A More Analyses

A.1 Evaluation of *Whitebox* and *Blackbox* Attacks at FMR$= 10^{-2}$

Table 7 and Table 8 of this appendix report the evaluation of attacks with *whitebox* and *blackbox* knowledge, respectively, of the system from which the template is leaked (i.e., $F_{\text{loss}} = F_{\text{database}}$) against SOTA FR models at **FMR$= 10^{-2}$** in terms of adversary's success attack rate (SAR) using our proposed method on the MOBIO and LFW datasets. As the results in these tables show, our method outperforms previous methods in the literature.

Table 7: Evaluation of attacks with *whitebox* knowledge of the system from which the template is leaked (i.e., $F_{\text{loss}} = F_{\text{database}}$) against SOTA FR models in terms of adversary's success attack rate (SAR) using our proposed method on the MOBIO and LFW datasets. The values are in percentage and correspond to the threshold where the target system has **FMR$= 10^{-2}$**. Cells are color coded according the type of attack as defined in Section 2 of the paper for attack 1 (light gray) and attack 2 (dark gray).

F_{database}	MOBIO					LFW				
	ArcFace	**ElasticFace**	**HRNet**	**AttentionNet**	**Swin**	**ArcFace**	**ElasticFace**	**HRNet**	**AttentionNet**	**Swin**
ArcFace	100.00	93.81	80.00	81.90	85.24	93.64	90.89	68.08	62.75	76.24
ElasticFace	90.95	93.33	78.57	83.81	84.29	87.88	92.80	71.82	64.24	75.70

Table 8: Evaluation of attacks (with *blackbox* knowledge of the system from which the template is leaked i.e., F_{database}) against SOTA FR models in terms of adversary's success attack rate (SAR) using different methods on the MOBIO and LFW datasets. The values are in percentage and correspond to the threshold where the target system has **FMR$= 10^{-2}$**. **M1**: NbNetB-M [Mai et al., 2018], **M2**: NbNetB-P [Mai et al., 2018], **M3**: [Dong et al., 2021], **M4**: [Vendrow and Vendrow, 2021], and **M5**: [Dong et al., 2023]. Cells are color coded according the type of attack as defined in Section 2 of the paper for attack 3 (lightest gray), attack 4 (middle dark gray), and attack 5 (darkest gray).

F_{database}	F_{loss}	F_{target}	MOBIO						LFW					
			M1	**M2**	**M3**	**M4**	**M5**	**Ours**	**M1**	**M2**	**M3**	**M4**	**M5**	**Ours**
ArcFace	**ElasticFace**	**ArcFace**	26.67	49.05	20.48	67.14	85.71	**89.52**	26.66	61.66	28.31	76.98	87.25	**87.85**
		ElasticFace	11.90	49.52	16.19	34.29	60.95	**86.67**	32.42	66.61	23.05	57.84	74.31	**87.43**
		HRNet	10.48	24.76	10.00	26.19	54.28	**79.05**	18.69	43.21	17.37	33.55	50.22	**60.93**
		AttentionNet	11.43	38.10	18.10	24.29	54.76	**80.48**	10.84	31.88	13.31	26.73	44.99	**53.86**
		Swin	10.48	45.24	10.95	29.52	58.09	**82.86**	14.79	45.80	16.98	38.03	57.71	**67.80**
ElasticFace	**ArcFace**	**ArcFace**	17.14	49.05	20.95	47.14	79.91	**95.24**	33.08	67.89	26.35	57.48	73.80	**91.23**
		ElasticFace	30.00	70.95	25.7	75.24	88.80	**94.76**	52.99	81.74	33.53	79.62	88.80	**93.34**
		HRNet	8.10	47.14	15.24	31.43	67.14	**83.81**	29.27	60.34	23.22	39.06	62.01	**76.68**
		AttentionNet	12.86	47.14	23.43	40.95	66.19	**87.14**	18.53	46.36	17.78	31.53	55.29	**69.45**
		Swin	10.00	54.76	13.81	37.14	68.57	**89.05**	24.50	60.19	21.40	41.13	65.82	**80.15**

A.2 Ablation Study

Ablation Study on the Effect of Feature Extractor in the ID loss To evaluate the effect of feature extractor in our loss function, we consider attack 3 on HRNet templates and use ArcFace and ElasticFace for $F_{\text{loss}}(.)$ in our loss function. Table 9 of this appendix reports the result of this ablation study. Comparing the results of different face recognition models used as $F_{\text{loss}}(.)$ in our loss function, we can see that the mapping which is trained using ArcFace achieves a higher SAR that the mapping that is trained with ElasticFace.

Table 9: Evaluating the effect of ID loss term in our loss function in attack 3 against HRNet in terms of SAR in the system with FMRs of 10^{-2} and 10^{-3} evaluated on the MOBIO and LFW datasets. The values are in percentage.

F_{loss} in ID loss	MOBIO		LFW	
	FMR=10^{-2}	**FMR=10^{-3}**	**FMR=10^{-2}**	**FMR=10^{-3}**
ArcFace	**91.90**	**86.19**	**76.01**	**48.22**
ElasticFace	86.71	82.38	72.59	43.71

Moreover, comparing these results with the recognition performances of ArcFace and ElasticFace reported in Table 2 of the paper, we can conclude that a face recognition method with a higher recognition performance can lead to a better reconstruction when used as F_{loss} in the blackbox attack using our proposed method.

Ablation Study on the Effect of Noise in our WGAN Training To evaluate the effect of noise used in our GAN training, we implement another ablation with the same configuration used for our ablation study in the paper (i.e., attack 1 against ArcFace), and we train two networks with and without noise in the input of the mapping network. Table 10 of this appendix reports the result of our ablation study. As this table shows, using noise in our WGAN training improves the performance of our face reconstruction method.

Table 10: Evaluating the effect of using noise in our method in attack 1 against ArcFace in terms of SAR in the system with FMRs of 10^{-2} and 10^{-3} evaluated on the MOBIO and LFW datasets. The values are in percentage.

	MOBIO		LFW	
	FMR=10^{-2}	FMR=10^{-3}	FMR=10^{-2}	FMR=10^{-3}
with noise	**100.00**	**92.38**	**93.64**	**86.82**
without noise	97.14	74.76	89.19	77.72

It is noteworthy that generally, in training GANs (even in conditional GANs) a noise (e.g., from Gaussian distribution) is used in the input of the generator network. The samples of noise in the input help the generator to learn the distribution of the output space, and therefore help the generator network to generate outputs from the same distribution of real data. The discriminator (or critic in WGAN) network tries to distinguish if the sample output is from the distribution of real data or not. In other words, adding random noise in the input makes the training stochastic which is suitable for learning a distribution. In our problem, it is very important that the generated latent code is from the same distribution as the intermediate latent space $\mathcal{W}$ of StyleGAN. In particular, if the generated latent code is not in the same distribution of $\mathcal{W}$, it can easily lead to a non-face-like image at the output of StyleGAN.

Ablation Study on the Mapping Space To evaluate the effect of the mapping space in our proposed method, we consider attack 1 on against ArcFace model, and train mapping to input latent space $\mathcal{Z}$ and the *intermediate* latent space $\mathcal{W}$ of StyleGAN. Table 11 of this appendix reports the result of our ablation study.

Table 11: Evaluating the effect of mapping space in our method in attack 1 against ArcFace in terms of SAR in the system with FMRs of 10^{-2} and 10^{-3} evaluated on the MOBIO and LFW datasets. The values are in percentage.

Mapping Space	MOBIO		LFW	
	FMR=10^{-2}	FMR=10^{-3}	FMR=10^{-2}	FMR=10^{-3}
$\mathcal{W}$	**100.00**	**92.38**	**93.64**	**86.82**
$\mathcal{Z}$	71.42	41.42	75.94	57.18

As the results in this table show, mapping to the *intermediate* latent space $\mathcal{W}$ leads to a higher performance. This is because the *intermediate* latent space has more information and is more controllable than input space $\mathcal{Z}$, which is originally of Gaussian distribution for noise in StyleGAN. This ablation study highlights the importance of mapping to the *intermediate* latent space $\mathcal{W}$ of StyleGAN, which has not been proposed in the literature for template inversion.

A.3 Using a Different Face Generator Network

In our experiments, we used StyleGAN which is one of the most popular face generator models in the literature. However, our method can also be used with other face generator networks. As another experiment, we use StyleSwin [Zhang et al., 2022], which is another face generator model based on transformers. Figure 7 of this appendix shows the reconstructed face images from ArcFace templates using StyleSwin in our method instead of StyleGAN. We used a similar mapping network and learned a mapping from facial templates to the intermediate latent space of StyleSwin. As these results show, our method can also be used with other face generator networks.

A.4 Application of Our Method for Face Recognition Models with Different Inputs/Outputs

While we use three different face recognition models in our problem formulation, since these models are applied in separate stages, there is no issue if the inputs and outputs (e.g., pre-processing steps or

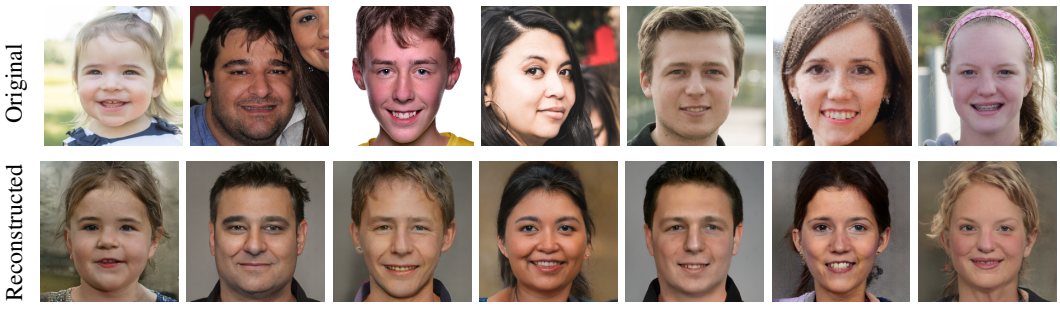

Figure 7: Sample face images from the FFHQ dataset and their corresponding reconstructed images from ArcFace templates using our template inversion method with **StyleSwin** [Zhang et al., 2022] as the face generator model. The values below each image show the cosine similarity between the corresponding templates of original and reconstructed face images.

dimensions) be different in each of these face recognition models. For differences in inputs (face images), because each of these models is applied independently on the given face image, the required pre-processing can be considered within the function of the face recognition model in our problem formulation. For differences in outputs (face templates), since the facial templates extracted by each model are compared to facial templates extracted by the same model, there is no conflict in the dimensions. The only point to be noted is that the input of our mapping network should have the same dimension as the templates of .

Let us consider the complete pipeline of our problem formulation as depicted in Figure 2 of the paper. The first face recognition model (i.e., F_{database}) uses its own pre-process and extracts facial templates from face images captured by the camera of the face recognition system (from which the template is leaked). These facial templates (extracted from F_{database}) are then used as input to our face reconstruction model. Therefore, the input of our mapping should have the same dimension as templates of F_{database}. In any case, the output of the face reconstruction network is a high-resolution (1024×1024) face image, regardless of the dimension of the input facial template. During training, the generated high-resolution face image is first pre-processed as required by F_{loss} (i.e., normalised, resized and aligned based on coordinates required by F_{loss}), and the extracted templates are compared with templates of the original image extracted from F_{loss} (with the required pre-processing for F_{loss}). During inference (i.e., attacking the target FR system), however, the generated high-resolution face image is pre-processed as required by F_{target}. Therefore, there is no conflict in the inputs/outputs in our pipeline.

In our experiments reported in the paper, all face recognition models except Swin take input with 112×112 resolution. However, the Swin model takes input with 224×224 resolution. The dimensions of facial templates extracted by all other face recognition models (in Table 3 of the paper) in our experiments are similar and equal to 512. To show that our method can also be used in case of different dimensions of facial templates and to showcase another face recognition model with different pre-processing, as a new experiment, we use a new model, VGGFace [Parkhi et al., 2015], with a different dimension of facial templates (2048-dimension) and different input image resolution (224×224) which has a different normalization as well as different landmark coordinates for face alignment. We use ArcFace as our F_{loss} and evaluate the reconstructed face images in attacks against different face recognition systems (as F_{target}) on the LFW dataset. The results in Table 12 of this appendix show that our proposed method can be applied in the case

Table 12: Evaluation of success attack rate for TI attack using VGGFace templates (as F_{database}) using ArcFace as F_{loss} in attack against FR systems with different models (as F_{target}). Note that pre-processing (normalization and alignment coordinates) of VGGFace is different than all target models and its input resolution is 224×224. The input resolution for ArcFace (used as F_{loss}) and ElasticFace is 112×112 but for Swin is 224×224. The templates extracted by VGGFace has 2048 dimensions, while templates of ArcFace, ElasticFace, and VGGFace have 512 dimension.

	ArcFace	**ElasticFace**	**Swin**
FMR $= 10^{-2}$	92.92	93.10	83.97
FMR $= 10^{-3}$	86.61	82.39	72.89

where the inputs/outputs of face recognition models in our problem formulation (F_{database}, F_{loss}, and F_{target}) are different, and still achieves high success attack rates against face recognition systems with different inputs/outputs.

B Ethics Statement

Motivations The proposed face reconstruction method is presented with the motivation of showing vulnerability of face recognition systems to template inversion attacks. We hope this work encourages researchers of the community to investigate the next generation of safe and robust face recognition systems and to develop new algorithms to protect existing systems. In addition, we should note that the project on which the work has been conducted has passed an Institutional Ethical Review Board (IRB).

Ethics Considerations While the proposed method might pose a social threat against unprotected systems, we do not condone using our work with the intent of attacking a *real* face recognition system or other malicious purposes. We should, however, note that for the next generation of safe face recognition systems, *any kind of potential attacks* should be completely studied by the researchers; and then based upon such vulnerability studies, new protection and defense algorithms will be proposed by the research community in the future. To facilitate future studies, we publish source code of our work as described in Section C of this appendix.

Mitigation of such Attacks This paper demonstrates an important privacy and security threat to the state-of-the-art unprotected face recognition systems. Along the same lines, data protection frameworks, such as the European Union General Data Protection Regulation (EU-GDPR) [European Council, 2016], put legal obligations to protect biometric data as sensitive information. To this end and to prevent such attacks to face recognition systems, several biometric template protection algorithms are proposed in the literature [Nandakumar and Jain, 2015, Sandhya and Prasad, 2017, Kaur et al., 2022, Kumar et al., 2020, Shahreza et al., 2022, 2023].

C Reproducibility Statement

In our experiments, we use PyTorch package and the pre-trained model of StyleGAN3[8] and StyleSwin[9] to generate high-resolution face images. We train our mapping network for 16 epochs with an initial learning rate of 0.1 using Adam optimizer [Kingma and Ba, 2015] and divide the learning rate by 2 every three epochs. Training our mapping network using our proposed method takes around two days on a system equipped with an NVIDIA GeForce RTX$^{\text{TM}}$ 3090. We build face recognition pipelines using Bob [Anjos et al., 2012, 2017] toolbox[10]. The source code of our experiments is publicly available[11] to help reproduce our results.

D Licenses and Copyright Permissions

Datasets We have signed the licenses (GDPR compliance) to use from the data controller of any of the datasets used in this paper (i.e., MOBIO, LFW, and FFHQ) and followed the terms of use of these datasets in this paper. We have also cited the corresponding paper for each dataset.

Models We used pretrained models of following deep neural networks and followed the license of each one in implementing our experiments:

- ArcFace, ElasticFace, and VGGFace face recognition models implemented in Bob [Anjos et al., 2012, 2017] toolbox (under BSD 3-Clause License)
- HRNet, AttentionNet, and Swin face recognition models implemented in FaceX-Zoo [Wang et al., 2021] toolbox (under Apache License, Version 2.0)

[8] Available at `https://github.com/NVlabs/stylegan3`
[9] Available at `https://github.com/microsoft/StyleSwin`
[10] Available at `https://www.idiap.ch/software/bob/`
[11] Available at `https://gitlab.idiap.ch/bob/bob.paper.neurips2023_face_ti`

- StyleGAN3 (official) model published under Nvidia Source Code License[12].
- StyleSwin (official) model published under MIT License.

[12] Available at https://github.com/NVlabs/stylegan3/blob/main/LICENSE.txt