelli, Dario Maio, Davide Maltoni, et al. Sfinge (synthetic fingerprint generator). 2004. 27

[3] https://www.staralliance.com/en/biometrics
[4] https://wvpublic.org/four-counties-to-implement-facial-recognition-for-school-safety/
[5] https://www.nytimes.com/2020/02/06/business/facial-recognition-schools.html
[6] https://www.econnectglobal.com/blog/how-integrated-resorts-and-casinos-are-leveraging-facial-recognition-software-for-increased-security
[7] https://journalrecord.com/2022/10/25/casino-uses-facial-recognition-technology-to-supplement-security/

[7] Xingbo Dong, Sangrae Cho, Youngsam Kim, Soohyung Kim, and Andrew Beng Jin Teoh. Deep rank hashing network for cancellable face identification. *Pattern Recognition*, 131:108886, 2022. 2

[8] Xingbo Dong, Hui Zhang, Yen Lung Lai, Zhe Jin, Junduan Huang, Wenxiong Kang, and Andrew Beng Jin Teoh. Wifakey: Generating cryptographic keys from face in the wild. *arXiv preprint arXiv:2407.14804*, 2024. 2

[9] P. Drozdowski, F. Struck, C. Rathgeb, and C. Busch. Benchmarking binarisation schemes for deep face templates. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 191–195, 2018. 19

[10] Joshua J Engelsma, Kai Cao, and Anil K Jain. Learning a fixed-length fingerprint representation. *IEEE transactions on pattern analysis and machine intelligence*, 43(6):1981–1997, 2019. 18, 27

[11] Joshua J Engelsma, Anil K Jain, and Vishnu Naresh Boddeti. Hers: Homomorphically encrypted representation search. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2022. 13, 14, 18

[12] Zekeriya Erkin, Martin Franz, Jorge Guajardo, Stefan Katzenbeisser, Inald Lagendijk, and Tomas Toft. Privacy-preserving face recognition. In *Privacy Enhancing Technologies: 9th International Symposium, PETS 2009, Seattle, WA, USA, August 5-7, 2009. Proceedings 9*, pages 235–253. Springer, 2009. 2

[13] David Evans, Yan Huang, Jonathan Katz, and Lior Malka. Efficient privacy-preserving biometric identification. In *Proceedings of the 17th conference Network and Distributed System Security Symposium, NDSS*, pages 90–98, 2011. 2

[14] Loubna Ghammam, Koray Karabina, Patrick Lacharme, and Kevin Thiry-Atighehchi. A cryptanalysis of two cancelable biometric schemes based on index-of-max hashing. *IEEE Transactions on Information Forensics and Security*, 15:2869–2880, 2020. 2

[15] IDEMIA Group. Mbss: Empowering world-class biometric identification, 2024. Accessed:2024-01-28. 23

[16] Jia Guo, Jiangkang Deng, Xiang An, Jack Yu, and Baris Gecer. Insightface: 2d and 3d face analysis project. https://github.com/deepinsight/insightface. accessed: 2024-05-16. 10

[17] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*, 2008. 11

[18] Hai Huang and Luyao Wang. Efficient privacy-preserving face identification protocol. *IEEE Transactions on Services Computing*, pages 1–10, 2023. 2, 18

[19] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Quantized neural networks: Training neural networks with low precision weights and activations. *Journal of Machine Learning Research*, 18(187):1–30, 2018. 18, 19

[20] Jong-Hyuk Im, Seong-Yun Jeon, and Mun-Kyu Lee. Practical privacy-preserving face authentication for smartphones secure against malicious clients. *IEEE Transactions on Information Forensics and Security*, 15:2386–2401, 2020. 2

[21] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613, 1998. 2

[22] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1):117–128, 2010. 19

[23] Yubing Jiang, Peisong Shen, Li Zeng, Xiaojie Zhu, Di Jiang, and Chi Chen. Cancelable biometric schemes for euclidean metric and cosine metric. *Cybersecurity*, 6(1):1–20, 2023. 2

[24] Andrew Teoh Beng Jin, David Ngo Chek Ling, and Alwyn Goh. Biohashing: two factor authentication featuring fingerprint data and tokenised random number. *Pattern recognition*, 37(11):2245–2255, 2004. 2

[25] Shuaifan Jin, He Wang, Zhibo Wang, Feng Xiao, Jiahui Hu, Yuan He, Wenwen Zhang, Zhongjie Ba, Weijie Fang, Shuhong Yuan, et al. {FaceObfuscator}: Defending deep learning-based privacy attacks with gradient descent-resistant features in face recognition. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 6849–6866, 2024. 2

[26] Zhe Jin, Jung Yeon Hwang, Yen-Lung Lai, Soohyung Kim, and Andrew Beng Jin Teoh. Ranking-based locality sensitive hashing-enabled cancelable biometrics: Index-of-max hashing. *IEEE Transactions on Information Forensics and Security*, 13(2):393–407, 2017. 2

[27] Arun Kumar Jindal, Srinivasa Rao Chalamala, and Santosh Kumar Jami. Securing face templates using deep convolutional neural network and random projection. In *2019 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–6. IEEE, 2019. 2

[28] Ari Juels and Martin Wattenberg. A fuzzy commitment scheme. In *Proceedings of the 6th ACM conference on Computer and communications security*, pages 28–36, 1999. 2

[29] Minchul Kim. Cvlface: High-performance face recognition all-in-one toolkit. accessed: 2024-07-01. 13

[30] Minchul Kim, Anil K Jain, and Xiaoming Liu. Adaface: Quality adaptive margin for face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18750–18759, 2022. 12, 17

[31] Minchul Kim, Yiyang Su, Feng Liu, Anil Jain, and Xiaoming Liu. Keypoint relative position encoding for face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 244–255, 2024. 12

[32] Sunpill Kim, Yunseong Jeong, Jinsu Kim, Jungkon Kim, Hyung Tae Lee, and Jae Hong Seo. Ironmask: Modular architecture for protecting deep face template. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16125–16134, 2021. 2

[33] Ron Kohavi and Mehran Sahami. Error-based and entropy-based discretization of continuous features. In *KDD*, pages 114–119, 1996. 19

[34] Arun Kumar Jindal, Srinivas Chalamala, and Santosh Kumar Jami. Face template protection using deep convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 462–470, 2018. 2

[35] Yenlung Lai, Zhe Jin, KokSheik Wong, and Massimo Tistarelli. Efficient known-sample attack for distance-preserving hashing biometric template protection schemes. *IEEE Transactions on Information Forensics and Security*, 16:3170–3185, 2021. 2

[36] Jing Lei, Qingqi Pei, Yao Wang, Wenhai Sun, and Xuefeng Liu. Privface: fast privacy-preserving face authentication with revocable and reusable biometric credentials. *IEEE Transactions on Dependable and Secure Computing*, 19(5):3101–3112, 2021. 2

[37] Yue Li, Wenrui Ding, Chunlei Liu, Baochang Zhang, and Guodong Guo. Trq: Ternary neural networks with residual quantization. In *Proceedings of the AAAI conference on artificial intelligence*, pages 8538–8546, 2021. 18, 19

[38] Yuxing Li, Liaojun Pang, Heng Zhao, Zhicheng Cao, Eryun Liu, and Jie Tian. Indexing-min–max hashing: Relaxing the security–performance tradeoff for cancelable fingerprint templates. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(10): 6314–6325, 2022. 2

[39] Meng-Hui Lim, Andrew Beng Jin Teoh, and Kar-Ann Toh. An analysis on equal width quantization and linearly separable subcode encoding-based discretization and its performance resemblances. *EURASIP Journal on Advances in Signal Processing*, 2011:1–14, 2011. 19

[40] Weiyang Liu and Yandong Wen. Opensphere. https://github.com/ydwen/opensphere. accessed: 2024-05-16. 12

[41] Shuming Ma, Hongyu Wang, Lingxiao Ma, Lei Wang, Wenhui Wang, Shaohan Huang, Li Dong, Ruiping Wang, Jilong Xue, and Furu Wei. The era of 1-bit llms: All large language models are in 1.58 bits. *arXiv preprint arXiv:2402.17764*, 2024. 18, 19

[42] Yukun Ma, Lifang Wu, Xiaofeng Gu, Jiaoyu He, and Zhou Yang. A secure face-verification scheme based on homomorphic encryption and deep neural networks. *IEEE Access*, 5:16532–16538, 2017. 18

[43] Guangcan Mai, Kai Cao, Xiangyuan Lan, and Pong C Yuen. Secureface: Face template protection. *IEEE Transactions on Information Forensics and Security*, 16:262–277, 2020. 2

[44] Dario Maio, Davide Maltoni, Raffaele Cappelli, Jim L Wayman, and Anil K Jain. Fvc2004: Third fingerprint verification competition. In *International conference on biometric authentication*, pages 1–7. Springer, 2004. 27

[45] Brianna Maze, Jocelyn Adams, James A Duncan, Nathan Kalka, Tim Miller, Charles Otto, Anil K Jain, W Tyler Niggel, Janet Anderson, Jordan Cheney, et al. Iarpa janus benchmark-c: Face dataset and protocol. In *2018 international conference on biometrics (ICB)*, pages 158–165. IEEE, 2018. 11

[46] Qiang Meng and Feng Zhou. Towards privacy-preserving, real-time and lossless feature matching. *arXiv preprint arXiv:2208.00214*, 2022. 18

[47] Quang Meng, Torsten Schlett, Kaiyu Yue, and Martin Knoche. Magface. https://github.com/IrvingMeng/MagFace. accessed: 2024-05-16. 12

[48] Qiang Meng, Shichao Zhao, Zhida Huang, and Feng Zhou. Magface: A universal representation for face recognition and quality assessment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14225–14234, 2021. 12

[49] Yuxi Mi, Yuge Huang, Jiazhen Ji, Minyi Zhao, Jiaxiang Wu, Xingkun Xu, Shouhong Ding, and Shuigeng Zhou. Privacy-preserving face recognition using random frequency components. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19673–19684, 2023. 2

[50] Yuxi Mi, Zhizhou Zhong, Yuge Huang, Jiazhen Ji, Jianqing Xu, Jun Wang, Shaoming Wang, Shouhong Ding, and Shuigeng Zhou. Privacy-preserving face recognition using trainable feature subtraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 297–307, 2024. 2

[51] Deen Dayal Mohan, Nishant Sankaran, Sergey Tulyakov, Srirangaraj Setlur, and Venu Govindaraju. Significant feature based representation for template protection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 2389–2396, 2019. 2

[52] Stylianos Moschoglou, Athanasios Papaioannou, Christos Sagonas, Jiankang Deng, Irene Kotsia, and Stefanos Zafeiriou. Agedb: the first manually collected, in-the-wild age database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 51–59, 2017. 11

[53] Frederick Mosteller. On Some Useful "Inefficient" Statistics. *The Annals of Mathematical Statistics*, 17(4):377 – 408, 1946. 5, 6

[54] Mervin E Muller. A note on a method for generating points uniformly on n-dimensional spheres. *Communications of the ACM*, 2(4): 19–20, 1959. 4

[55] Arsha Nagrani, Joon Son Chung, Weidi Xie, and Andrew Zisserman. Voxceleb: Large-scale speaker verification in the wild. *Computer Speech & Language*, 60:101027, 2020. 26

[56] NIST. Nist 2024 speark recognition evaluation (sre24). accessed: 2024-10-21. 26

[57] Dailé Osorio-Roig, Christian Rathgeb, Pawel Drozdowski, and Christoph Busch. Stable hash generation for efficient privacy-preserving face identification. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 4(3):333–348, 2022. 18

[58] Seunghun Paik, Sunpill Kim, and Jae Hong Seo. Security analysis on locality-sensitive hashing-based biometric template protection schemes. In *34th British Machine Vision Conference 2023, BMVC 2023, Aberdeen, UK, November 20-24, 2023*. BMVA, 2023. 2

[59] Tim Rohwedder, Dailé Osorio-Roig, Christian Rathgeb, and Christoph Busch. Benchmarking fixed-length fingerprint representations across different embedding sizes and sensor types. In *2023 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–6. IEEE, 2023. 27

[60] Soumyadip Sengupta, Jun-Cheng Chen, Carlos Castillo, Vishal M Patel, Rama Chellappa, and David W Jacobs. Frontal to profile face verification in the wild. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9. IEEE, 2016. 11

[61] Leiming Su and Shizuo Sakamoto. Extremely-large-scale biometric authentication system-its practical implementation. *NEC Technical Journal*, 7(2):57, 2012. 23

[62] Veeru Talreja, Matthew C Valenti, and Nasser M Nasrabadi. Zero-shot deep hashing and neural network based error correction for face template protection. In *2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–10. IEEE, 2019. 2

[63] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17 (3):261–272, 2020. 3

[64] Hanrui Wang, Xingbo Dong, Zhe Jin, Andrew Beng Jin Teoh, and Massimo Tistarelli. Interpretable security analysis of cancellable biometrics using constrained-optimized similarity-based attack. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 70–77, 2021. 2

[65] Zhibo Wang, He Wang, Shuaifan Jin, Wenwen Zhang, Jiahui Hu, Yan Wang, Peng Sun, Wei Yuan, Kaixin Liu, and Kui Ren. Privacy-preserving adversarial facial features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8212–8221, 2023. 2

[66] Yandong Wen, Weiyang Liu, Adrian Weller, Bhiksha Raj, and Rita Singh. Sphereface2: Binary classification is all you need for deep face recognition. In *International Conference on Learning Representations*, 2022. 12

[67] Jee weon Jung, Youjin Kim, Hee-Soo Heo, Bong-Jin Lee, Youngki Kwon, and Joon Son Chung. Pushing the limits of raw waveform speaker recognition. In *Interspeech 2022*, pages 2228–2232, 2022. 26

[68] Ivan Yakovlev, Rostislav Makarov, Andrei Balykin, Pavel Malov, Anton Okhotnikov, and Nikita Torgashov. Reshape dimensions network for speaker recognition. In *Interspeech 2024*, pages 3235–3239, 2024. 26

[69] Chenzhuo Zhu, Song Han, Huizi Mao, and William J Dally. Trained ternary quantization. In *International Conference on Learning Representations*, 2017. 18, 19

[70] Zheng Zhu, Guan Huang, Jiankang Deng, Yun Ye, Junjie Huang, Xinze Chen, Jiagang Zhu, Tian Yang, Jiwen Lu, Dalong Du, et al. Webface260m: A benchmark unveiling the power of million-scale deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10492–10502, 2021. 13