Property Tunable Image Manipulation with Self-Generated Supervision

Anonymous WACV submission

Paper ID 697

Abstract

We propose instance-invariant image synthesis steered by continuously tunable properties. The key challenge is that training data distribution is often sparse, including the extreme case of having only one instance sample. We introduce an infinite synthetically-generated scheme to continuously manipulate the input image. For each training image, the model self-generates its manipulated image with the synthesized desired property label as the self-supervision for the model's training. A key objective is to produce desired property change while maintaining instance-invariance. We implement this using a TunerGAN: infusing parallel streams of input-output messages to the GAN network. The messages, once decoded into feature maps represent tunable properties and instance invariance. The output messages extracted from the synthesized image, are self-checked with the input messages to ensure instance-level synthesis is done correctly: e.g.*, face rotated correctly with identity preserved. Our framework is general for image manipulation and we have conducted experiments on a variety of tasks to demonstrate its effectiveness.*

1. Introduction

Image generation via domain style transfer has demonstrated the ability to create images with desired properties at a category level [38, 1]. One can generalize this approach to multiple-property manipulations by treating property states as different domains/styles, as described in [26, 18, 4]. A less explored but equally important direction is to manipulate specific properties of images at *instance-level*. For example, we want to rotate a person's head without changing his/her identity and expression. This is not only helpful for more controllable image editing but also beneficial to many image understanding tasks such as state-estimation and recognition (identification, categorization, etc.). Learning to manipulate a specific property (*e.g.* head pose) or multiple parallel properties can be realized by learning two disentangled representations, one for the related properties themselves and one for the other information, which is independent of these properties. The former representation is facilitated to estimate the state of the selected property (head pose), while the latter is more robust for recognition tasks (*e.g.* face recognition) due to its invariance to the properties' changes. This is conceptually consistent with the general strategy of perception by synthesizing.

The distinction between category-level vs instance-level image manipulation is important. For category-level image manipulation, the learned model is not required to separate the tuning property from instance invariant information. For example, to change one's facial expression, we would not want to add sunglasses to his/her face. Approaches such as InfoGAN [1] can generate an image with desired property change but often confound related factors together. Three fundamental challenges for instance-level properties tuning are 1) succinct controllability, 2) instance-level descriptiveness, and 3) instance-level discriminativeness.

Succinct controllability means we want to have simple control variables, such as degree of head pose/smile or exact age, to control image generation. MUNIT-like approaches [12] use latent vectors to control image generations would need to automate the generation of the latent vectors to work in this case. This is not an easy task since we don't have data for most of the image manipulation targets. In our algorithm, we self-generate data and learn from it at the same time. Furthermore, we ensure succinct controllability by learning a decoder that turns a tunable property (*e.g.* pose) into a latent feature vector.

Instance-level descriptiveness means that we want to preserve as many details about that instance (*e.g.* a person's identity and appearance when tuning his/her head pose). Instance-level discriminativeness means we have a precise checking on the generated images to ensure the desired properties variation while maintaining instance-invariance. The two objectives of descriptiveness and discriminativeness are often in conflict with each other. The richer descriptiveness of the instance details, the harder it is for the discriminativeness check. One way to prevent cheating is to ensure self-generation consistency: check if the descriptiveness feature can generate the original image. The approach in MUNIT [12] seems to offer a solution: it ensures the im-

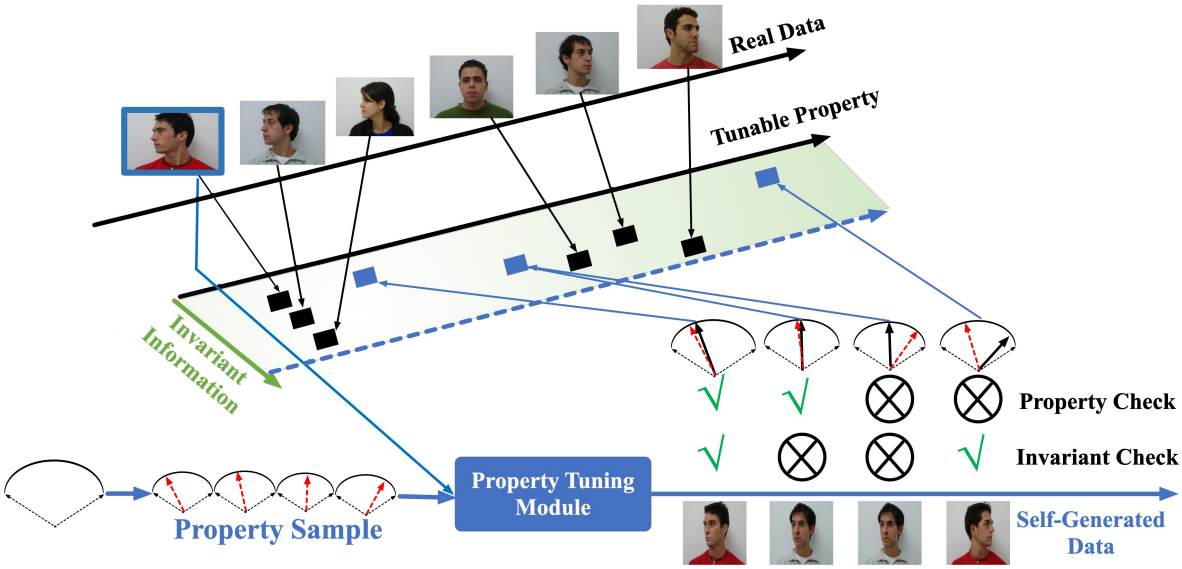

Figure 1. Tunable image manipulation with self-generated supervision. The real data distribution on the top is sparse and unbalanced. For an input image, we disentangle its representation into the tunable property and the invariant information. Then we generate images having different property values by the property tuning module. The disentanglement is checked by whether the generated image is the same person as the source image and whether the property is the same as the sampled property. To show more effectively, we only show the framework to tune one specific property. For tuning multiple properties, the tunable capsule network has multiple parallel outputs, which are responsible for different properties.

ages before-and-after manipulation share the same 'content' feature code. However, sharing exactly the same code throw away too many informative features. For example, the nose has unique features different from the frontal to profile view, and yet we want to preserve two different nose details. We solve this using Capsule Network as a discriminativeness checking mechanism. The many-to-one mapping property of the Capsule network provides us the flexibility of having a rich multimodal descriptiveness instance encoding while ensuring precise instance-level discriminativeness.

We formulate instance-level image properties tuning as a message-passing image generator. The message-passing framework explicitly exposes the input streams of 1) succinct controllability, 2) instance-level descriptiveness features, and output stream of 3) instance-level discriminativeness features. These messages play the roles of 'style' and 'content' code of the MUNIT, but they are succinct and continuously tunable. Inside of the generator, these messages are expanded into expressive multimodal instance-level feature vectors. To realize the above designs, we develop a novel generative adversarial network named TunerGAN, which contains the following three distinctive novelties:

- **Self-Generated Supervision**: the supervision for tuning (pairs of a synthesized label and corresponding manipulated data) is self-generated, as shown in Figure 1. Such a self-supervision frees the model from relying on paired training samples so that any infinite feasible state values can be sampled for training. This also allows training with diverse and arbitrarily sparsely distributed data in the property space (including the extreme case of having only one single sample for some training instances).

- **Capsule Structure for Representation**: two capsule blocks are introduced for learning representations for property-invariant information and property-state information, respectively. Using capsule networks has two advantages: a) its self-attention mechanism allows more robust representation; b) its vector representation can model multimodal distributions to regress complex continuous signals.

- **Multi-Branch Estimator**: property state estimation is critical for effective tuning as it provides feedback for loss checking, and with a multi-branch architecture the state estimator can focus on different aspects and then fuse the results, leading to superior performance.

2. Related Work

2.1. Image Translation and Image Manipulation

In general, image-to-image translation describes a task converting an image with source representation to an image with target representation. Many typical computer vision topics can be summarized as image-to-image translation tasks, including semantic segmentation [21, 33], image restoration and enhancing [22, 35], image editing and

inpainting [6, 25, 32], super resolution [16, 3, 28]. Due to the success of extensions on conditional GANs [24, 13, 30, 4], many works (such as CycleGAN [38], etc. [34, 17, 12]) develop translation models between two distinctive domains, and some others like StarGAN [4] and STGAN [18] can even handle the translation among multiple domains. DRIT [17] and MUNIT [12] introduced a content-preserving loss to achieve representation disentanglement by using two encoders to encode content and style separately, which allows easier style translation. For image manipulation, a representative work [2] first explores multiple paths to perform the translation and generate the manipulated intermediate images between two given samples. However, these methods are limited to the translation among multiple fixed domains or interpolation between two specific samples. In contrast, our TunerGAN can synthesize a manipulated output with targeted property/properties to be of arbitrary value(s), not restricted to those of the actual samples. By tuning property/properties with direct input values whilst preserving the instance-invariant information, TunerGAN can supervise itself via property-specified synthesizing and invariant property value checking after translation, which is not possible in any existing work or their possible combinations (*e.g.* StarGAN + MUNIT).

2.2. Capsule Network

The concept of capsules is invented in [9] and used recently in [27, 10]. CapsNet [27] is designed for image feature extraction, which is developed based on CNN. Unlike traditional CNN, in which the presence of feature is represented with scalar values in feature maps, the features in CapsNet are represented with capsules (vectors). In [27], the direction of capsules reflects the properties of the features and the length (2-norm) of capsules reflects the probability of the presence of different features. The transmission of information between layers follows the dynamic routing mechanism, which can help us obtain richer information than before. For example, the elements of the capsule may be used for learning different local or global properties, such as color, shape, size, or pose of objects. And the whole length of the capsule can indicate the probability of existence. We adopt capsule architecture to obtain a better global and local representation of visual images.

3. The Proposed Method

This section presents the overall framework and key components, with design motivations and insights explained. Due to the space limits, detailed network structures and implementations are left to the supplementary material.

3.1. Image Manipulation by Property Tuning

Image manipulation can be formulated as domain/style transfer or synthesis tasks. We aim to modulate a tunable property α (*e.g.* head pose) on an instance image I_s from one source state α_s to another state α_t, while keeping other properties β unchanged. This can be realized by learning to disentangle two kinds of representations, one for the tunable property α itself and the other for the rest information β which is independent of α. We generate the disentangled representations using a property decoder P, an invariant encoder E and an invariant capsule module C. P projects a low dimensional property value to the property representation so that we can tune the specific property by controlling the value of α. E encodes the invariant properties. The image manipulation with the disentangled feature can be realized by:

$$I_t = G(E(I_s) + P(\alpha_t)), \tag{1}$$

where G is the image generation function, I_s is the source image, and I_t is the tuned image.

We propose a self-guided generation of data with the continuous-valued tunable property. It is a significant step because we can now sample an infinite number of labels as well as manipulate any training image according to the self-generated labels. By separating the image information from the tunable properties, we also obtain control over the invariant properties of each sampled image. As shown in Figure 1 and Figure 2, we randomly sample one target state value α_t to generate I_t from the image I_s. Then we optimize an estimator η to measure the state representation for both the real and generated images. To be noted, the two disentangled features are both checked: 1) we check whether the invariant features are the same, *i.e.* $C(E(I_s)) \stackrel{?}{=} C(E(I_t))$, to ensure the generated sample images retain the same identity, where C is the invariant capsule to project the feature representation to low-dimensional capsule output; 2) for the tunable property, we also check its state by $\eta(I_t) \stackrel{?}{=} \alpha_t$. Moreover, we compute the cycle-consistency loss between I_s and re-manipulated image $\hat{I}_s = G(E(I_t) + P(\alpha_s))$.

Training with self-generated data makes it possible to continuously tune the property and efficiently explore the property space. Checking both the invariant and the tunable property allows the representation to be disentangled as expected, so we can manipulate the tunable property while maintaining other properties. An adversarial discriminator is also utilized to teach the generator to focus on the details which are irrelevant to the tunable property, and we adopt a multi-branch structure to stabilize the training.

3.2. Training with Self-generated Data

The training sample for the image manipulation is paired with the corresponding label, where the input image I_s checks with the labeled property α_s by $\eta(I_s) \stackrel{?}{=} \alpha_s$, and the generated image I_t checks with α_t by $\eta(I_t) \stackrel{?}{=} \alpha_t$. As a result, the property label α_s is paired with the train

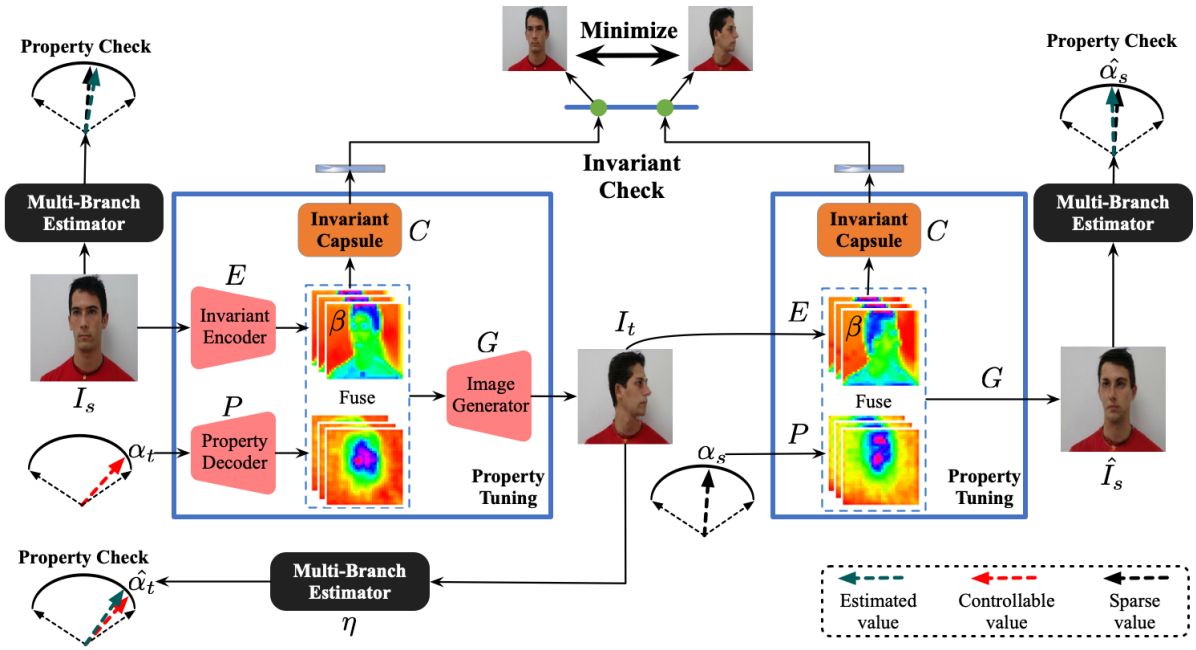

Figure 2. Training the property tuning module with the invariant check and the property check. The image feature from the invariant encoder is converted into the capsule vector by the invariant capsule module. We conduct the invariant checking by minimizing the distance between the capsule vectors. The source and generated images have their properties checked by a multi-branch estimator to get aligned with the labeled or desired properties. Besides the invariant capsule (C), other components of the second reconstruction part including the invariant encoder (E), property decoder (P), and image generator (G) are not depicted to avoid congested illustration.

image I_s collected from a very sparse property space. In contrast, by sampling the target property α_t from a continuous property space, the input image I_s can be tuned to any α_t. In addition, we train with the paired data of generated image $\hat{I}_s$ and source image I_s by checking whether $\eta(\hat{I}_s) \stackrel{?}{=} \eta(I_s)$. As illustrated in Figure 2, the generated sample I_t has its corresponding self-generated label α_t. The self-generated data ensures any feasible property values can be explored during training by randomly sampling α_t, so the model can be freed from relying on a large amount of labeled training samples and makes it possible to disentangle the tuning property with the invariant property. It also allows the model to train with diverse and arbitrarily sparsely distributed data. For example, starting from one single face image of a person, the self-generated data can leverage the transferred knowledge from another person with a different face pose to generate self-supervised data for self-learning.

3.3. Checking with Capsule Representation

To disentangle the tunable property vs invariant ones, we need to have a self-attention mechanism which can select the relevant information from each image. For example, as the face turns, a self-attention mechanism is needed to relocate the key facial features for checking. Furthermore, the invariant properties could be diverse with multimodal distributions. We adopt the capsule network to support the disentanglement. We use the capsule representation for two purposes: 1) dynamic routing of capsule finds where is the relevant spatial information, so we can be adaptive to the image variation due to tuning; 2) by using the vector representation, the capsule can encode a multimodal distance by condensing the error measure to the length of the vector.

Invariant Check. As shown in Figure 2, we adopt a capsule network to project the invariant information extracted from an image to a low-dimensional vector output. A crucial design of our system is using capsule to guide the image generation process, as well as checking the invariant properties. For the example of head rotation, we aim to make sure that the face identity is maintained during the tuning process. The invariant information β is modeled by the capsule network as a vector representation. We establish the loss function for the invariant check with capsule vectors by:

$$\mathcal{L}_{\text{i}} = |\|C(E(I_t))\|_2 - \|C(E(I_s))\|_2| \qquad (2)$$

where $C(E(I_s))$ is the capsule vector of I_s, $C(E(I_t))$ is the capsule vector of I_t and $\|\cdot\|_2$ indicates L2-norm. During back propagation, the gradient from the capsule updates the network to make sure that the two images I_s and I_t share the same invariant information, as shown in Figure 2.

Tunable property Check. As shown in Figure 3, a capsule network is also proposed for tunable property check. A arbitrary image (*e.g.* I_t) is first fed into a backbone network, and then the output feature maps are processed by a capsule

network based estimator (*e.g.* E_1 in the Figure) to extract the tunable property related information and turn it into a capsule vector whose L2-norm serves as the tunable property state estimation. Comparing with directly regressing from the feature map, the capsule network selects most relevant information that can suppress irrelevant information such as lighting and shading. As explained in the following subsection, we choose to use a multi-branch structure for the property estimator in our TunerGAN, thus each branch η_i contains a capsule-based estimator which generates an estimated property $\hat{\alpha}_t^i$ that can be used for computing the following loss:

$$\mathcal{L}_{\mathrm{p}}^i = \|\eta_i(I_t) - \alpha_t\|_1 = \left\|\hat{\alpha}_t^i - \alpha_t\right\|_1 . \quad (3)$$

3.4. Multi-Branch Estimator and Discriminator

To make sure we are generating a real image from the desired distribution, additional adversarial loss, and tunable property checking loss are adopted by introducing a Multi-branch Estimator and Discriminator module (MBED). It has two functionalities: 1) property checking and 2) image quality checking, realized by estimators and discriminators, respectively. To prevent overfitting, we deploy partial parameter sharing between "property checking" and "image quality checking" to facilitate gradient propagation between them. The GAN training often suffers from vanishing gradient and mode collapse. To stabilize the training, we design an ensemble of multiple branches for both estimator and discriminator. We observe in our experiments, as also shown in Figure 3, the multi-branch structure can force the proper checking to be more precise and the discriminators to focus on different facial features. Each estimator use a capsule network for property checking with $\mathcal{L}_{\mathrm{p}}^i$ (Eq. 3) and the overall property checking loss is just a simple sum of the losses from the branches: $\mathcal{L}_{\mathrm{p}} = \sum_{i=1}^{N} \mathcal{L}_{\mathrm{p}}^i$, where N is the number of branches and it is set to 4 in our experiments. For image quality checking, we use the PatchGAN loss [13] to constrain each discriminator.

4. Experiments

4.1. Datasets

UTKFace [37] is a large-scale face dataset with long age span (range from 0 to 116 years old). The dataset consists of over 20,000 face images with annotations of age, gender, and ethnicity. It is a challenging task to generate continuous facial images with age changes.

RaFD [15] contains pictures with 67 identities displaying 8 emotional expressions. In our experiments, we use identities between 1 and 60 identities for training and the rest for evaluation. Note we only use the frontal faces to capture the emotion distribution.

FEI [29] is a Brazilian face database. There are 14 images for each of 200 individuals, resulting in a total of 2800 images. The profile rotation is up to about 180 degrees with 10 discrete samples. The first 160 identities are used for training and others for evaluation.

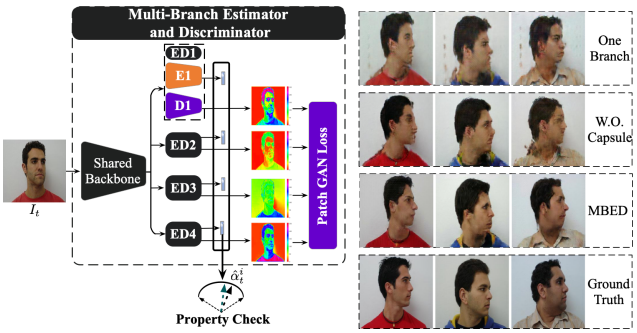

Figure 3. On the left, we show the structure of our multi-branch Estimator and Discriminator (MBED). On the right, we compare the performance of the property tuning network trained by MBED, one capsule with one discriminator, and the multi-discriminator without capsule. All images are generated by the same head pose as the ground truth. The MBE can teach the generator to give better images even for a big head pose changing.

Foggy Kitti [7] provides 7 categories fog images with different visibility: 30m, 40m, 50m, 75m, 150m, 375m, 750m. We follow the original train/test split.

CelebA [19] is a large face dataset containing ~210,000 images with 40 different attribute annotations. We perform multiple-property tuning on this dataset.

4.2. Evaluation Metric and Experimental Setup

Fréchet Inception Distance (FID) [8] is proposed to compute the similarity between the generated sample distribution and real data distribution. A lower FID score indicates better-generated image quality.

Learned Perceptual Image Patch Similarity (LPIPS) is proposed by [36], which computes the perceptual similarity between two image patches. Lower LPIPS suggests two image patches have closer perceptual similarities.

MSE, **RMSE** [31], **PSNR** [11] and **SSIM** [11] are used to measure the distance for the paired data. The lower MSE and RMSE values indicate better performance while the higher PSNR and SSIM indicate a closer similarity.

Classification Acc [4] measures the distance on the feature expression. We use the off-the-shelf classification model with Inception-V3 architecture to compute the accuracy.

To evaluate our TunerGAN, we compare with state-of-the-art methods in different tasks. For paired samples, we compare our method with paired methods: Pix2pix [13] and PG^2 [23]. As for unpaired methods, we compare our method with StarGAN [4], STGAN [18], and StarGAN+MUNIT.

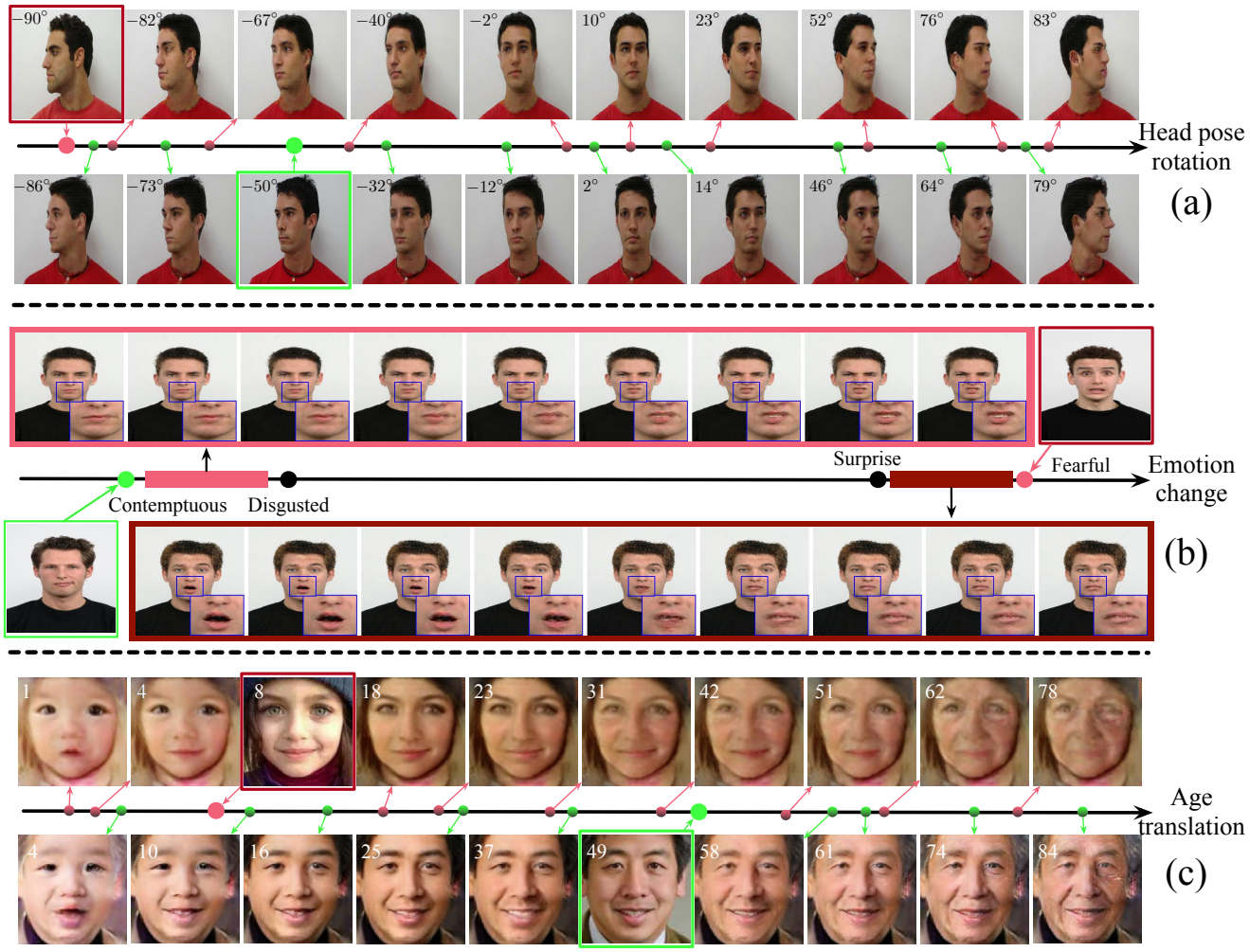

Figure 4. The visual translation results of TunerGAN on FEI, RaFD, and UTKface datasets. The images in red and green boxes show the real input samples, and other images are generated samples. The upper left corner of the image is marked with the pose angle of the generated images. The carnations and green dots represent the projection in the latent space of the generated samples.

4.3. Single-property tuning

We first evaluate our method on the continuous **head pose rotation** task using the FEI dataset. Figure 4(a) exhibits the results of the tuning process of the head pose. The images in red and green boxes are real input images while others are generated samples. We project the images into head pose space and use a red dot for the first image and green for the second. The results demonstrate that our method successfully generates images with an arbitrary head pose, which indicates our property checking can guide the property tuning network to generate data with target property. At the same time, the image with the target pose can still retain the same identity of the input image, which shows our invariant checking can anchor the invariant information during image manipulation. To perform a quantitative comparison with other methods, we generate 10 images by head pose with ground truth images. The Pix2pix [13] is compared with by using an additional one-hot encoding to generate images. We conduct PG^2 by using the facial landmarks extracted by dlib [14] as pose guidance. StarGAN and STGAN are compared with for the generation of discrete samples. Table 1 reports the quantitative comparison of the above methods. We outperform other methods on some of the evaluation metrics, no matter whether they are trained with paired or unpaired data. The visualization of the comparison results is shown in Figure 5. Compared with StarGAN and STGAN, the image generated by TunerGAN can retain the same emotion, identity as the input, and have more real details and fewer artifacts on the facial features.

In addition, we conduct the continuous **emotional expression tuning** on RaFD [15], which is similar to emotion changing as described in GANimation [26], without requiring a lot of intermediate samples and hand-

Table 1. Quantitative comparison of viewpoint transfer (facial pose translation) on FEI dataset. The symbol ↑ (↓) indicates that the larger (smaller) the value, the better the performance. † represents the methods trained with paired samples.

Method	SSIM↑	PSNR↑	MSE↓	RMSE↓	LPIPS↓	FID↓	Classification Acc↑
Pix2pix† [13]	0.9215	16.90	0.0162	0.1232	0.2273	74.31	0.9450
PG^{2}† [23]	**0.9354**	**17.34**	**0.0145**	**0.1160**	**0.1887**	70.70	0.9525
StarGAN [4]	0.8812	15.17	0.0247	0.1512	0.2567	93.26	0.9325
STGAN [18]	0.8931	15.44	0.0232	0.1483	0.2514	76.56	0.9350
TunerGAN	0.9219	16.68	0.0199	0.1288	0.2123	**66.56**	**0.9650**

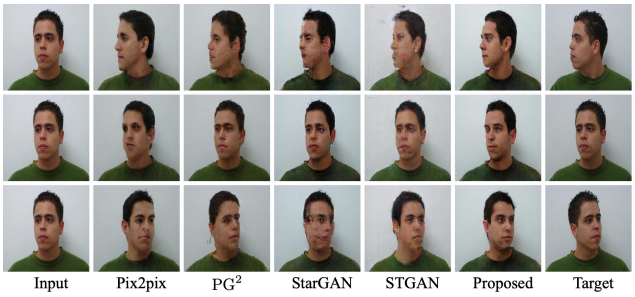

Figure 5. Visual result examples of supervised and unsupervised methods on FEI dataset.

crafted labels. We tune the continuous expression among 8 sparse emotional samples, following the guidelines of the FACS system [5] and the emotional changing procedure: Neutral→Contemptuous→Disgusted→Happy→Angry→Sad →Surprise→Fear. Figure 4(b) shows continuous emotional changing procedure between two different emotional representations. The quality of the tunable property of emotion can be checked especially from the visualization of the pose of the mouth. From the illustrations, we can see that the property is continuously changed along with the emotion space, while identity remains invariant. We provide the visual and quantitative comparisons of all the methods in the supplementary file.

Finally, **age manipulation** aims to generate the past and future images given only a single image of a certain person. The generated image should have the same identity and gender but with different ages. The results of tuning the age property are shown in Figure 4(c). The images in red and green boxes are real samples and others are generated samples by tuning age on them. Our method can realize a continuous age changing process with one image, even if there is only one image for some persons in training. To evaluate the image manipulation performance we use the online face detection API [1] to estimate the age of the generated images. Table. 2 shows the quantitative comparison of different methods. For evaluation, we compute the average age error (AE) between the estimated age and the desired age label, and the face similarity (FS) between the generated image and the real image to evaluate whether the identity is retained. We generate 40 images (age ranging between 20 and 59) for each image. We evaluated 100 images from different persons, and the labels of input images are also between 20 and 59. We divide the input images into four ranges 20-29, 30-39, 40-49, and 50-59. As shown in Table. 2, we achieve the lowest average age error and FID compared with StartGAN and STGAN on all of the age ranges at the cost of a few percent relative loss in FS. StarGAN and STGAN just maintain the similar appearance of the input images so that the face similarity is high.

Besides tuning face attributes, we also conduct experiments on scene data. We show an example of applying TunerGAN to fog thickness tuning on the Foggy Kitti dataset [7], with visual results shown in Figure 6. More visual generation results and comparison with other methods can be found in our supplementary file.

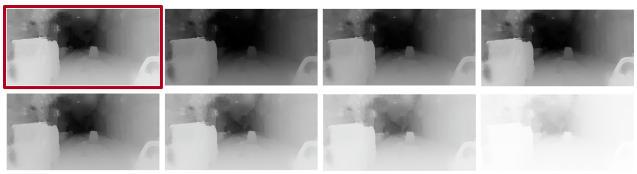

Figure 6. Visual results on fog thickness tuning over an input image with 375m visibility. The image in red box is the input.

4.4. Multiple-property tuning

Our algorithm can also be modified to handle multiple-property tuning by adding parallel input/output tuning messages, one per property. We conduct experiments on the CelebA dataset [20]. We select 6 parallel properties for tuning: neutral to smile, no beard to goatee, bald to bangs, black hair to brown hair, young to old, and male to female. Figure 7 shows the visual generation results. The first 6 rows show the tuning the six parallel face attributes individually, the 7th and 8th rows represent the results of tuning two properties at the same time. We tune all the properties at the same time in the last row.

We conduct ablation experiments on facial head pose rotation task, with results shown in Table. 3 to analyze the effectiveness of the proposed modules. **Multi-Branch Estimator**: we evaluate it with four different settings: 1. one-branch capsule regression and discriminator (w/o

[1] https://www.faceplusplus.com.cn/face-detection

Table 2. Quantitative comparison of viewpoint transfer (facial pose translation) on UTKface dataset. AE and FS represent the average age estimation error and face similarity, separately.

Method	FID↓	20-29(AE↓ / FS↑)	30-39(AE↓ / FS↑)	40-49(AE↓ / FS↑)	50-59(AE↓ / FS↑)
StarGAN [4]	70.26	4.129/0.9782	4.198/0.9672	4.569/**0.9718**	4.657/0.9741
STGAN [18]	62.36	3.387/**0.9819**	2.872/**0.9796**	3.873/0.9862	3.671/**0.9827**
TunerGAN	**50.80**	**2.346**/0.9632	**1.984**/0.9736	**2.468**/0.9582	**2.580**/0.9481

Table 3. Quantitative comparison of different settings on FEI dataset.

ID	Method	SSIM↑	PSNR↑	MSE↓	RMSE↓	LPIPS↓	FID↓	Classification Acc↑
Exp1	w/o MB	0.8813	15.11	0.0273	0.1652	0.2612	115.6	0.9025
Exp2	w/ MB(N=2)	0.9126	15.97	0.0226	0.1401	0.2291	76.83	0.9325
Exp3	w/o capsule	0.9089	15.79	0.0239	0.1405	0.2285	82.63	0.9425
Exp4	StarGAN+MUNIT	0.9015	15.27	0.0236	0.1501	0.2418	78.22	0.9375
Exp5	TunerGAN	**0.9219**	**16.68**	**0.0199**	**0.1288**	**0.2123**	**66.56**	**0.9650**

Figure 7. Tuning multiple properties on CelebA.

MB), 2. two-branch capsule regression and discriminator (w/MB(N=2)), 3. four-branch regression without capsule and discriminator and 4. four branches as proposed. They correspond to Exp1, Exp2, Exp3 and Exp5, respectively. We can observe the multi-branch architecture can promote the performance of both discrimination and regression.

Capsule Regression: Comparing Exp3 with Exp5, we can see that adding the capsule regression to check the property improves the image generation performance on all metrics. This demonstrates that the disentanglement of representation can help the generator to capture more informative details. Exp1 and Exp5 show that we obtain more accurate regression with a multi-branch structure.

4.5. Ablation study

Invariant feature: To illustrate that our invariant feature does not end up learning a constant vector, we have visualized the activation of the instance-invariant capsule. We compare two settings: 1) pose invariant identity change, 2) identity invariant pose change. The visual results are shown in Figure 8. From the visualization, we observe that the invariant representation learned in these two tasks are very different. For pose change, it focuses on facial features (nose, eyes), while for identity change, it focuses on hair and cloth. We also computed the mean feature map per pose angle to visualize the learned invariance.

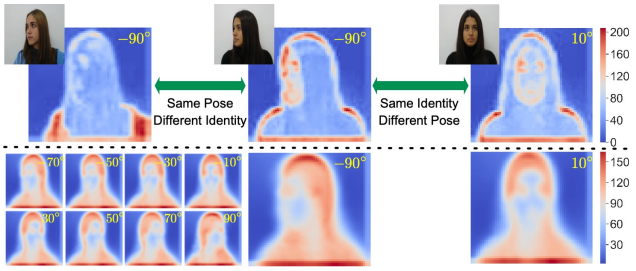

Figure 8. Upper: Learned invariant features for 3 exemplar images. Lower: Mean feature map per pose angle of FEI dataset.

To illustrate that our method is not just a combination of StarGAN and MUNIT by applying the content-preserving loss to StarGAN, we implemented the StarGAN+MUNIT in Exp4 (see Table 3). TunerGAN outperforms the StarGAN+MUNIT by automatically sampling data-label pairs, going beyond existing labels. Our coupled capsule network also provides a stronger property invariant protection than the content-preserving loss in MUNIT.

5. Conclusion

We have proposed an instance-invariant property tunable image manipulation method that is trained with synthetically-generated images. We disentangled the image representation into the invariant information and property information. We ensured the instance-invariant information to be consistent during the manipulation and checked whether the image having the target property by two types of capsule networks. We have demonstrated the proposed TunerGAN method outperforms all existing algorithms on instance level property tuning image manipulation tasks.

References

[1] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*, pages 2172–2180, 2016. 1

[2] Ying-Cong Chen, Xiaogang Xu, Zhuotao Tian, and Jiaya Jia. Homomorphic latent space interpolation for unpaired image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2408–2416, 2019. 3

[3] Zhimin Chen and Yuguang Tong. Face super-resolution through Wasserstein GANs. *arXiv preprint arXiv:1705.02438*, 2017. 3

[4] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, pages 8789–8797, 2018. 1, 3, 5, 7, 8

[5] Rosenberg Ekman. *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*. Oxford University Press, USA, 1997. 6

[6] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015. 3

[7] Shirsendu Sukanta Halder, Jean-François Lalonde, and Raoul de Charette. Physics-based rendering for improving robustness to rain. In *IEEE/CVF International Conference on Computer Vision*, 2019. 5, 7

[8] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *NIPSs*, pages 6626–6637, 2017. 5

[9] Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. Transforming auto-encoders. In *International Conference on Artificial Neural Networks*, pages 44–51, 2011. 3

[10] Geoffrey E Hinton, Sara Sabour, and Nicholas Frosst. Matrix capsules with em routing. 2018. 3

[11] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *International Conference on Pattern Recognition*, pages 2366–2369. IEEE, 2010. 5

[12] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *European Conference on Computer Vision*, pages 172–189, 2018. 1, 3

[13] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 5967–5976, 2017. 3, 5, 6, 7

[14] Davis E King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10(Jul):1755–1758, 2009. 6

[15] Oliver Langner, Ron Dotsch, Gijsbert Bijlstra, Daniel HJ Wigboldus, Skyler T Hawk, and AD Van Knippenberg. Presentation and validation of the Radboud Faces Database. *Cognition and Emotion*, 24(8):1377–1388, 2010. 5, 6

[16] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, pages 4681–4690, 2017. 3

[17] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *European Conference on Computer Vision*, pages 35–51, 2018. 3

[18] Ming Liu, Yukang Ding, Min Xia, Xiao Liu, Errui Ding, Wangmeng Zuo, and Shilei Wen. Stgan: A unified selective transfer network for arbitrary image attribute editing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3673–3682, 2019. 1, 3, 5, 7, 8

[19] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 5

[20] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *International Conference on Computer Vision*, pages 3730–3738, 2015. 7

[21] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 2

[22] Yu Luo, Yong Xu, and Hui Ji. Removing rain from a single image via discriminative sparse coding. In *ICCV*, pages 3397–3405, 2015. 2

[23] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *Advances in Neural Information Processing Systems*, pages 406–416, 2017. 5, 7

[24] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 3

[25] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, pages 2536–2544, 2016. 3

[26] Albert Pumarola, Antonio Agudo, Aleix M Martinez, Alberto Sanfeliu, and Francesc Moreno-Noguer. Ganimation: Anatomically-aware facial animation from a single image. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 818–833, 2018. 1, 6

[27] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, pages 3859–3869, 2017. 3

[28] Casper Kaae Sønderby, Jose Caballero, Lucas Theis, Wenzhe Shi, and Ferenc Huszár. Amortised map inference for image super-resolution. In *ICLR*, pages 1–17, 2017. 3

[29] Carlos Eduardo Thomaz and Gilson Antonio Giraldi. A new ranking method for principal components analysis and its application to face image analysis. *Image and Vision Computing*, 28(6):902–913, 2010. 5

[30] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018. 3

[31] Cort J Willmott and Kenji Matsuura. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate research*, 30(1):79–82, 2005. 5

[32] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *CVPR*, pages 6721–6729, 2017. 3

[33] Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *CVPR*, pages 5485–5493, 2017. 2

[34] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. DualGAN: Unsupervised dual learning for image-to-image translation. In *International Conference on Computer Vision*, pages 2849–2857, 2017. 3

[35] He Zhang, Vishwanath Sindagi, and Vishal M Patel. Image de-raining using a conditional generative adversarial network. *arXiv preprint arXiv:1701.05957*, 2017. 2

[36] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018. 5

[37] Zhifei Zhang, Yang Song, and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *CVPR*, pages 5810–5818, 2017. 5

[38] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *International Conference on Computer Vision*, pages 2223–2232, 2017. 1, 3