HAL
archives-ouvertes.fr

A Decoupled 3D Facial Shape Model by Adversarial Training

Victoria Fernández Abrevaya, Adnane Boukhayma, Stefanie Wuhrer, Edmond Boyer

▶ **To cite this version:**

HAL Id: hal-02064711

https://hal.archives-ouvertes.fr/hal-02064711v3

Submitted on 7 Sep 2019

A Decoupled 3D Facial Shape Model by Adversarial Training

Victoria Fernández Abrevaya[1], Adnane Boukhayma[2], Stefanie Wuhrer[1], Edmond Boyer[1]

[1] Inria - Univ. Grenoble Alpes - CNRS - LJK, France

`{victoria.fernandez-abrevaya, stefanie.wuhrer, edmond.boyer}@inria.fr`

[2] University of Oxford, UK

`adnane.boukhayma@eng.ox.ac.uk`

Abstract

Data-driven generative 3D face models are used to compactly encode facial shape data into meaningful parametric representations. A desirable property of these models is their ability to effectively decouple natural sources of variation, in particular identity and expression. While factorized representations have been proposed for that purpose, they are still limited in the variability they can capture and may present modeling artifacts when applied to tasks such as expression transfer. In this work, we explore a new direction with Generative Adversarial Networks and show that they contribute to better face modeling performances, especially in decoupling natural factors, while also achieving more diverse samples. To train the model we introduce a novel architecture that combines a 3D generator with a 2D discriminator that leverages conventional CNNs, where the two components are bridged by a geometry mapping layer. We further present a training scheme, based on auxiliary classifiers, to explicitly disentangle identity and expression attributes. Through quantitative and qualitative results on standard face datasets, we illustrate the benefits of our model and demonstrate that it outperforms competing state of the art methods in terms of decoupling and diversity.

1. Introduction

Generative models of 3D shapes are widely used for their ability to provide compact representations that allow to synthesize realistic shapes and their variations according to natural factors. This is particularly true with faces whose 3D shape spans a low dimensional space, and for which generative models often serve as strong priors to solve underconstrained problems such as reconstruction from partial data. Given that the facial shape presents natural factors of variation (*e.g.* identity and expression), modeling these in a *decoupled* manner is an important aspect, as it allows to incorporate semantic control when performing inference or synthesis tasks. Having interpretable representations in terms of pre-defined factors of variation opens the door to several applications, such as 3D face animation [41, 21], accurate expression transfer [22, 38], recognition [1] and artifical data synthesis [33].

Since the seminal work of Blanz and Vetter [4], numerous approaches have been proposed to build data-driven generative models of the 3D face. Most commonly, variations among different identities are modeled by linear shape statistics such as PCA [4, 5]. When expressions need to be considered the identity and expression subpaces are typically modeled as two independent linear factors which are additively combined [1]. In practice this can produce artifacts when transferring expressions among very different facial shapes, an issue that has to be explicitly accounted for, *e.g.* [38]. Multilinear models [40, 8, 13] present relative improvements by considering a tensor decomposition combining the two spaces, but training requires a complete labeled data tensor which is very hard to get in practice, and transferring expressions by simply switching the latent coefficients can still present artifacts [17].

With the aim to relax the linear assumption in modeling 3D faces, deep generative models with autoencoder architectures have recently been proposed. They demonstrate benefits in modeling geometric details [3], non-linear deformations present in facial expressions [29], and increasing robustness to different types of capture noise [13]. Yet, none of these approaches decouple the factors of variation with the exception of [13], where an initialization with fully labeled data is required whose size increases exponentially in the number of considered factors.

In this work we investigate the use of Generative Adversarial Networks (GANs) [16] for 3D face modeling and provide insights on their ability to learn decoupled representations. In particular, our comparisons with recent approaches based on autoencoder architectures [13, 29] demonstrate that our proposed approach can better decouple identity and expression, and exhibit more variability in the generated data.

While current deep learning techniques have shown im-

pressive results in the image domain, extending these to 3D data is not straightforward. We propose a novel 3D-2D architecture in which a multilayer perceptron generates a 3D face shape given a latent code, while a regular convolutional network is used as a 2D discriminator. This is allowed by an intermediate *geometry mapping* layer that transforms a 3D surface mesh into a geometry image encoding the mesh vertex locations. To effectively decouple the factors of variation we build on auxiliary classifiers [27] that aim to correctly guess the label associated with each factor, *e.g.* identity and expression, and introduce a loss on the classifier features for unlabeled samples.

To summarize, our contributions are:

1. A generative 3D face model that captures non-linear deformations due to expression, as well as the relationship between identity and expression subspaces.

2. A novel 3D-2D architecture that allows to generate 3D meshes while leveraging the discriminative power of CNNs, by introducing a *geometry mapping layer* that acts as bridge between the two domains.

3. A training scheme that enables to effectively decouple the factors of variation, leading to significant improvements with respect to the state of the art.

2. Related Work

Due to the importance of 3D face modeling for numerous applications, many works have been proposed to learn generative models. We focus here on data-driven approaches, often called *3D Morphable Models* (3DMM) in the literature. Blanz and Vetter [4] use principal component analysis (PCA) to learn the distribution of the facial shape and appearance across different identities scanned in a neutral expression. To handle other expressions, subsequent works model them by either adding linear factors [1] or by extending PCA to a multilinear model [40]. Thanks to their simple structure these models are still heavily used, and have recently been extended by training from large datasets [5], modeling geometric details [26, 6, 7], and including other variations such as skeletal rotations [25].

Autoencoders for 3D Faces Recent works leverage deep learning methods to overcome the limitations of (multi-)linear models. Ranjan *et al.* [29] proposed an autoencoder architecture that learns a single global model of the 3D face, and as such the different factors cannot be decoupled directly. However, an extension called DeepFLAME is proposed that combines a linear model of identity [25] with the autoencoder trained on expression displacements. While expressions are modeled non-linearly, the relationship between identity and expression is not addressed explicitly. Fernández Abrevaya *et al.* [13] developed a multilinear autoencoder (MAE) in which the decoder is a multilinear ten-

sor structure. While the relationship between the two spaces is accounted for, transferring expressions still presents artifacts. Furthermore, to achieve convergence the tensor needs to be initialized properly, which implies that the size of labeled training data needed for initialization increases exponentially in the number of factors considered. We compare our proposed approach to DeepFLAME and MAE, as they achieve state-of-the-art results on decoupling identity and expression variations.

Bagautdinov *et al.* [3] propose a multiscale model of 3D faces at different levels of geometric detail. Two recent works [39, 37] use autoencoders to learn a global or corrective morphable model of 3D faces and their appearance based on 2D training data. However, none of these methods allow to disentangle factors of variation in the latent space. Unlike the aforementioned works, we investigate the use of GANs to learn a decoupled model of 3D faces.

GANs for 3D faces Some recent works have proposed to combine a 3DMM with an *appearance* model obtained by adversarial learning. Slossberg *et al.* [35] train a GAN on aligned facial textures and combine this with a linear 3DMM to generate realistic synthetic data. Gecer *et al.* [15] train a similar model and show that GANs can be used as a texture prior for accurate fitting to 2D images. Deng *et al.* [10] fit a 3DMM to images and use a GAN to complete the missing parts of the resulting UV map. All of these methods rely on linear 3DMMs, and hence to shape spaces limited in expressiveness. While the focus is on improving the appearance, we follow a different objective with a generative shape model that decouples identities and expressions.

To the best of our knowledge, the only work that learns 3D facial *shape* variations using a GAN is [33], which is an extension of [35]. The authors propose to learn identity variations by training a GAN on geometry images, but unlike our work they do not model the non-linear variations due to expression nor the correlation between identity and expression, since the main focus is on the appearance.

Two other methods learn to enhance an input 3D face geometry with photometric information using a GAN. Given a texture map and a coarse mesh, Huynh *et al.* [20] augment the latter with fine scale details, and given an input image and a base mesh, Yamaguchi *et al.* [42] infer detailed geometry and high quality reflectance. Both works require the conditioning of an input, and unlike us they do not build a generative 3D face model.

3. Background

Generative Adversarial Networks [16] are based on a minimax game, in which a discriminator D and a generator G are optimized for competing goals. The discriminator is tasked with learning the difference between real and fake samples, while the generator is trained to maximize the mis-
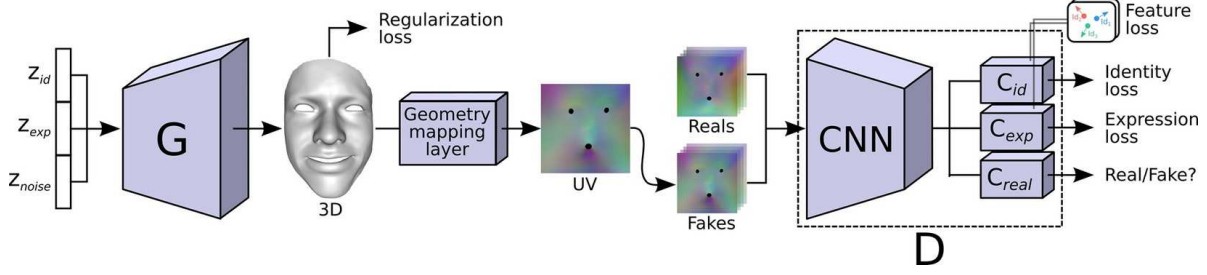
Figure 1: Our proposed architecture. A MLP generates the 3D coordinates of the mesh, while the discrimination occurs in 2D space thanks to the *geometry mapping* layer. Identity and expression codes z_{id}, z_{exp} are used to control the generator, and classification losses are added to decouple between the two. A feature loss is introduced to ensure consistency over features with fixed identities or expressions.

takes of the discriminator. At convergence, G approximates the real data distribution. Training involves the optimization of the following:

$$\min_{G} \max_{D} \mathcal{L}_{GAN} = \mathbb{E}_{x \sim p_{data}}[\log D(x)]$$
$$+ \mathbb{E}_{z \sim p_z}[\log(1 - D(G(z)))], \quad (1)$$

where p_{data} denotes the distribution of the training set, and p_z denotes the prior distribution for G, typically $\mathcal{N}(0, I)$.

GANs have been shown to be very challenging to train with the original formulation and prone to low diversity in the generated samples. To address this, Arjovsky *et al.* [2] propose to minimize instead an approximation of the Earth Mover's distance between generated and real data distributions, which is the strategy we adopt in this work:

$$\mathcal{L}_{GAN} = \mathbb{E}_{x \sim p_{data}}[D(x)] - \mathbb{E}_{z \sim p_z}[D(G(z))]. \quad (2)$$

In particular we use the extension in [19] which uses a gradient penalty in order to enforce that D is 1-Lipschitz.

When labels are available, using them has proven to be beneficial for GAN performance. Odena *et al.* [27] proposed Auxiliary Classifier GANs (AC-GAN), in which D is augmented so that it outputs the probability of an image belonging to a pre-defined class label $c \sim p_c$. In this case, the loss function for G and D is extended with:

$$\mathcal{L}_C^{real} = \mathbb{E}_{x \sim p_{data}, c \sim p_c}[\log P(C = c|x)], \quad (3)$$

$$\mathcal{L}_C^{fake} = \mathbb{E}_{z \sim p_z, c \sim p_c}[\log P(C = c|G(z, c))]. \quad (4)$$

In order to evaluate if a model is correctly decoupling, we need to be able to distinguish whether two identites or expressions sharing the same latent code are perceptually similar. Thus, our work builds on the idea of auxiliary classifiers in order to learn a decoupling of the shape variations into factors, as will be explained in the next section.

4. Method

We consider as input a dataset of registered and rigidly aligned 3D facial meshes, where each mesh is defined by $(\mathcal{V}, \mathcal{F})$, the set of 3D vertices $\mathcal{V} \in \mathbb{R}^{3 \times n_v}$ and the set of triangular faces $\mathcal{F} \in \mathbb{N}^{3 \times n_f}$ that connect the vertices. Our goal is to build an expressive model that can decouple the representation based on known factors of variation. In contrast to classical approaches in which a reconstruction error is optimized, we rely instead on the adversarial loss enabled by a convolutional discriminator. To this end, we introduce an architecture in which a *geometry mapping layer* serves as bridge between the generated 3D mesh and the 2D domain, for which convolutional layers can be applied (Section 4.1). To learn a decoupled parameterization, we build on the idea of Auxiliary Classifiers and introduce a feature loss to further improve the results (Section 4.3). We will consider here a model that decouples between identity and expression, however the principle can be easily extended to more factors.
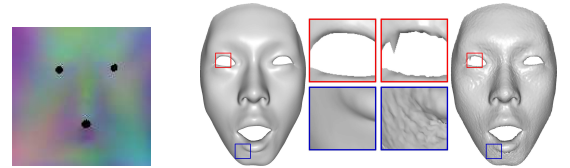
(a) Geometry image   (b) Original and reconstructed meshes

Figure 2: While a GAN could be used to generate geometry images, recovering the mesh from them is prone to artifacts, e.g. erroneous boundary interpolations (red) and precision loss (blue) in 2b. In this work we generate instead the 3D mesh, while geometry images are used only for discrimination.

4.1. Geometry Mapping Layer

While deep learning can be efficiently used on regularly sampled signals, such as 2D pixel grids, applying it to 3D surfaces is more challenging due to their irregular structure. In this work we propose to generate the 3D coordinates of the mesh using a multilayer perceptron, while the discriminative aspects are handled in the 2D image domain. This allows to benefit from efficient and well established architectures that have been proven to behave adequately under adversarial training, while still generating the 3D shape in its natural domain.

In particular, a 2D representation of a mesh can be achieved through a UV parameterization $\phi : \mathcal{V} \to D$ that associates each vertex $v_k \in \mathcal{V}$ with a coordinate $(u, v)_k$ in the unit square domain D. Continuous images can be obtained by interpolating the (x, y, z) vertex values according to the 2D barycentric coordinates, and storing them in the image channels. Borrowing the term from [18], we call this a *geometry image* (see Figure 2a).

Note that although our method could generate geometry images instead of 3D meshes, this would introduce an unnecessary additional reconstruction step that is likely to cause information loss and artifacts in the final meshes, as illustrated in Figure 2b. This is due to the fact that a single planar unfolding of a mesh may create distortions such as triangle flipping [34], and a many-to-one mapping may be obtained even with a bijective parameterization due to the finite size of images. In addition, as elaborated in [18], unless border vertices are preassigned to distinct pixels which can be challenging for large meshes, sampling these locations results in erroneous interpolations. Generating 3D point coordinates instead allows to avoid reconstruction artifacts, and to apply common mesh regularization techniques that simplify and improve the learning process. We use geometry images only as the representation for the discriminative component that evaluates the 3D generator through CNNs.

The mapping layer operates as follows. Given a mesh made of vertices $\mathcal{V} = \{v_k / k = 1..n_v\}$, a target image size $n \times n$, and a pre-computed UV parameterization ϕ, we build two images $I^{\mathcal{U}}$, $I^{\mathcal{V}}$ of dimension $n \times n$, and three images I^{v_1}, I^{v_2} and I^{v_3} of dimension $n \times n \times 3$ each. For each pixel (i, j), we consider the ϕ-projected mesh triangle $(\hat{v}_1, \hat{v}_2, \hat{v}_3)$ containing it. The barycentric abscissa and ordinate of pixel (i, j) in triangle $(\hat{v}_1, \hat{v}_2, \hat{v}_3)$ are then stored in images $I^{\mathcal{U}}$ and $I^{\mathcal{V}}$ respectively, and the original face vertex coordinates v_1, v_2 and v_3 are stored in images I^{v_1}, I^{v_2} and I^{v_3}. The mapping layer computes the output geometry image $\mathcal{I}$ as:

$$\mathcal{I} = I^{\mathcal{U}} * I^{v_1} + I^{\mathcal{V}} * I^{v_2} + (\mathbf{1} - I^{\mathcal{U}} - I^{\mathcal{V}}) * I^{v_3}, \quad (5)$$

where $*$ denotes element-wise multiplication and $\mathbf{1} \in \mathbb{R}^{n \times n}$ is the matrix of ones. Since this layer simply performs indexing and linear combinations on the elements of $\mathcal{V}$ us-

ing the predefined parameters in $I^{\mathcal{U}}$ and $I^{\mathcal{V}}$, all operations are differentiable and the gradients can be back-propagated from the discriminated image to the generated mesh.

4.2. Architecture

Figure 1 depicts our proposed architecture. The generator consists of two fully connected layers that map the latent code z to a vector of size $3n_v$ containing the stacked 3D coordinates of displacements from a reference face mesh. The output vertex positions are passed through the mapping layer to generate a geometry image of size $n \times n$, which is then processed by the discriminator in order to classify whether the generated mesh is real or fake. We also consider auxiliary classifiers for the discriminator, denoted as C_{id} and C_{exp}. The design of D shows two main differences with respect to the original AC-GAN. First, instead of classifying only one type of labels, we use here classifiers for both identity and expression. This favors decoupling, since the classification of one factor is independent of the choice of the labels for the other factors. Second, we provide distinct convolutional layers for the real/fake, identity and expression blocks. This is motivated by the observation that the features required to classify identities and expressions are not necessarily the same.

4.3. Decoupled Model Learning

We rely on the discriminator not only to generate realistic faces, but also to decouple the factors of variation. For this, we optimize D such that it maximizes

$$\mathcal{L}_D = \mathcal{L}_{GAN} + \lambda_C(\mathcal{L}_{ID} + \mathcal{L}_{EXP}). \quad (6)$$

Here, $\mathcal{L}_{GAN}$ denotes the standard adversarial loss (see Equation 2), and $\mathcal{L}_{ID}, \mathcal{L}_{EXP}$ the classification losses measured against the labels provided with the dataset and weighted by scalar λ_C. These losses are defined similarly to Equation 3 as:

$$\begin{aligned} \mathcal{L}_{ID} &= \mathbb{E}_{x \sim p_{data}, c \sim p_c^{id}}[\log P(C = c|x)], \\ \mathcal{L}_{EXP} &= \mathbb{E}_{x \sim p_{data}, c \sim p_c^{exp}}[\log P(C = c|x)], \quad (7) \end{aligned}$$

where p_c^{id} and p_c^{exp} denote the distribution of identity and expression labels, respectively. We ignore the sample contribution in the classification loss if it is not labeled.

The generator G takes as input a random vector $z = \{z_{id}, z_{exp}, z_{noise}\}$, which is the concatenation of the identity code $z_{id} \sim p_{id}$, the expression code $z_{exp} \sim p_{exp}$ and a random noise $z_{noise} \sim p_{noise}$. It produces the location of n_v displacement vectors from a reference mesh, and is trained by minimizing:

$$\begin{aligned} \mathcal{L}_G &= \lambda_1 \mathcal{L}_{GAN} - \lambda_2 \left(\mathcal{L}_{CL}^{id} + \mathcal{L}_{CL}^{exp} \right) \\ &+ \lambda_3 \left(\mathcal{L}_{FEAT}^{id} + \mathcal{L}_{FEAT}^{exp} \right) + \lambda_4 \mathcal{L}_{reg}, \end{aligned} \quad (8)$$

where $\mathcal{L}_{GAN}$ is the standard GAN loss (Equation 2); $\mathcal{L}_{CL}^{id}$ and $\mathcal{L}_{CL}^{exp}$ are classification losses; $\mathcal{L}_{FEAT}^{id}$ and $\mathcal{L}_{FEAT}^{exp}$ are feature losses that aim to further increase the decoupling of the factors; $\mathcal{L}_{reg}$ is a regularizer; and $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are weights for the different loss terms. We explain each of these in the following.

Classification Loss In addition to the adversarial loss, the generator is trained to classify its samples with the correct labels by maximizing:

$$
\begin{aligned}
\mathcal{L}_{CL}^{id} &= \mathbb{E}_{z \sim p_z, c \sim p_c^{id}}[\log P(C = c | G(z))] \\
\mathcal{L}_{CL}^{exp} &= \mathbb{E}_{z \sim p_z, c \sim p_c^{exp}}[\log P(C = c | G(z))]. \quad (9)
\end{aligned}
$$

In order to generate data belonging to a specific class, we sample one identity/expression code z_{id}, z_{expr} for each label and fix it throughout the training; this becomes the input for G each time the classification loss must be evaluated. We denote the set of fixed codes for identity and expression as $\mathcal{T}^{id}$ and $\mathcal{T}^{exp}$ respectively.

Feature Loss The classification loss is limited to codes in $\mathcal{T}^{id}/\mathcal{T}^{exp}$, which have associated labels. We found that better decoupling results can be obtained if we include a loss on the classifier features. We measure this by generating samples in pairs which share the same identity or expression vector, and measuring the error as:

$$
\mathcal{L}_{FEAT}^{id} = \frac{2}{N} \sum_{z_{id}} \left(1 - cos(\mathbf{f}_{1,z_{id}}, \mathbf{f}_{2,z_{id}})\right), \quad (10)
$$

$$
\mathcal{L}_{FEAT}^{exp} = \frac{2}{N} \sum_{z_{exp}} \left(1 - cos(\mathbf{f}_{1,z_{exp}}, \mathbf{f}_{2,z_{exp}})\right). \quad (11)
$$

Here, N is the batch size, and $\mathbf{f}_{i,z_{id}} = \mathbf{f}\left(G(z_{id}, z_{exp,i}, z_{noise,i})\right)$ are feature vectors obtained by inputting the sample $G(z_{id}, z_{exp,i}, z_{noise,i})$ through the classifier C_{id} and extracting the features from the second to last layer. That is, given two inputs which were generated with the same identity vector, $\mathcal{L}_{FEAT}^{id}$ enforces that their feature vectors in the identity classifier are also aligned. The definition is analogous for $\mathbf{f}_{i,z_{exp}}$ with C_{exp}.

To enable training with both classification and feature loss, for each batch iteration we alternate between the sampling of labeled identity codes $z_{id} \in \mathcal{T}^{id}$ with unlabeled expression codes $z_{exp} \sim p_{exp}$, and the sampling of unlabeled identity codes $z_{id} \sim p_{id}$ with labeled expression codes $z_{exp} \in \mathcal{T}^{exp}$. The classification is evaluated for the labeled factor only, while the feature loss is used for unlabeled codes, and the alternation allows to better cover the identity and expression sub-spaces during training.

Regularization Generating a 3D mesh allows us to reason explicitly at the surface level and define high order loss functions using the mesh connectivity. In particular, we enforce spatial consistency over the generated faces by minimizing the following term on the output displacements $\mathbf{v} = G(z)$:

$$
\mathcal{L}_{reg} = ||L\mathbf{v}||_2^2, \quad (12)
$$

where L is the cotangent discretization of the Laplace-Beltrami operator.

5. Results

We provide in this section results obtained with the proposed framework, which demonstrate its benefits particularly in decoupling. We first clarify our set-up with implementation details in Section 5.1 and the datasets used in 5.2. We explain in Section 5.3 the proposed metrics for the evaluation of a 3D face model, and introduce a new measure for analyzing the diversity of the generated samples. In Section 5.4 we perform ablation studies to verify that all the components are necessary to effectively train an expressive model. Finally, in Section 5.5 we compare our results to state-of-the-art 3D face models that can decouple the latent space, and show that our approach outperforms with respect to decoupling and diversity. Additional results can be found in the supplemental material.

5.1. Implementation Details

We set the weights to $\lambda_C = 0.1$ (Equation 6), $\lambda_1 = \lambda_2 = 1$, $\lambda_3 = 0.5$ and $\lambda_4 = 100$ (Equation 8). The classification losses are further weighted to account for unbalanced labels [23]. For the generator, we use two fully connected layers with an intermediate representation of size 512 and ReLU non-linearity. For the discriminator we use a variant of DC-GAN [28], with the first two convolutional blocks shared between C_{real}, C_{id} and C_{expr}, while the remaining are duplicated for each module (more details can be found in supplemental). The models were trained for 200 epochs using ADAM optimizer [24] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, a learning rate of 0.0002 and a batch size of 64. During training we add instance noise [36] with $\sigma = 0.1$ to the input of D. The discriminator is trained for 3 iterations each time we train the generator. The models take around 2 hours to train on a NVidia GeForce GTX 1080 GPU.

The template mesh contains 22129 vertices. We precompute the UV map ϕ using harmonic parameterization [12], setting the outer boundary face vertices to a unit square to ensure full usage of the image domain. We generate geometry images of size 64×64; we experimented with other image sizes but the best decoupling results were obtained with this resolution. The dimensions for $(z_{id}, z_{exp}, z_{noise})$ are set to $(65, 15, 5)$ to facilitate comparison with [13], and the feature vectors used in Equations 10 and 11 are of size 2048.

5.2. Datasets

All models were trained using a combination of four publicly available 3D face datasets. In particular, we use two datasets containing static 3D scans of multiple subjects: BU-3DFE [44] and Bosphorus [31], and combine these with two datasets of 3D motion sequences of multiple subjects: BP4D-Spontaneous [45] and BU-4DFE [43]. The static datasets provide variability of identities, while the motion datasets provide variability of expressions and a larger number of training samples. We registered BU-3DFE and Bosphorus with a template fitting approach [30], and the motion datasets with a spatiotemporal approach [14].

The final dataset contains 30559 registered 3D faces and was obtained by subsampling the motion sequences. We provide identity labels for all meshes, while the expression labels are limited to the seven basic emotional expressions, which appear in both static datasets. For BU-4DFE, expression labels are assigned to three frames per sequence: the neutral expression to the first and last frame, and the labeled expression of the sequence to the peak frame. For BP4D, one neutral frame is manually labeled per subject (this is a requirement for comparison to [29]). Overall, due to the use of motion data, only 7% of it is assigned expression labels.

5.3. Evaluation Metrics

We evaluate the models in terms of *diversity* of the generated samples, *decoupling* of identity and expression spaces, and *specificity* to the 3D facial shape. We believe it is necessary to simultaneously consider all the metrics, as they provide complementary information on the model. For instance, a good decoupling value can be obtained when the diversity is poor, since small variations facilitate the classification of samples as "same". Conversely, a large diversity value can be obtained when decoupling is poor, since the identities/expressions sharing the same code can yield very different shapes. We detail these in the following.

Diversity We consider it important to measure the diversity of the 3D face shapes generated by a model, particularly with GANs that are known to be prone to mode collapse. To the best of our knowledge, this has not yet been considered in the context of 3D face models and we propose therefore to evaluate as follows. We sample p pairs of randomly generated meshes and compute the mean vertex distance among the pairs; diversity is then defined as the mean of the distances over the p pairs. We expect here to see higher values for more diverse models. We evaluate on three sets of sampled pairs: (1) among pairs chosen randomly (*global diversity*), (2) among pairs that share the same identity code (*identity diversity*) and (3) among pairs that share the same expression code (*expression diversity*). For all cases we evaluate on 10000 pairs. For comparison, the training set is also evaluated on these three metrics by leveraging the labels.

Decoupling To evaluate decoupling in both identity and expression spaces we follow the protocol proposed in [11]. In particular, we first train two networks, one for identity and one for expression, that transform an image representation of the mesh to an n-dimensional vector using triplet loss [32], where $n = 128$ in our experiments. The trained networks allow to measure whether two meshes share the same identity or expression by checking whether the distance between their embeddings is below a threshold τ.

To measure identity decoupling, we generate n random faces $\mathbf{x}_i = G(z_{id}^i, z_{exp}^i, z_{noise}^i)$, and for each random face we fix the identity code and sample m faces $\mathcal{Y}(\mathbf{x}_i) = \{G(z_{id}^i, z_{exp}^j, z_{noise}^j), j = 1..m\}$. We then use the embedding networks to evaluate whether the original faces $\mathbf{x}_i$ and their corresponding samples in $\mathcal{Y}(\mathbf{x}_i)$ correspond to the same identity, and report the percentage of times the pairs were classified as "same". We proceed analogously for expression decoupling. We set $n = 100$, $m = 100$, $\tau = 0.14$ for identity and $\tau = 0.226$ for expression; more implementation details are given in the supplemental material.

Specificity Specificity is a metric commonly used for the evaluation of statistical shape models [9] and whose goal is to quantify whether all the generated samples belong to the original shape class, faces in our case. For this, n samples are randomly drawn from the model and for each the mean vertex distance to each member of the training set is measured, keeping the minimum value. The metric then reports the mean of the n values. We use here $n = 1000$.

5.4. Ablation Tests

We start by demonstrating that each of the proposed components is necessary to obtain state-of-the-art results in the proposed metrics. To this end, we compare our approach against the following alternatives: (1) without mesh regularization (Equation 12); (2) with identity classification only; (3) with expression classification only; and (4) without feature loss (Equations 10 and 11).

Table 1 gives the evaluation metrics for each of these options, and Figure 3 provides qualitative examples. From the results we observe that: (1) The mesh regularization is crucial to generate samples that are realistic facial shapes. This is reflected by a very large value in specificity as well as low diversity, due to the fact that the model never converged to realistic faces (see Figure 3a). (2) Considering classification in only one factor significantly reduces the capacity of the model to preserve semantic properties in the other factor, as indicated by the very low decoupling values obtained in the corresponding rows. This justifies the use of classifiers for each of the factors. (3) Without the feature loss the model can still achieve good results, but both expression decou-

(a) Without mesh regularization

(b) Without expression classification

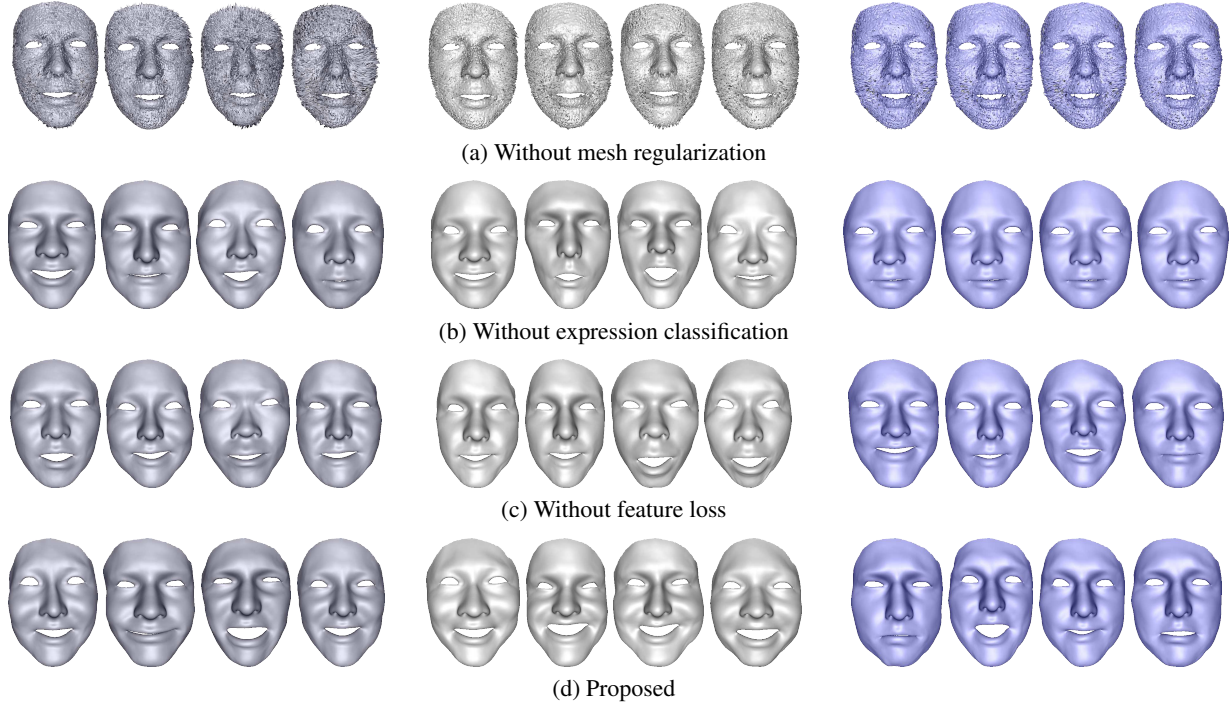(c) Without feature loss

(d) Proposed

Figure 3: Qualitative results for alternative approaches. From left to right: randomly generated samples (dark gray), random samples with a same expression code (light gray), random samples with a same identity code (purple).

	Dec-Id	Dec-Exp	Div	Div-Id	Div-Exp	Sp.
Training data	—	—	4.89	3.30	5.04	—
w/o mesh reg.	99.6	99.1	1.41	0.65	1.25	3.61
w/o exp. class.	100.0	42.8	4.81	0.11	4.87	2.01
w/o id. class.	7.8	98.9	5.28	4.87	2.05	2.22
w/o feat. loss	96.0	80.3	4.47	1.75	4.01	2.00
3DMM [1]	99.6	65.6	3.53	1.95	2.89	2.30
MAE [13]	99.5	53.3	3.89	0.92	3.76	2.00
CoMA [29]	97.5	65.5	3.38	1.71	2.90	2.47
Ours	98.6	89.7	4.74	1.94	4.22	2.01

Table 1: Quantitative evaluation with respect to decoupling of identity and expression (*Dec-*, percentage), diversity (*Div-*, in mm) and specificity (*Sp.*, in mm.). Higher is better, except for specificity.

pling and diversity are lower than with the full model and the inclusion of the feature loss improves expression classification by almost 10%. Note that decoupling the expression space is significantly more challenging than identity, as the provided labels are very sparse. This effect is illustrated on Figure 3c, where models with the same expression code can lead to faces with slightly different expressions. Our approach provides more coherent faces, as shown in Figure 3d.
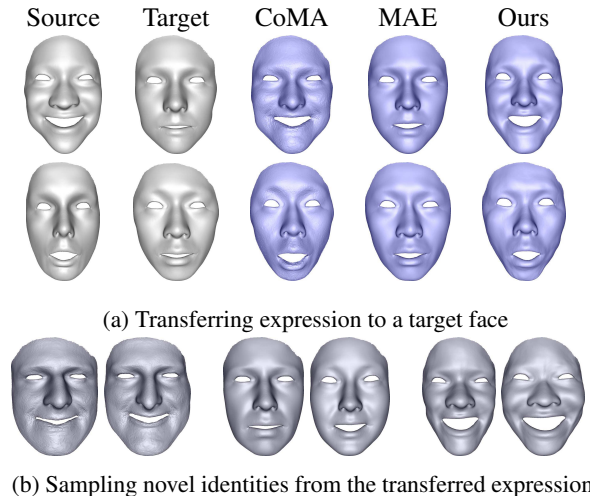

(a) Transferring expression to a target face

(b) Sampling novel identities from the transferred expression.

Figure 4: Comparison in terms of expression transfer. Top: expression code z_{expr} transferred to a target identity. Bottom: using z_{expr} from the source in the top row, we sample novel identities (left to right: CoMA, MAE, ours).

5.5. Comparisons

We compare the proposed approach against state-of-the-art generative 3D face models. Our goal is to build a de-

coupled latent space, and thus we focus the comparison to works that either enforce this explicitly [13], or combine a model trained on expressions with a linear space of identities [29, 1]. We train all models using the same dimensions (65 for identity and 20 for expression).

The model proposed in [13], called MAE in the following, was trained with the same dataset and the same label information (Section 5.2) for 200 epochs, with the default parameters given in the paper. We initialize the encoder and the decoder from the publicly available models.

The model proposed in [29], called CoMA in the following, does not explicitly favor decoupling and thus we use the DeepFLAME alternative [25], which we also train with the same dataset. This results in a PCA model built from 299 identities and an autoencoder trained on 30330 displacements from the corresponding neutral face. For the identity space we manually selected one neutral frame for each sequence in BP4D-Spontaneous, as this dataset does not provide labels. The model was trained using the publicly available code for 200 epochs.

We also trained an additive linear model as described in [1] using our dataset, and the same neutral/expression separation selected for CoMA (see above). We refer to this model as 3DMM.

Model quality We show quantitative results with respect to decoupling, diversity and specificity in the bottom of Table 1. Note that the proposed approach significantly outperforms the others in terms of *expression decoupling*, which is more challenging than identity due to the sparse labeling. This is shown qualitatively in Figure 4, where we transferred expressions by simply exchanging the latent code z_{exp}. We can see here that the expression is well preserved by our model.

With respect to *identity decoupling* the four methods perform similarly well, with 3DMM achieving the highest value. Note that, in the case of MAE, the large decoupling value is combined with the lowest diversity in identity (*Div-Id*), which suggests limited generative capabilities (see supplemental for a qualitative example).

We also outperform all methods in terms of *diversity*. Combined with a specificity value that is among the best, this implies that our model has learned to generate significant variations that remain valid facial shapes.

Reconstruction of Sparse Data We also tested the generalization of the model with the reconstruction of partial face data given very sparse constraints. To this purpose, we use the dataset provided by [29], which contains 12 subjects performing 12 extreme expressions. We take the middle frame of each sequence and manually label 85 landmarks (see supplemental), resulting in a testing set of 144 subjects.

Method	$\lambda = 0$	$\lambda = 0.01$	$\lambda = 10$
3DMM [1]	6.62	4.64	2.46
MAE [13]	4.46	4.06	2.78
CoMA [29]	3.05	3.02	2.83
Ours	**2.62**	**2.55**	**2.42**

Table 2: Reconstruction of sparse data under different regularization weights (RMSE, in mm).

The face model is fitted by minimizing:

$$\arg\min_z \sum_{i=1}^{p} ||\tilde{\mathbf{v}}_i(z) - \mathbf{v}_i||_2^2 + \lambda ||z||_2^2, \qquad (13)$$

where $\mathbf{v}_i$ are the 3D locations of the p key-points in the testing set, $\tilde{\mathbf{v}}_i(z)$ are the corresponding key-points in the face model generated with code z, and λ the regularization weight. We optimize using a gradient descent approach [24] starting from a randomly sampled code z. Note that this is a challenging scenario since the training set does not contain such expressions, and the correspondences are very sparse.

We compare our results with those obtained with 3DMM, MAE and CoMA, using the same optimization for all methods. We measure the reconstruction error against the ground-truth surface and report the RMSE. Quantitative results can be found in Table 2 for different regularization weights λ. Our method outperforms in all cases, including without regularization ($\lambda = 0$). We found that our model can produce reasonable faces in most cases, while MAE and CoMA easily produce un-realistic faces when the regularization is not strong enough (qualitative examples can be found in the supplemental material).

5.6. Extension to other factors

The proposed framework can easily be extended to other factors of variation, such as identity/expression/viseme. We refer to the supplemental material for an example of such a model.

6. Conclusion

We explored in this work the use of adversarial training for learning decoupled 3D facial models and showed that we can achieve state-of-the-art performance in terms of decoupling and diversity of the generated samples. This was obtained through a novel 3D-2D architecture, as well as a training scheme that explicitly encourages decoupling through the use of classifiers. Although the expressiveness of the model remains limited by the diversity of the training data and the accuracy of its labels, we show that adversarial learning has the potential to make better use of the available data in building performant 3D facial models.

References

[1] Brian Amberg, Reinhard Knothe, and Thomas Vetter. Expression invariant 3d face recognition with a morphable model. In *Conference on Automatic Face and Gesture Recognition*, pages 1–6, 2008. 1, 2, 7, 8

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017. 3

[3] Timur Bagautdinov, Chenglei Wu, Jason Saragih, Pascal Fua, and Yaser Sheikh. Modeling Facial Geometry using Compositional VAEs. In *Conference on Computer Vision and Pattern Recognition*, volume 1, page 1, 2018. 1, 2

[4] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH*, pages 187–194, 1999. 1, 2

[5] James Booth, Anastasios Roussos, Stefanos Zafeiriou, Allan Ponniah, and David Dunaway. A 3d morphable model learnt from 10,000 faces. In *Conference on Computer Vision and Pattern Recognition*, 2016. 1, 2

[6] Alan Brunton, Timo Bolkart, and Stefanie Wuhrer. Multilinear wavelets: A statistical shape space for human faces. In *European Conference on Computer Vision*, 2014. 2

[7] Chen Cao, Derek Bradley, Kun Zhou, and Thabo Beeler. Real-time high-fidelity facial performance capture. *Transactions on Graphics*, 2015. 2

[8] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Facewarehouse: a 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 3:413–425, 2014. 1

[9] Rhodri Davies, Carole Twining, and Chris Taylor. *Statistical models of shape: Optimisation and evaluation*. Springer Science & Business Media, 2008. 6

[10] Jiankang Deng, Shiyang Cheng, Niannan Xue, Yuxiang Zhou, and Stefanos Zafeiriou. Uv-gan: Adversarial facial uv map completion for pose-invariant face recognition. In *CVPR*, pages 7093–7102, 2018. 2

[11] Chris Donahue, Zachary C Lipton, Akshay Balsubramani, Julian McAuley, Sepehr Rezvani, Nader Mokari, Mohammad R Javan, Maria Henar Salas-Olmedo, Juan Carlos Garcia-Palomares, Javier Gutierrez, et al. Semantically decomposing the latent spaces of generative adversarial networks. *International Conference on Learning Representations*, 2018. 6

[12] Matthias Eck, Tony DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsbery, and Werner Stuetzle. Multiresolution analysis of arbitrary meshes. In *SIGGRAPH*, 1995. 5

[13] Victoria Fernández Abrevaya, Stefanie Wuhrer, and Edmond Boyer. Multilinear autoencoder for 3d face model learning. In *Winter Conference on Applications of Computer Vision*, pages 1–9, 2018. 1, 2, 5, 7, 8

[14] Victoria Fernández Abrevaya, Stefanie Wuhrer, and Edmond Boyer. Spatiotemporal modeling for efficient registration of dynamic 3d faces. In *International Conference on 3D Vision*, pages 371–380, 2018. 6

[15] Baris Gecer, Stylianos Ploumpis, Irene Kotsia, and Stefanos Zafeiriou. Ganfit: Generative adversarial network fitting for high fidelity 3d face reconstruction. *arXiv preprint arXiv:1902.05978*, 2019. 2

[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 1, 2

[17] Stella Graßhof, Hanno Ackermann, Sami S Brandt, and Jörn Ostermann. Apathy is the root of all expressions. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 658–665. IEEE, 2017. 1

[18] Xianfeng Gu, Steven J. Gortler, and Hugues Hoppe. Geometry images. In *SIGGRAPH*, 2002. 4

[19] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of Wasserstein GANs. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017. 3

[20] Loc Huynh, Weikai Chen, Shunsuke Saito, Jun Xing, Koki Nagano, Andrew Jones, Paul Debevec, and Hao Li. Mesoscopic facial geometry inference using deep neural networks. In *Conference on Computer Vision and Pattern Recognition*, pages 8407–8416, 2018. 2

[21] Alexandru Eugen Ichim, Sofien Bouaziz, and Mark Pauly. Dynamic 3d avatar creation from hand-held video input. *ACM Transactions on Graphics (ToG)*, 34(4):45, 2015. 1

[22] Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Niessner, Patrick Pérez, Christian Richardt, Michael Zollhöfer, and Christian Theobalt. Deep video portraits. *ACM Transactions on Graphics (TOG)*, 37(4):163, 2018. 1

[23] Gary King and Langche Zeng. Logistic regression in rare events data. *Political analysis*, 9(2):137–163, 2001. 5

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference for Learning Representations*, 2015. 5, 8

[25] Tianye Li, Timo Bolkart, Michael Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ACM Transactions on Graphics*, 36(6):194:1–17, 2017. 2, 8

[26] Thomas Neumann, Kiran Varanasi, Stephan Wenger, Markus Wacker, Marcus Magnor, and Christian Theobalt. Sparse localized deformation components. *ACM Transactions on Graphics*, 32:179:1–10, 2013. 2

[27] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional Image Synthesis with Auxiliary Classifier GANs. In *International Conference on Machine Learning*, 2017. 2, 3

[28] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations*, 2016. 5

[29] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J Black. Generating 3d faces using convolutional mesh autoencoders. In *European Conference on Computer Vision*, 2018. 1, 2, 6, 7, 8

[30] Augusto Salazar, Stefanie Wuhrer, Chang Shu, and Flavio Prieto. Fully automatic expression-invariant face correspondence. *Machine Vision and Applications*, 25(4):859–879, 2014. 6

[31] Arman Savran, Neşe Alyüz, Hamdi Dibeklioğlu, Oya Çeliktutan, Berk Gökberk, Bülent Sankur, and Lale Akarun. Bosphorus database for 3d face analysis. In *European Workshop on Biometrics and Identity Management*, pages 47–56. Springer, 2008. 6

[32] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 6

[33] Gil Shamai, Ron Slossberg, and Ron Kimmel. Synthesizing facial photometries and corresponding geometries using generative adversarial networks. *arXiv preprint arXiv:1901.06551*, 2019. 1, 2

[34] Alla Sheffer, Emil Praun, and Kenneth Rose. Mesh parameterization methods and their applications. *Found. Trends. Comput. Graph. Vis.*, 2(2):105–171, 2006. 4

[35] Ron Slossberg, Gil Shamai, and Ron Kimmel. High quality facial surface and texture synthesis via generative adversarial networks. In *European Conference on Computer Vision*, pages 498–513. Springer, 2018. 2

[36] Casper Kaae Sønderby, Jose Caballero, Lucas Theis, Wenzhe Shi, and Ferenc Huszár. Amortised map inference for image super-resolution. In *ICLR*, 2017. 5

[37] Ayush Tewari, Michael Zollhöfer, Pablo Garrido, Florian Bernard, and Hyeongwoo Kim. Self-supervised multi-level face model learning for monocular reconstruction at over 250 hz. In *Conference on Computer Vision and Pattern Recognition*, 2018. 2

[38] Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2387–2395, 2016. 1

[39] Luan Tran and Xiaoming Liu. Nonlinear 3d face morphable model. In *Conference on Computer Vision and Pattern Recognition*, 2018. 2

[40] Daniel Vlasic, Matthew Brand, Hanspeter Pfister, and Jovan Popović. Face transfer with multilinear models. *ACM Transactions on Graphics*, 24(3):426–433, 2005. 1, 2

[41] Thibaut Weise, Sofien Bouaziz, Hao Li, and Mark Pauly. Realtime performance-based facial animation. In *ACM transactions on graphics (TOG)*, volume 30, page 77. ACM, 2011. 1

[42] Shuco Yamaguchi, Shunsuke Saito, Koki Nagano, Yajie Zhao, Weikai Chen, Kyle Olszewski, Shigeo Morishima, and Hao Li. High-fidelity facial reflectance and geometry inference from an unconstrained image. *ACM Transactions on Graphics*, 37(4):162, 2018. 2

[43] Lijun Yin, Xiaochen Chen, Yi Sun, Tony Worm, and Michael Reale. A high-resolution 3d dynamic facial expression database. In *International Conference on Automatic Face & Gesture Recognition*, pages 1–6. IEEE, 2008. 6

[44] Lijun Yin, Xiaozhou Wei, Yi Sun, Jun Wang, and Matthew J Rosato. A 3d facial expression database for facial behavior research. In *International Conference on Automatic face and gesture recognition*, pages 211–216. IEEE, 2006. 6

[45] Xing Zhang, Lijun Yin, Jeffrey F Cohn, Shaun Canavan, Michael Reale, Andy Horowitz, Peng Liu, and Jeffrey M Girard. Bp4d-spontaneous: a high-resolution spontaneous 3d dynamic facial expression database. *Image and Vision Computing*, 32(10):692–706, 2014. 6