

Supplement to Guided Flow Field Estimation by Generating Independent Patches

Mohsen Tabejamaat
mohsen.tabejamaat@inria.fr

Inria

Farhood Negin
farhood.negin@inria.fr

Francois Bremond
francois.bremond@inria.fr

A Why do α and β meet their own functionalities?

One fundamental question is how we can ensure that α and β respectively focus on the visible and invisible parts of the source sample. This can be explained by three different characteristics of these functions:

- α is a coefficient applied on the source sample which is squashed to the range $[0, 1]$, so it can suppress the pixels but does not introduce anything into the image content.
- β is bounded in the range $[0, \infty)$. Being greater than zero, it can not suppress nor reduce the pixel values of the source sample but add some new contents into it.
- α and β are part of the learnable parameters in our generative model, so they learn through the loss function of the network: $L(y, G(x; \alpha, \beta))$. For a loss function, a simpler solution means a shorter path in the loss space. Keeping the similar pixels of the source and target samples untouched provides a much simpler solution than making a fundamental change in all the pixels and then generating them from scratch. For α , this exclusively means honing on the visible part of the sample, trying to keep fixed as much pixel values as possible, while for β it means trying to exclusively draw the invisible parts of samples, ignoring those pixels that has already been available by the source sample. Following this shorter path is ensured by a specialized optimizer (like Adam optimizer in our model).

B Implementation Details

In this section, we briefly review the implementation details of our method (Figure 1). Apart from the dimensions of the input layer, both the encoders are built upon the same architecture, which is a set of N convolutional layers, each followed by the Batch Normalization and ReLU activation function. We consider their output to be of the size $C \times H \times W$, where we set C to 256 for Deepfashion and 128 for the Market1501 dataset. For the perceptual

$s_{old} \in R^{C \times m \times n}$: Old appearance code, for the first IC, it is output of the encoder E_1
 $s_{new} \in R^{C \times m \times n}$: Updated feature map
 $t_{old} \in R^{C \times m \times n}$: Old pose code, for the first IC, it is output of the encoder E_2
 $s_{new} \in R^{C \times m \times n}$: Updated pose code

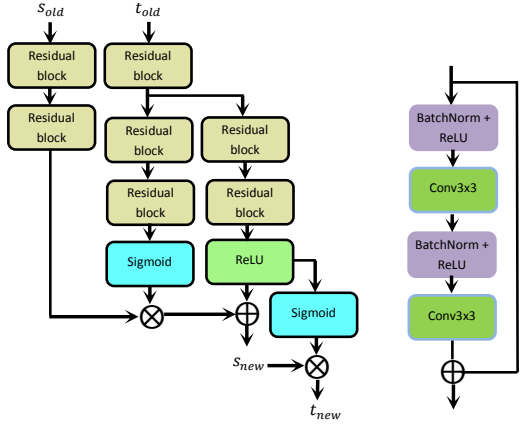
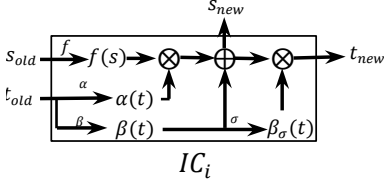


Figure 1: Overall framework of an Incarnation Block; left: Structure of different functions in each block, middle: Implementation using Residual Blocks, right: Residual component in our proposed architecture of Incarnation Blocks

and style losses, we utilize the VGG19, pre-trained on ImageNet database and extract the embeddings from the $Conv_i; i = 1 : 5$ layers. α , β , and f are all constructed based on the Batch Normalization. As the standard design choice, we consider 9 IC blocks that meet our demands for a fair trade-off between the accuracy and speed. For the Merging module, we use three convolutional layers for the first and one layer for the second part of the merging operation, each followed by the Batch normalization and ReLU activation function. All the convolutions are performed at the stride 1, therefore the output of this module has the same spatial dimensions as the input. The structure of the decoder is considered to be symmetric with the E_1 , except for the deconvolutional operations.

C Ablation study

In this section, we discuss some alternative configurations of our method to validate the structure we selected for each part of the network. All the experiments are conducted on the Deepfashion database. The basic structure of the network is the same as we discussed in Section 3 of the paper with a modification made just to a specific part.

C.A PG vs PATN

This section aims to examine whether generating patches works in reality and performs better than the blind combination of the pose and appearances [4]. To do so, we initialize our method with the transfer blocks of PATN (PATBs) [4] instead of the PG blocks in the original configuration. For a fair comparison, both the configurations (PG and PATBs) are endowed with 9 blocks. The comparison results are shown in Table 1. As can be seen, FID score for the original structure of our method is 10.8 which is about 2 scores better than the alternative configuration. In fact, using PATBs makes the network to loss some efficiency in keeping the fidelity of the generated textures which can be allocated to the inference of the blocks in the task of displacing the patches. This interference and the resulting dependence between the

modules, makes it unable to correctly interact with the output of the PT module. In addition, part of this failure can also be allocated to the blind combination of the pose and textures which comes from the consecutive concatenations of the PATB blocks.

	IS \uparrow	SSIM \uparrow	FID \downarrow	LPIPS \downarrow
PATN+PG	3.3892	0.7770	12.18	0.2214
PT+PG	3.4621	0.7767	10.80	0.1991

Table 1: Performance evaluation by replacing PG module with PATNs in the proposed structure of our method

C.B Contribution of PG

In this section, we study the effectiveness of the PG module in the overall performance of our network. To do so, we disentangle the PG module from the main configuration and examine the performance of the resulting network. The results are shown in Table 2. As can be seen, by ignoring the PG module, we get some degraded results compared to the full version of our network. This confirms that, PG does really play a positive role in directing the warping features towards the critical areas of the output sample such as clothing regions.

	IS \uparrow	SSIM \uparrow	FID \downarrow	LPIPS \downarrow
Our method without the PG module	3.3125	0.6938	101.48	0.4308
Our method without the PT module	3.4292	0.7668	11.97	0.2107
Full structure	3.4621	0.7767	10.80	0.1991

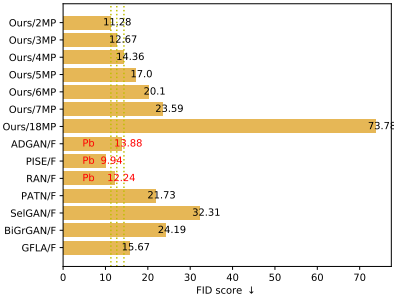
Table 2: Performance evaluation of our method with ablated PG and PT modules

C.C Contribution of warping features

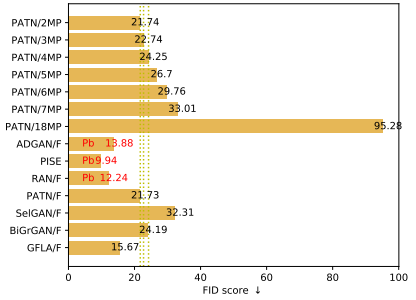
In this section, we validate the contribution of the warping features in improving the performance of our network. Table 2 compares the results using the full version of the network and also its counterpart without the warping module. As can be seen, ignoring the warping features drop the SSIM by 1% which is not a significant number as warping seems to more contribute in correcting the color tones and fine grained textures. Comparing the results with Tables 1, it can be seen that the overall performance is much better when PG and PT are employed together rather than just using one of these modules.

C.D Consistency of normalization

This section aims to validate the effectiveness of the interconsistency in the performance of our method. For this purpose, we consider two different structures. In the original configuration, we consider a Spectral+Bach normalization scheme for the encoders and decoder of the network while endowing the others with just Batch Normalization layers. For the second strategy, all the modules are constructed with the Batch normalization layers. To measure the concept of consistency, we propose to calculate the l_1 distance between the output of the merging module in the forward mode and output of E1 when utilized in a backward mode, and refer to it as the consistency score. This way, we measure the fidelity of the output



(a) Performance evaluation of our method when there are up to K missing points in the target pose $k \in \{2, 7\}$, F denotes a full model when there is no missing keypoints and kMP stands for k Missing Points.



(b) Performance evaluation of PATN when there are up to K missing points in the target pose $k \in \{2, 7\}$, F denotes a full model when there is no missing keypoints and kMP stands for k Missing Points.

Figure 2: Comparison of the robustness of our method to the sensitivity of the pose estimator

patches to the original textures. With the backward mode, we mean feeding the network with the target sample instead of the source image. Our experiment shows that the interconsistency can boost the FID by 0.2 score which is not significant but it still has a significant role in stabilizing the training process (Table 3).

	Full model	All in Batch Normalization	PATN
Int. Cons. Score	0.4305	0.5708	0.8885

Table 3: Performance evaluation for intraconsistency of our method

C.E Robustness to missing points

We conduct an experiment to examine the robustness of our method to the sensitivity of the pose estimator. To do so, we first train the network with the complete number of skeletal points. For inference, we randomly drop out up to K of the keypoints from the target pose and remeasure the performance. This way, we can determine the sensitivity of our method to that missing points. For a fair comparison, PATN with 9 blocks is considered as the baseline method. From the results in Figure 2, we see that FID of our method just drops by about 3.5 scores when dropping up to 4 keypoints from the target sample. This verifies the robustness of our method against the sensitivity of pose estimator that can be allocated to using heatmaps as the pose representation rather than drawing raw skeletons. When compared the results with PATN, despite the fact that both the methods benefit from the heatmaps, our method shows more robustness (21.74→95.25 for PATN vs. 11.28→73.78 for our method) that seems to come from our strategy for the disentangled experts on localizing the source and target samples where even with the failure in one of the experts, there is still another one to guide the sample towards the correct locations.

C.F Ablation of the loss function

We also conduct another experiment to evaluate the individual contribution of the loss terms. Each time a term is ablated from the overall function and then the performance is compared with the baseline model (the model with all terms of the loss function). The experiment includes ablating perceptual, adversarial, and content losses. The results are shown in Table 4. As can be seen, adversarial and style losses both reduce the IS and SSIM scores, but has a significant role on improving the naturalism of image, so that ablating them (especially the style loss) significantly increases the FID score.

	IS↑	SSIM↑	FID↓	LPIPS↓
Our method w/o L_{pr}	3.4240	0.7707	12.41	0.203
Our method w/o L_{st}	3.4742	0.7788	13.34	0.208
Our method w/o L_{adv}	3.4930	0.7787	12.36	0.202
Our method (baseline)	3.4621	0.7767	10.80	0.194

Table 4: Performance evaluation by ablating different terms of the loss function

D Additional qualitative results

Figures 3, and 4 are additional qualitative results of our method.

E How to chose the regularization coefficients

Clearly, setting the regularization weights should be in principle based on an extensive grid search. But in our case, huge number of network parameters hinders us from doing a real grid walk on a fair grid of space. As an alternative, we simply set each coefficient so as to make its corresponding loss term roughly less than 3 (except for the l_1 loss which is set to be less than 1, in order to suppress the blurring effect of this term), e.g. $\lambda_1 * l_{loss} = < 1$. Our experiments demonstrate that there is not much differences between the start value of the loss functions in comparison with other values, just being in the roughly same range makes a reasonable choice for the start point of the terms.

F PG vs SPADE

SPADE [14] is a normalization technique proposed to deal with the uniform neural response of generators in photorealistic image synthesis from semantic layouts. It proposes to make the fixed de-normalization part of the Batch normalization technique conditioned on an input semantic map aimed to preserve those parts of semantic information which might wash away through the affine transformation of the Batch normalization. There are some works that tried to utilize SPADE in the context of pose generation techniques [15, 16]. However, the problem is that, each SPADE block needs to take as input the semantic map of the source sample which is not applicable in a parser-free framework. Unlike SPADE, our method can progressively estimate pose maps and incorporate them into the next estimation of the output sample, it does not require any semantic map, and its pose estimation is progressively performed through conditioning on just an initial key point representation.



Figure 3: Additional Samples generated by our method, in each triplet the left is the source sample and the next ones are generated by our method

G Visualization of α and β

Visualizing α and β can provide us with a better understanding about the functionality of these parameters. It is noteworthy that both α and β are volumetric tensors that their 2D visualizations will sacrifice the interactive capability of their channels. But just as a very rough estimation we visualize the averaged value of all the channels. α starts with attending to the locations of the source sample, but as the sample is modified towards the target shape, α is also modified to the new locations, such that for the last blocks it seems that α more attends to the locations of the target sample. Some examples of $\alpha f(s)$ and β are illustrated in Figure 5. For the first example, we can easily observe that $\alpha * f(s)$ starts by trying to remove the side view of the arm which is not going to be visible in the target view while keeping the remaining parts untouched in order to be further modified towards the values of

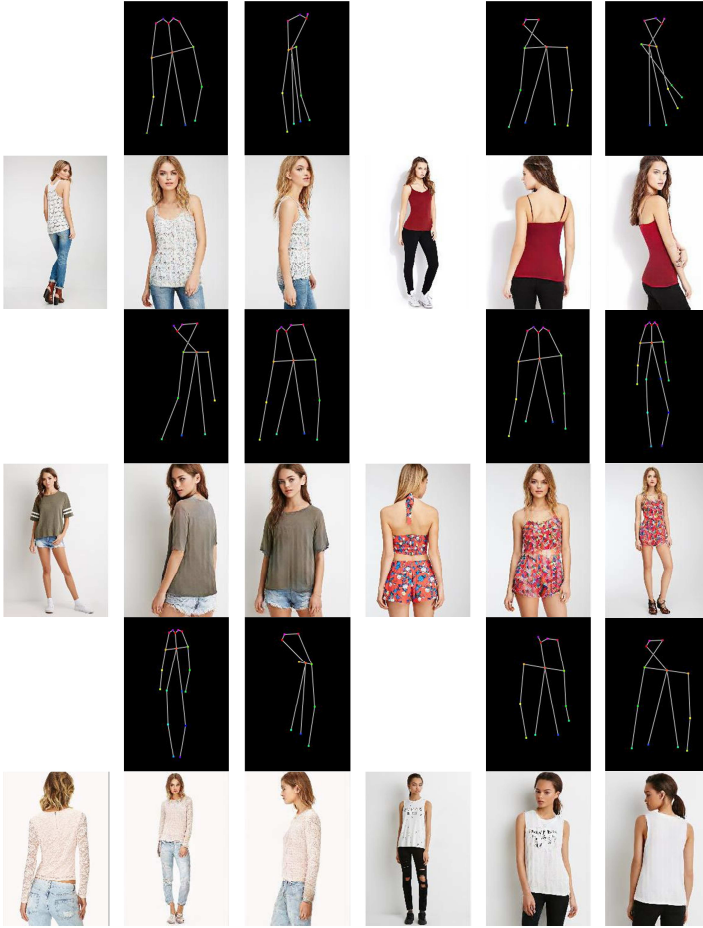


Figure 4: Additional Samples generated by our method, in each triplet the left is the source sample and the next ones are generated by our method

the target sample. In contrast, β always attend to the most critical parts of the target sample which has not been appropriately reconstructed by the previous estimation of the α value.

References

- [1] Zhengyao Lv, Xiaoming Li, Xin Li, Fu Li, Tianwei Lin, Dongliang He, and Wang-meng Zuo. Learning semantic person image generation by region-adaptive normalization. *arXiv preprint arXiv:2104.06650*, 2021.
- [2] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019.

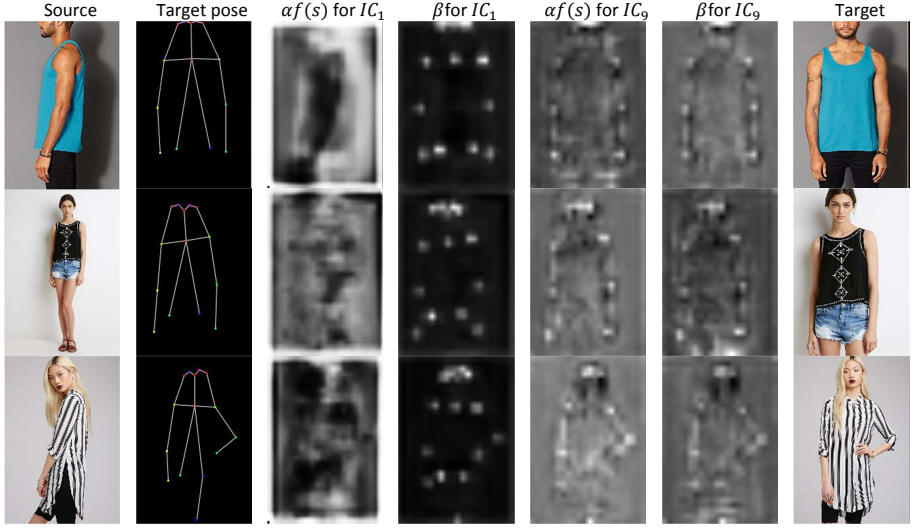


Figure 5: Visualization of α and β for the incarnation blocks of the PG module

- [3] Jinsong Zhang, Kun Li, Yu-Kun Lai, and Jingyu Yang. Pise: Person image synthesis and editing with decoupled gan. *arXiv preprint arXiv:2103.04023*, 2021.
- [4] Zhen Zhu, Tengpeng Huang, Baoguang Shi, Miao Yu, Bofei Wang, and Xiang Bai. Progressive pose attention transfer for person image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2347–2356, 2019.